

Received June 12, 2021, accepted August 13, 2021, date of publication August 17, 2021, date of current version August 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105429

# A Low-Power BIST Scheme Using Weight-Aware Scan Grouping and Scheduling for Automotive ICs

KWONHYOUNG LEE<sup>ID</sup>, SANGJUN LEE<sup>ID</sup>, JONGHO PARK<sup>ID</sup>, INHWAN LEE,  
AND SUNGHO KANG<sup>ID</sup>, (Senior Member, IEEE)

Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Sungho Kang (shkang@yonsei.ac.kr)

This work was supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Evaluation Institute of Industrial Technology (KEIT) (No. 20012010).

**ABSTRACT** Scan-based logic built-in self-test (LBIST) is widely used for supporting the in-system test in automotive systems. Although this technology has the advantage of low-cost testing, it suffers from low fault coverage and high switching activity during the test. This can lead to many undetected defects, excessive heat dissipation, and IR drop, inducing catastrophic risks to functional safety. Therefore, improving these two key factors is crucial to alleviate reliability problems in the automotive domain. Most previous works have focused on controlling the enormous toggling level of random patterns; however, one of the main disadvantages of these approaches is low fault coverage. Unfortunately, additional hardware costs associated with memory elements or test points are required for detecting the remaining faults. We propose a novel LBIST scheme based on weight-aware scan grouping and scheduling (WGS) to overcome these difficulties. Since the required test time of each automotive product is limited, the proposed scheme freezes the test time and focuses on improving both aforementioned factors significantly. Our approach divides scan cells into two categories: the coverage-efficient scan group and power-efficient scan group, and then it conducts weight-based scan cell reordering. Biased random patterns are fed to enhance fault coverage for the first category. For the second category, scheduling and disabling are performed to reduce switching activity. Finally, physical-aware reordering based on an inverter is performed to reduce routing overhead. Experimental results demonstrate the feasibility of the WGS methodology on the ITC'99 and OpenRISC benchmark circuits.

**INDEX TERMS** Design for testability, low-power testing, scan-based logic BIST, scan cell ordering, scan chain scheduling.

## I. INTRODUCTION

The global automobile industry has reached an inflection point. The original purpose of the automobile—providing simple transportation—has evolved to serve as a mobile hub characterized by keywords like electric vehicles and autonomous driving. Hence, the demand for integrated circuits (ICs) for supporting such systems is expected to increase rapidly. Automotive electronics content will account for 50% of the cost of a car by 2030 [1]–[3]. In particular, items related to functional safety are crucially considered because the operation of most ICs in this domain has a direct impact on human lives. Besides the manufacturing test, the in-system test, which can periodically test ICs during functional operation

or idle time, should be implemented to ensure the reliability of chips embedded in safety-critical application systems. For instance, the ISO 26262 standard mandates compliance with the automotive safety integrity level (ASIL) to prevent fatal car accidents due to chip malfunction in electronic components embedded in vehicles [4]. Moreover, the ASIL grades range from A to D, with D representing the highest degree of automotive hazard. An advanced solution for each ASIL grade is required to produce high-quality test data within a given test application time.

Scan-based logic built-in self-test (LBIST) is considered to be an appropriate technique in many industries to satisfy these requirements. This scheme employs the pseudorandom pattern generator (PRPG) producing all possible random patterns, excluding all zeros, to detect potential faults [5], [6]. This scheme is more commonly adopted than

The associate editor coordinating the review of this manuscript and approving it for publication was Poki Chen<sup>ID</sup>.

other approaches such as software-based self-test (SBST) and deterministic BIST due to its decisive advantage of being easy to implement with small hardware overhead. SBST [7]–[10] does not require additional on-chip hardware resources; however, long-term development efforts to understand the circuit under test (CUT) and create high-quality patterns are required. Deterministic BIST [11]–[13] is limited because it requires memory to store deterministic patterns. Therefore, it is only used for specific products that require extremely high fault coverage achievement.

Despite the advantages mentioned above, the conventional LBIST scheme is vulnerable to random pattern resistant (RPR) faults [14]; it consumes more enormous switching power than the typical mission mode [15]. These are due to the characteristic of the unbiased pseudorandom patterns used for internal circuits in which each pattern has a 50% average combination of zeros and ones. Moreover, significant RPR faults and excessive switching activity problems can lead to undetected defects, as well as IR-drop, power noise, and excessive thermal dissipation. Eventually, these issues can develop into complex and severe problems such as reliability degradation, chip malfunction, and yield loss [16], which can pose catastrophic risks to functional safety. Therefore, improving key factors of fault coverage and switching activity within a limited test application time is crucial and challenging in the automotive domain.

Numerous studies were conducted to address these challenges. In the past, weighted random pattern testing [17]–[21], perturbed random pattern testing [22]–[25], deterministic pattern encoding [26] were considered efficient approaches to detect RPR faults. In the first category, unbiased random patterns are transformed into various sets of biased random patterns to detect RPR faults probabilistically while loading and unloading patterns. In the second category, desired patterns that contribute to fault coverage are obtained by perturbing existing pseudorandom patterns. For example, the PRPG reseeding [24] and bit-flipping [25] methods modify the sequence of random patterns to skip undesired test patterns that do not contribute to the detection of faults significantly. In the last category, deterministic base patterns transformed by regularity analysis are encoded into scan chains to generate parent and child patterns [26]. Therefore, the methods introduced so far effectively improve fault coverage or test time by incorporating additional hardware resources. Although these methods are useful, it remains a challenge to minimize high switching activity concurrently.

Scan cell ordering, scan chain disabling techniques, and low-transition pattern generators were proposed to reduce excessive switching activity during testing. The scan cell ordering schemes [27]–[30] are an alternative to reduce switching activity by selecting scan cells with high internal switching power and reordering them to a location with low switching weights. However, these schemes can induce excessive routing overhead based on constraints on the position of scan cells. Accordingly, the improvement rate of wire length can be low even if physical-aware scan reordering is

performed. Moreover, because most methods are based on deterministic patterns and not on random patterns, additional hardware costs (read-only memory and other memory elements) can incur for storing these patterns. The scan chain disabling schemes [16], [31]–[33] can reduce switching activity by preventing the random patterns from being applied to specific scan chains using additional gating logic during the scan shift-in and shift-out. Zoellin *et al.* described a low power test planning scheme based on a set of scan chains to be enabled on specific patterns [31]. Filipek *et al.* presented a control register that can disable scan chain groups per pattern to minimize the toggling level [32]. Afterward, the preselected toggling generator [33] was introduced to produce pseudorandom patterns with a more accurate switching level. Additionally, the low-transition pattern generator techniques [34]–[37] can handle a certain level of toggling by producing modified intermediate low-transition patterns. Omana *et al.* proposed the low-cost approach (LCA) scheme that is capable of reducing the active factor by 50% with low hardware overhead [37]. The approaches mentioned above are effective in terms of controlling switching power; however, it is challenging to improve fault coverage concurrently using low-transition random patterns. Unfortunately, additional hardware costs (e.g., deterministic test data and test points) are required to address these difficulties.

Alternatively, the recent approaches [38], [39] reduced the size of the deterministic pattern set by selectively flipping the base parent pattern. These methods require changing the test data instead of modifying the scan structure, meaning the preprocessing step of engineering change orders (ECOs) is unnecessary. Other alternative methods [40], [41] were proposed for inserting conflict-aware test points to improve random testability effectively. However, it may be challenging to apply these methods for designs that require limited hardware resources. Furthermore, adopting an extensive set of deterministic patterns and test points can be prohibitive because it is expected that the optimization of the performance power area (PPA) of ICs will become more challenging on increases in circuit complexity [13]. Therefore, it is necessary to improve both fault coverage and low-test power effectively using low-cost hardware within the targeted test application time.

This paper proposes a novel scheme using weight-aware scan grouping and scheduling (WGS) to enhance fault coverage and reduce switching activity with limited test time and hardware overhead. Since the test time required for each automotive product to perform the online or offline test is limited [42], our method freezes the test time in advance and focuses on significantly improving both the target factors. Most of the previous methods can handle the high toggling level of random patterns; however, they negatively affect fault coverage. In order to overcome these difficulties, the proposed method first partitions scan cells into two categories based on the specified bit usage. These categories are called the coverage-efficient scan group and power-efficient scan group. In the former category, biased random patterns are

used to increase fault coverage effectively. In the latter category, the toggling level can be reduced by scheduling and disabling appropriate scan groups based on a constant interval. After completing these processes, routing overhead is further reduced by alleviating scan cell ordering constraints. In the experiments, the ITC'99 and OpenRISC benchmark circuits are considered to evaluate the proposed method based on WGS. The experimental results demonstrate that our approach significantly outperforms the conventional BIST for both target factors. Additionally, we achieve average fault coverage improvement rates of 13% and 16% than those obtained by [32] and [37], respectively, based on an average power reduction rate of approximately 50%. Additionally, since the new scheme can predict the required memory capacity and power improvement rate in advance, the iterations of the ECOs can be saved significantly. It consequently minimizes the impact on the design cycle time.

The remainder of this paper is organized as follows. Section II introduces the motivation and overall flow of the proposed method. Section III demonstrates the proposed algorithms with step-by-step examples. Section IV details the hardware architecture required for the proposed solution. Experimental results are presented in Section V. Finally, Section VI concludes with summaries and the importance of our work.

## II. MOTIVATION AND OVERALL FLOW OF THE PROPOSED METHOD

### A. MOTIVATION

The proposed method focuses on three observations. First, scan cells using a large number of specified bits are concentrated into a few scan chains due to the characteristics of the

**TABLE 1. Results for the average specified bit ratio of each group based on the ISCAS'89, ITC'99, and OpenCores benchmark circuits.**

Benchmark	CUT	Total patterns	SR for parent scan cells (%)		
			0–10%	10–20%	20–30%
ISCAS'89	s13027	151	<b>47.9%</b>	22.5%	14.7%
	s15850	121	<b>58.8%</b>	36.2%	27.2%
	s38584	144	<b>58.5%</b>	31.4%	22.8%
ITC'99	b17	1079	<b>26.8%</b>	9.5%	6.8%
	b18	2209	<b>15.8%</b>	7.1%	4.9%
	b19	3872	<b>9.2%</b>	4.1%	3.0%
OpenCores	or1200	1846	<b>10.1%</b>	2.6%	1.5%
Average			<b>32.4%</b>	16.2%	11.6%

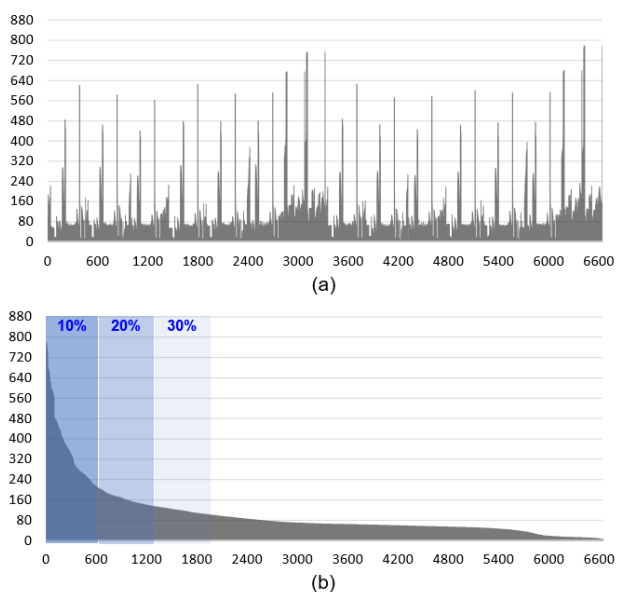
automatic test pattern generation (ATPG) tool engine [38]. Second, fault coverage ramps up significantly at the beginning of random patterns and it then gradually saturates at a certain point. Third, two different biased random patterns can be loaded into a single scan chain through weight-inversion-based scan reordering.

The first observation is that a few scan cells typically use considerable amounts of specified bits within a deterministic pattern set. Fig. 1(a) shows an example of the distribution of the specified bit usage in each scan cell for the ITC'99 benchmark b19 circuit. In this circuit, the scan cells are ordered in alphanumeric order using the conventional method. Hence, the scan cells with high and low specified bit usage are randomly mixed in this ordering. To determine the distribution of scan cells with a high specified bit usage, we sort all the scan elements in descending order based on the number of specified bits, as shown in Fig. 1(b). Then, the elements that belong to the top 10%, 10–20%, and 20–30% of the total number of scan cells are grouped and the average specified bit ratio used for each group is calculated as follows:

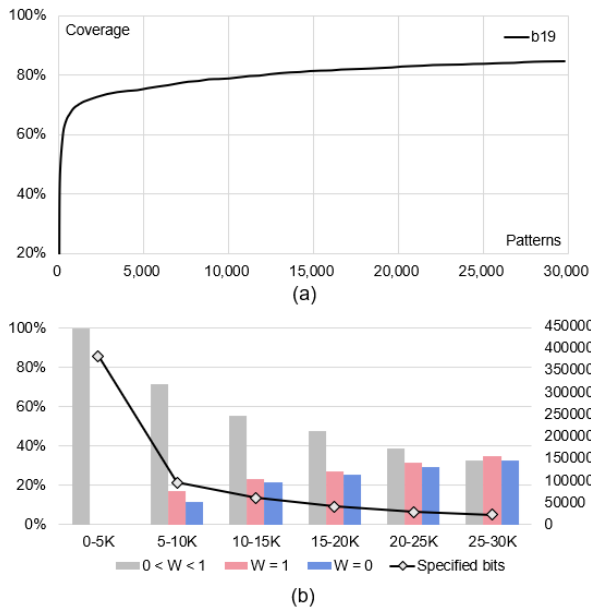
$$SR(G) = \frac{\text{total \# of specified bits used in } G}{\text{total\#of patterns} \times \text{\# of elements in } G} \quad (1)$$

where  $SR$  denotes the ratio of the average number of specified bits used in group  $G$  over the total number of patterns. Table 1 supports this observation, even for other benchmark circuits. According to the results in this table, approximately 10% of the upper scan cells have an average specified bit ratio of 32% during the test. On the other hand, the remaining groups show relatively high usage of don't-care bits. Since the number of upper scan cells that show high usage of the specified bits only accounts for a small fraction of the total number of scan cells, it is expected that fault coverage can be enhanced with low hardware overhead by primarily producing biased random patterns for a few upper scan cells.

The second observation pertains to the typical fault coverage gradient obtained with random patterns. Fig. 2(a) presents an example of the fault coverage gradient for the aforementioned b19 circuit. In this graph, the gradient is initially very steep and then gradually converges to zero. This



**FIGURE 1. Distribution of specified bit usage for each scan cell (a) Based on alphanumeric order (b) Based on the descending order by the number of specified bits.**



**FIGURE 2.** Examples of the second observation. (a) Fault coverage according to the increase in the number of random patterns. (b) Distribution of weights for scan cells and the trend of the number of specified bits used in the deterministic pattern set per 5K interval.

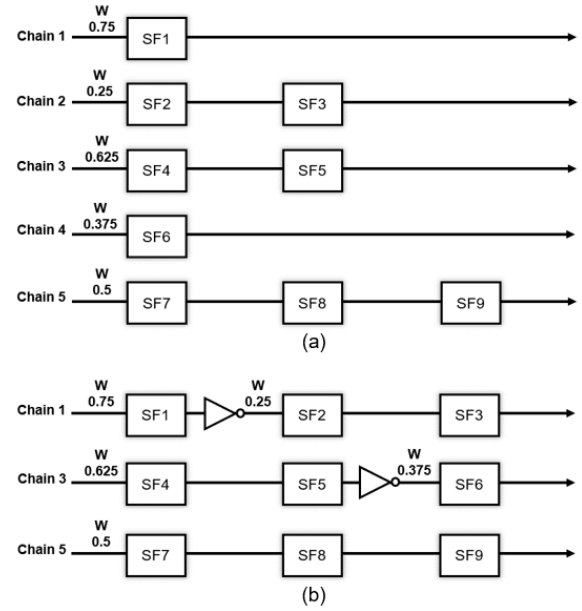
implies that the detected fault rate rapidly decreases with an increase in the number of random patterns. Accordingly, the number of specified bits used in each scan cell may be relatively low when referencing the deterministic test set generated from a small number of faults detected by random patterns. From this observation, the weight distribution of whole scan cells per constant interval can be obtained to examine the possibility of improving the toggling level in the following order:

- Partition the total number of random patterns,  $T$ , into the  $C$  pattern count. This yields  $T/C$  sets for random patterns.
- Extract detected fault lists by simulating each of the partitioned random pattern sets.
- Generate the deterministic test sets by using ATPG for each detected fault list.
- Calculate the weights of each scan cell using (2) by referring to each deterministic test set.

$$W(s) = \frac{\text{\# of specified 1 bits for } s}{\text{total \# of patterns} - \text{\# of don't care bits for } s} \quad (2)$$

where  $W$  denotes the ratio of the number of specified-1 bits to the total number of specified bits used in a scan cell. If the number of specified bits used in  $s$  is zero, then  $W$  is zero (Note that (2) is used in Section III-A and III-B).

Fig. 2(b) shows the results of the above procedure for  $T = 30K$  and  $C = 5K$ . For instance, as the scan cells in the current interval (5K to 10K) use a smaller number of specified bits than those in the previous interval (0K to 5K), the number of scan cells with weights of 0 or 1 gradually increases. This



**FIGURE 3.** Example of scan chain structure (a) Before weight-inversion-based scan cell reordering (b) After weight-inversion-based scan cell reordering.

observation can provide a clue for reducing the switching activity in adjacent scan cells when the scan cells under consideration are continuously loaded with constant values (equal to  $W$ ) for each  $C$  pattern count.

The third observation is related to weight-inversion-based scan cell reordering. As can be seen in Figs. 3(a)–3(b), since it is possible to apply several types of biased random patterns with a small number of scan chains [26], advantages in terms of a reduction in area can be obtained by replacing the combinational gates, which produce biased random patterns, with the inverter cells. Additionally, because the length of biased scan chains can be easily balanced by that of other scan chains, the loading and unloading time per pattern can be maintained along with the rest of the scan chains. Lastly, scan cells with the same weight range can be grouped together, and each group is partitioned based on the inverter. By adopting this scheme, scan cells within the identical group can be reordered to reduce the routing overhead further while minimizing the negative impact on fault coverage. Based on the above observations, each phase of our approach is detailed in Section III.

**B. OVERALL FLOW**

The overall flow of the proposed method using WGS is presented in Fig. 4. Each new procedure begins based on a gate-level netlist with scan chains. In the conventional electronic design automation (EDA) flow, scan cells are first ordered alphanumerically in the scan insertion stage. Subsequently, physical-aware scan reordering is performed incrementally to minimize the wire length of the scan chains. However, the proposed approach simultaneously improves

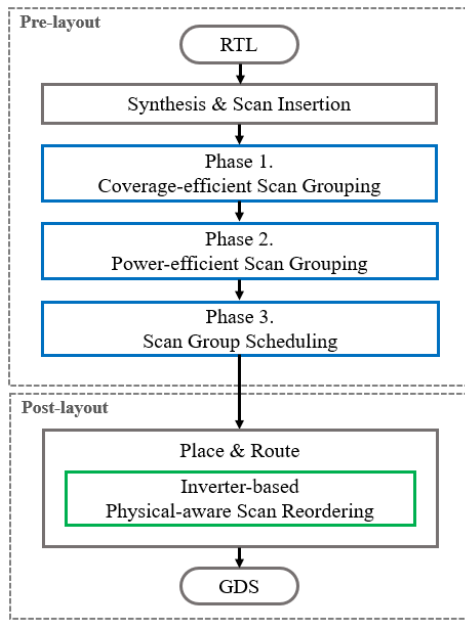


FIGURE 4. Overall flow of the proposed method.

two factors, namely fault coverage and switching activity, by adding three new phases (highlighted in blue) between the scan insertion and post-layout stages. Additionally, physical-aware scan reordering (highlighted in green) is performed to alleviate routing overhead.

The proposed method consists of three major phases and an additional wire-length-improvement phase: 1) coverage-efficient scan grouping, 2) power-efficient scan grouping, 3) scan group scheduling, and 4) inverter-based physical-aware scan reordering. These phases are conducted sequentially to improve the target factors.

In the first phase, for the achievement of high fault coverage within a limited test application time, a few effective scan cells are first chosen by calculating the specified bit usage and the weight of each scan cell from the deterministic test set. Next, the selected scan cells are reordered into a separated scan group, hereinafter referred to as the *coverage-efficient scan group (CSG)*. Biased random patterns are then applied to this scan group during the scan shift-in and shift-out.

In the second phase, to reduce the average toggling level while preserving the fault coverage enhancement introduced in the first phase, deterministic test sets are generated based on the detected faults using random patterns. These faults are acquired through the fault-dropping method. From the content of the deterministic test sets, the remaining scan cells with weights of 0 or 1 are identified and reordered into the rest of the scan groups, hereinafter referred to as the *power-efficient scan groups (PSGs)*. These scan groups are then sequentially disabled for each constant pattern count.

In the third phase, the sequence of the *PSG* enable times is adjusted to reduce the maximum toggling level while maintaining the average toggling level improvement

### Algorithm 1 Coverage-efficient Scan Grouping

Prepare the netlist with scan chain length  $L$  and list of all scan cells  $S = \{s_1, s_2, \dots, s_n\}$

- 1: Generate a deterministic test set by doing ATPG
  - a. Calculate the specified bit usage for all elements of  $S$
  - b. Sort the  $S$  in descending order by the specified bit usage
  - c. Divide the overall range of weights into  $b_1$  (0–0.25, 0.75–1),  $b_2$  (0.26–0.375, 0.625–0.74), and  $b_3$  (0.376–0.624)
  - d. Calculate weights for the elements of  $S$
  - e. Update  $S$  by removing elements with the range of  $b_3$
- 2: Determine the number of candidate cells  $m$
- 3: # Sequentially reorder the  $m$  scan cells into the biased scan chains
- 4: Initialize  $i \leftarrow 0$ ;
- 5: **while** ( $i < m$ ) **do**
- 6: Check the weight range of the highest-ranking scan cell
- 7: According to  $W$ , reorder the scan cell to the input or output of the corresponding biased scan chain  $B_1$  or  $B_2$
- 8:  $i \leftarrow i + 1$ ;
- 9: **end while**
- 10: Balance the length of  $B_1$  and  $B_2$  according to the length  $L$
- 11: Insert inverter cells between scan cells with inverted weights

FIGURE 5. Algorithm for coverage-efficient scan grouping.

introduced in the second phase. This can be achieved by repeatedly changing enable time slot with high switching activity to another time slot with low switching activity until the global minimum point of the weighted transition metric (WTM) [43] is identified.

Finally, physical-aware scan reordering can be performed incrementally based on inverter cells of each scan group with the help of the EDA tool engine to reduce the wire length of the scan chains. Because the order of most scan cells is frozen in the previous phases, an intermediate SCANDEF file is extracted to modify the hard cell ordering constraints manually. The downstream tool can reorder scan cells with identical weight ranges in each scan group by referring to this updated file.

## III. PROPOSED METHOD

### A. PHASE 1: COVERAGE-EFFICIENT SCAN GROUPING

Fig. 5 illustrates the proposed algorithm of Phase 1. The goal of this algorithm is to select a small number of scan cells that contribute significantly to the improvement of the fault coverage. The selected scan cells are then grouped into a separated *CSG*. Initially, a gate-level netlist with scan chain length  $L$  is employed to run this algorithm. Also, a list of all the scan cells  $S = \{s_1, s_2, \dots, s_n\}$  is prepared for selecting the coverage-efficient scan cells. Given these inputs, a deterministic test set is generated by the ATPG process and analyzed to select  $m$  candidate cells to be reordered. In this process, the scan cells with a high specified bit usage in the biased ranges of  $b_1$  and  $b_2$  are primarily extracted from list  $S$ . These scan cells are sequentially reordered into the corresponding biased scan chains ( $B_1$  or  $B_2$ ). Finally, after the scan reordering is complete, the lengths of the biased scan chains are balanced

until they match the scan chain length  $L$ . Inverter cells are then inserted between the scan cells with opposite weight ranges. A single group of biased scan chains created in this manner is defined as a CSG and is always enabled during the test, loading the biased random patterns.

*Example 1:* The proposed method can be described by considering the example shown in Figs. 6–8. Let us assume that the scan chain length  $L$  is 3, the list of all the scan cells is  $S = \{SF0, SF1, SF2, \dots, SF14\}$ , and the number of deterministic patterns obtained through ATPG is 320. Each scan cell's specified bit usage can be calculated by parsing the contents of the deterministic test set. In Fig. 6(a), the numbers of specified-1 bits, specified-0 bits, and don't-care bits used in

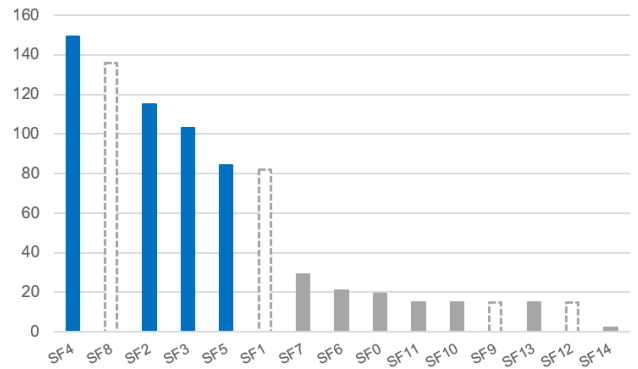


FIGURE 7. Distribution of specified bit usage for each scan cell in the updated list  $S$ .

Cell name	# 1s	# 0s	# Xs	# Specified bits
SF0	6	13	301	19
SF1	43	39	238	82
SF2	18	97	205	115
SF3	97	6	217	103
SF4	45	104	171	149
SF5	62	22	236	84
SF6	15	6	299	21
SF7	23	6	291	29
SF8	67	69	184	136
SF9	8	7	305	15
SF10	4	11	305	15
SF11	3	12	305	15
SF12	6	9	305	15
SF13	2	13	305	15
SF14	2	0	318	2

(a)

Cell name	# 1s	# 0s	# Xs	Weight	Bias type
SF4	45	104	171	0.30	b2
<b>SF8</b>	<b>67</b>	<b>69</b>	<b>184</b>	<b>0.49</b>	<b>b3</b>
SF2	18	97	205	0.16	b1
SF3	97	6	217	0.94	b1
SF5	62	22	236	0.74	b2
<b>SF1</b>	<b>43</b>	<b>39</b>	<b>238</b>	<b>0.52</b>	<b>b3</b>
SF7	23	6	291	0.79	b1
SF6	15	6	299	0.71	b2
SF0	6	13	301	0.32	b2
SF11	3	12	305	0.20	b1
SF10	4	11	305	0.27	b2
<b>SF9</b>	<b>8</b>	<b>7</b>	<b>305</b>	<b>0.53</b>	<b>b3</b>
SF13	2	13	305	0.13	b1
<b>SF12</b>	<b>6</b>	<b>9</b>	<b>305</b>	<b>0.40</b>	<b>b3</b>
SF14	2	0	318	1.00	b1

(b)

FIGURE 6. List of scan cells,  $S$ , (a) Based on alphanumeric order (b) Based on the descending order by specified bit usage.

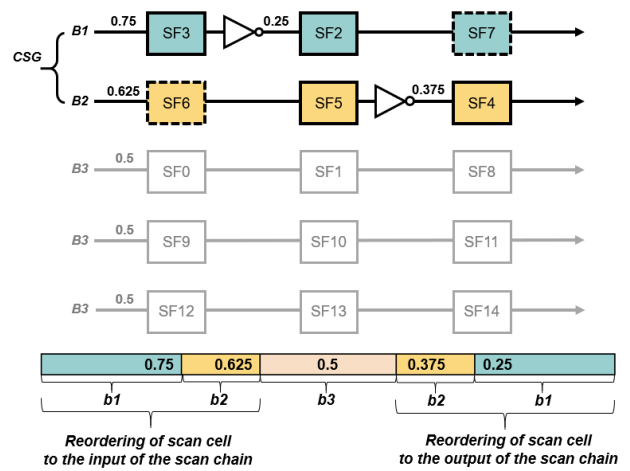


FIGURE 8. Example of weight inversion-based scan cell reordering in Phase 1.

SF0 are 6, 13, and 301, respectively. In this case, the specified bit usage can be found by subtracting the number of don't-care bits from the total number of patterns. The specified bit usage of this scan cell is 19 in this way. As shown in Fig. 6(b), all scan elements of  $S$  are then sorted in descending order based on the specified bit usage; thus, the highest-ranking scan cell SF4 is placed at the top of the list. For the application of the corresponding biased random patterns, the overall range of the weights is divided into  $b1$  (0–0.25, 0.75–1),  $b2$  (0.26–0.375, 0.625–0.74), and  $b3$  (0.376–0.624). Here, the scan cells within the range of  $b3$ , called the unbiased group, are eliminated from  $S$  using (2). In this example, SF8, SF1, SF9, and SF12 (highlighted in red) are removed because of this reasoning.

Next, the appropriate number of candidate cells  $m$  is determined by referring to the histogram of the specified bit count distribution in the updated  $S$  (see Fig. 7). Since the scan cells with relatively large numbers of specified bits are SF4, SF2, SF3, and SF5, the number of candidate cells can be heuristically set to  $m = 4$ . The selected  $m$  scan cells are sequentially reordered toward the input or output of the

corresponding biased scan chain (see Fig. 8). For example, because SF2 belongs to group *b1*, which has a weight of 0.16, it can be placed starting from the output of scan chain *B1*, whereas SF3, which has a different weight of 0.94 in the same group, is reordered starting from the input of the same scan chain. After distributing the  $m$  scan cells, the algorithm checks the length of the biased scan chains to balance them with the rest of the scan chains. If the length of the biased scan chain is greater than  $L$ , it is split to match the length  $L$ . Otherwise, the scan chains are balanced by sequentially distributing the remaining elements of  $S$  in the same way until the scan chain length reaches  $L$ . In Fig. 8, because the biased scan chain length is shorter than  $L$ , the remainder of the scan cells (excluding the  $m$  selected elements) are sequentially reordered in the remaining space (indicated by the dashed box) in the same manner. Finally, the inverter cells are inserted between the scan cells with inverted weight ranges, allowing different types of biased random patterns to be applied to a few scan chains. The group of biased scan chains formed thus far is called the *CSG*, and it is always active during the CUT test. The remaining nine scan cells are used to reduce the switching activity in the next phases (Sections III-B and III-C).

## B. PHASE 2: POWER-EFFICIENT SCAN GROUPING

Fig. 9 presents the algorithm of Phase 2. The algorithm is utilized to reduce the average switching activity while preserving the fault coverage improved in the previous phase. This is achieved by grouping the scan cells that are not distributed to the *CSG* into several *PSGs*, which are sequentially disabled. This algorithm is run based on a gate-level netlist (optimized in Phase 1). This netlist has an improved fault coverage gradient because a few scan cells are already placed into the biased *CSG*. In addition to the netlist, the input configurations (the scan chain length  $L$ , total number of patterns  $T$ , pattern count  $C$ , and user threshold value  $THR$ ) should be determined before performing the following steps.

- 1) First, a base  $C$  is assigned to a variable  $k$ , where  $k < T$ , to sequentially disable the remaining scan groups (except for the biased scan group) for each  $C$ . Additionally, a list of scan cells  $S$  is created to identify the candidate cells. Here, the scan elements previously reordered in the *CSG* should be excluded from  $S$ .
- 2) The detected fault list  $D_T$  is obtained by running fault simulation on  $T$  random patterns.
- 3) The detected sub-fault list  $D_k$  is extracted by simulating  $k$  random patterns ( $k$  is obtained in Step 1). Then, by eliminating the faults of  $D_k$  from those of  $D_T$ , a new list  $D_{new}$  is generated, which is detected in the duration from pattern  $k$  to pattern  $(T - 1)$ . Next, a deterministic test set is produced through the ATPG process by using  $D_{new}$ .
- 4) The following five sub-steps are performed using the new deterministic pattern set generated in Step 3 to select candidate cells and reorder them into the scan group  $PSG_k$ . First, the list  $S$  created in Step 1 is copied

### Algorithm 2 Power-efficient Scan Grouping

Prepare the netlist with scan chain length  $L$  (created in Phase 1), total number of patterns  $T$ , pattern count  $C$ , and user-threshold value  $THR$

```

1: while ( $THR < RD_{max}$ ) do
2:    $k \leftarrow C$ ;  $S \leftarrow S - S_{csg}$ ;
3:   Simulate  $T$  random patterns to obtain  $D_T$ 
4:   while ( $k < T$ ) do
5:     Simulate  $k$  random patterns to obtain  $D_k$ 
6:      $D_{new} \leftarrow D_T - D_k$ ;
7:     Generate a deterministic test set on  $D_{new}$ 
8:      $S_{tmp} \leftarrow S$ ;
9:     Calculate the weight of each element in  $S_{tmp}$ 
10:    Remove the elements ( $0 < W < 1$ )
11:    Sort  $S_{tmp}$  in descending order by  $W$ 
12:    if ( $sizeof(S_{tmp}) \geq L$ ) then
13:      Sequentially reorder the elements into the  $PSG_k$ 
14:      Insert an inverter between scan cells with inverted weights
15:    else
16:      Remove the elements of  $S_{tmp}$ 
17:    end if
18:     $S \leftarrow S - S_{psg}$ ;
19:     $k \leftarrow k + C$ ;
20:  end while
21: Calculate  $RD_{max}$  for the next condition of  $C/2$ 
22:  $C \leftarrow C/2$ ;
23: end while

```

FIGURE 9. Algorithm for power-efficient scan grouping for each constant pattern count.

to a temporary list labeled  $S_{tmp}$ . Second, the weights for all elements of  $S_{tmp}$  are calculated using (2). Third, the elements with weights of 0 or 1 are retained and the others are removed from  $S_{tmp}$ . The remaining scan elements are then sorted in descending order by weight. Fourth, the size of  $S_{tmp}$  is checked. If it exceeds the user-defined chain length  $L$ , the remaining scan elements are sequentially reordered into the  $PSG_k$ . Then an inverter cell is inserted at the location where the value of the weight is inverted. Otherwise, all elements in  $S_{tmp}$  are eliminated and the current step (Step 4) is skipped. Finally, list  $S$  is updated by removing scan elements reordered into  $PSG_k$ .

- 5) In this step,  $k$  is increased by  $C$ , and Steps 3–5 are repeated until  $k$  is greater than  $T$ . When  $C$  has a larger value than  $T$ , the algorithm proceeds to Step 6.
- 6) In the step to follow, determine whether to continue carrying out Steps 1–5 after dividing  $C$  in half. This is accomplished by comparing the  $THR$  to  $RD_{max}$ , where  $RD_{max}$  is the parameter that predicts the maximum reduction rate of the average switching activity for the next  $C/2$  condition. If  $RD_{max}$  is greater than  $THR$ , then the algorithm returns to Step 1 after decreasing  $C$  by half. Otherwise, the resultant netlist is returned and all procedures are promptly terminated.

*Example 2:* Figs. 10–11 show an example of Step 4, which is a key step of the proposed algorithm. First, let us assume that the scan chain length  $L$  is 3, which is identical to the

value in Phase 1, and the input configurations ( $T = 30K$  and  $C = 10K$ ) are determined. The list  $S_{tmp}$  copied from Step 1 is shown in Fig. 10(a). As can be observed, the six scan elements SF2, SF3, SF4, SF5, SF6, and SF7 are not present in this list because they have already been reordered into the CSG (in Section III-A). Additionally, since a new deterministic pattern is generated in Step 3, the number of specified bits and don't-care bits used in each scan cell are different from those in Phase 1. The weight of each element in  $S_{tmp}$  is calculated using (2). The scan elements SF1, SF8, and SF10, whose weight is not 0 or 1 (indicated in red), are eliminated from the list. Next, the remaining elements are

Cell name	# 1s	# 0s	# Xs	Weight
SF0	0	5	326	0.00
<b>SF1</b>	<b>32</b>	<b>47</b>	<b>252</b>	<b>0.41</b>
<b>SF8</b>	<b>45</b>	<b>33</b>	<b>253</b>	<b>0.58</b>
SF9	0	9	322	0.00
<b>SF10</b>	<b>3</b>	<b>4</b>	<b>324</b>	<b>0.43</b>
SF11	0	0	331	0.00
SF12	0	22	309	0.00
SF13	5	0	326	1.00
SF14	13	0	318	1.00

(a)

Cell name	# 1s	# 0s	# Xs	Weight
SF14	13	0	318	1.00
SF13	5	0	326	1.00
SF12	0	22	309	0.00
SF9	0	9	322	0.00
SF0	0	5	326	0.00
SF11	0	0	331	0.00

(b)

FIGURE 10. List of scan cells,  $S_{tmp}$ . (a) Based on alphanumeric order (b) Based on the descending order by weight.

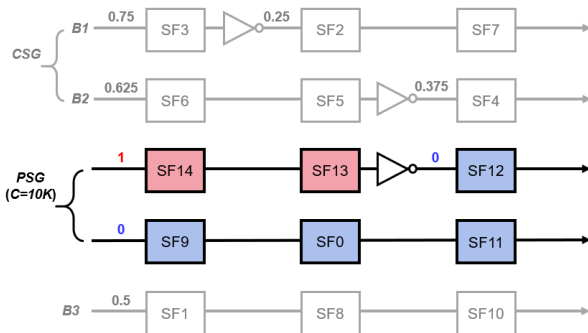


FIGURE 11. Example of weight inversion-based scan cell reordering in Phase 2.

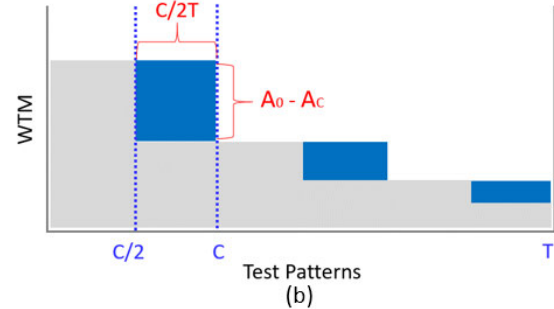
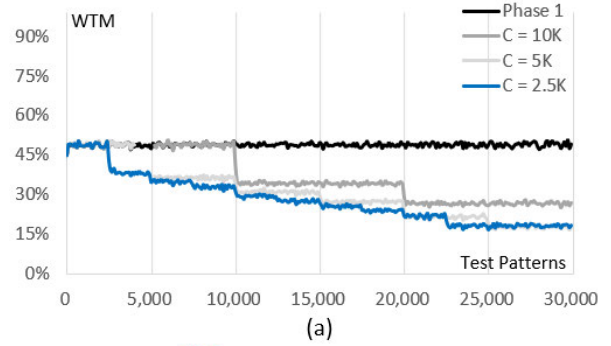


FIGURE 12. (a) WTM for different  $C$  (10K, 5K, and 2.5K) using b19 circuit and (b) Example of  $RD_{max}$  calculation.

sorted in descending order by weight, as shown in Fig. 10(b). Since the size of the updated list exceeds  $L$ , these scan cells are sequentially reordered from the upper scan elements to the scan group  $PSG_{10K}$  (Fig. 11). The number of scan chains in  $PSG_{10K}$  can be defined as the integer value of the size of the updated  $S_{tmp}$  divided by  $L$ . Finally, an inverter cell is inserted at the position of the inverted weight. After this process is completed, a constant value of 0 or 1, which is equal to the weight, is delivered to  $PSG_{10K}$  between patterns  $10K$  and  $(T - 1)$ . The procedure described so far is repeated by increasing  $C$  until  $C > T$ .

Fig. 12(a) shows the validation results for b19 circuit using the proposed method. As shown in the experimental results, the average switching activity gradually decreases as the constant  $C$  is reduced by half (10K, 5K, and 2.5K). This implies that  $C$  converges to the optimal state in terms of the mean WTM. For example, if  $C = 2.5K$  is determined, then the average toggling level may be further minimized compared with  $C = 5K$ .

However, the  $C$  directly affects the computation time of the inner loop (Steps 3–5). If  $C$  is indefinitely divided, a major problem arises from an increase in the algorithm's iteration count and the test control data required for disabling the scan groups. Let us assume that the total number of patterns  $T = 30K$  and the  $C$  sequence (10K, 5K, 2.5K, and 1.25K) are determined. In this case, the iterations (Steps 3–5) increases in the order of 3, 6, 12, and 24. Hence, the total number of iterations is 45, which can lead to a long computational time.

Equation (3) is proposed to alleviate these negative effects on algorithm computation time. It enables to eliminate many unnecessary iterations by predicting the maximum reduction



**Algorithm 3** Scan Group Scheduling

Prepare the netlist (created in Phase 2), total number of patterns  $T$ , and pattern count  $C$  (determined in Phase 2)

```

1: Create the duration list  $G = \{g_c, g_{2c}, g_{3c}, \dots, g_{nc}\}$ 
2: Initialize  $i \leftarrow 1$ ;

3: # Iteratively select the replaceable time slots and shift the enable
   # time
4: while ( $G$  is not empty) do
5:   Compute max WTM during  $g_c$  and assign it to  $max(g_c)$ 
6:   Find the time slot list  $r_{ic}$  with the minimum mean WTM
     (equivalent to the length of  $g_{ic}$ )
7:   Identify the scan group that is active only during  $g_{ic}$ 
8:   Shift the enable time of this scan group to the identified slots  $r_{ic}$ 
9:   Compute the max WTM during  $g_{nc}$  and assign it to  $max(g_{nc})$ 
10:  if ( $max(g_{ic}) > max(g_{nc})$ ) then
11:     $i \leftarrow i + 1$ ;
12:    continue;
13:  else
14:     $roll\_back()$ ;
15:    break;
16:  end if
17:end while

```

**FIGURE 13.** Algorithm for scan group scheduling.

rate for the mean WTM for the next  $C/2$  condition.

$$RD_{max} = \sum_{k=1}^{T/C} \frac{C(A_{C(k-1)} - A_{Ck})}{2T} \quad (3)$$

where  $RD_{max}$  is the parameter that predicts how much to minimize the average switching activity for the next  $C/2$  condition. Initially,  $T$  and  $C$  are required to obtain  $RD_{max}$ . Furthermore,  $A_i$  is defined as (# of active scan chains / total # of scan chains) at the  $i$ th pattern. If all scan chains are enabled at this pattern, then  $A_i$  is equal to one.  $C/2T$  is the ratio of  $C/2$  over the total number of patterns  $T$ . Therefore, each portion (blue area in Fig. 12(b)) of the total area is incrementally obtained and summed until  $k$  equals  $T/C$ .

$$WTM_{C/2} \geq WTM_C - (WTM_{init} \times RD_{max}) \quad (4)$$

The above equation (4) can be used to obtain the expected mean WTM for  $C/2$ .  $WTM_{init}$  denotes the initial measure of the average switching activity of all the scan cells in Phase 1.  $WTM_C$  and  $WTM_{C/2}$  indicate the results of the average switching activity for  $C$  in the current stage and  $C/2$  in the next stage, respectively.

**C. PHASE 3: SCAN GROUP SCHEDULING**

Fig. 13 illustrates the algorithm of Phase 3. This algorithm aims to reduce the peak switching activity without impacting the average switching activity improved in Phase 2. As mentioned in Section III-B, the procedure of Phase 2 reduces the average toggling level while maintaining the enhanced fault coverage. However, the application of the technique for gradually disabling the scan groups ( $PSGs$ ) causes all scan groups to be enabled initially, which can lead to a high switching activity of approximately 50% for the whole scan cells. To avoid this excessive toggling level, the proposed algorithm

schedules the enable time of each  $PSG$  by shifting the time slot with the highest WTM to the lowest time slot. Similarly, this P3 algorithm is executed based on the gate-level netlist (optimized in Phase 2) and the input configurations ( $T$  and  $C$ ) determined in Phase 2. For these inputs, the main procedures for shifting the enable time are as follows:

- 1)  $T$  is divided on the basis of  $C$  and a list of duration  $G = \{g_c, g_{2c}, g_{3c}, \dots, g_{nc}\}$  is created. The element  $g_i$  is defined as the duration from patterns 0 to  $(i - 1)$ . For example, if it has been determined that  $T = 30K$  and  $C = 10K$ , list  $G$  consists of three elements:  $g_{10K}$ ,  $g_{20K}$ , and  $g_{30K}$ . Here,  $g_{10K}$  is the duration from patterns 0 to  $(10K - 1)$ . After the list  $G$  is created, the following steps (Steps 2–6) are iteratively performed starting from the first element  $g_c$ .
- 2) Power calculation is performed to obtain max WTM during  $g_c$ . In other words, the maximum value of the toggling level is acquired among the results progressed from patterns 0 to  $(C - 1)$ , and it is assigned to  $max(g_c)$ .
- 3) Given  $g_c$ , one can find the replaceable time slot  $r_c$  that has the same length as duration  $g_c$  while having the lowest mean WTM for all periods. Here, max WTM (obtained during  $r_c$ ) should be lower than  $g_c$ .
- 4) The scan group  $PSG_c$  which is activated only in the duration  $g_c$  is identified. Then, the enable time of this  $PSG_c$  is shifted to  $r_c$  to reduce max WTM further.
- 5) After scheduling the enable time, max WTM is recalculated for the entire duration and the result is assigned to  $max(g_{nc})$ .
- 6) Finally, the results of Steps 2 and 5 are compared. If  $max(g_{nc})$  is less than  $max(g_c)$ , then Steps 2–6 are repeated for the next element of  $G$ . Otherwise, rollback to past Step 3 and terminate the algorithm.

*Example 3:* Fig. 14 presents a typical example of the algorithm in Phase 3. This example progresses from Figs. 14(a)–14(d). Initially, let us assume that the netlist obtained from Phase 2 and the input configurations ( $T = 30K$  and  $C = 5K$ ) are set for running the proposed algorithm. Based on this configuration, a list of durations  $G = \{g_{5K}, g_{10K}, g_{15K}, g_{20K}, g_{25K}, g_{30K}\}$  can be produced. In Fig. 14(a), each scan group, turn by turn, is deactivated in the order of the elements of  $G$ , resulting in a gradual reduction in the mean WTM. However, because max WTM is still relatively high during  $g_{5K}$ , a replaceable time slot  $r_{5K}$  (indicated by the green line) that is capable of being scheduled is first searched to check if there is room for reducing max WTM. Here,  $r_{5K}$  should have not only the lowest mean WTM in the entire interval but also the same duration as  $g_{5K}$ . The scan group  $PSG_{5K}$  (indicated by the dashed box), which is active only during  $g_{5K}$  and inactive in the other durations, is identified. The enable time of  $PSG_{5K}$  is then scheduled to  $r_{5K}$ . Fig. 14(b) presents the results of the above scheduling. After this procedure is completed, max WTM is computed for all periods (indicated by the blue line) and compared with max WTM obtained before scheduling (indicated by the red line). As long as the blue line is below

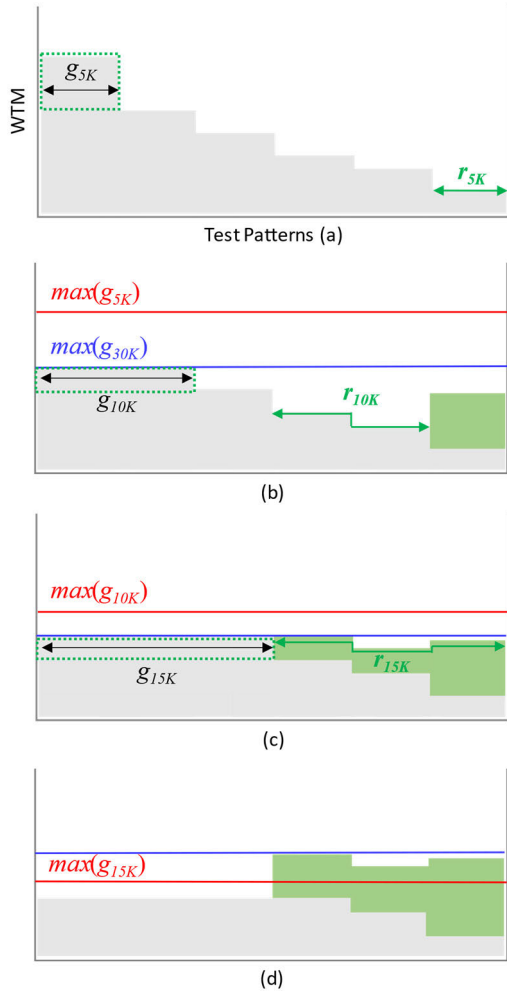


FIGURE 14. Example of the proposed algorithm in Phase 3. Target Input elements are (a)  $g_{5K}$  (b)  $g_{10K}$  (c)  $g_{15K}$  (d)  $g_{20K}$ .

the red line, the above procedures are repeated continuously starting from the next element  $g_{10K}$ , as shown in Fig. 14(c). However, as the procedure continues, max WTM can have a larger value than that in the previous stage (Fig. 14(d)). In such a case, the scheduling is rolled back to the previous stage and the algorithm is terminated.

D. PHYSICAL-AWARE SCAN REORDERING

The objective of this phase is to improve the wire length of the scan chains with minimal impact on the factors that have already been enhanced in Phases 1–3. Physical-aware scan reordering can be performed at the post-layout stage with the help of the EDA tool. However, because the scan ordering constraints are applied to most scan elements in Phases 1 and 2, it is difficult to further reorder these scan cells at a later stage. In other words, there is little room for further routing overhead improvement, even if physical-aware scan reordering is conducted incrementally.

Fig. 15 presents an example of a scan chain structure after the completion of Phases 1 and 2. This structure comprises a biased scan group (CSG) and  $N$  unbiased scan groups (PSGs). Here, the CSG is determined in Phase 1, and the multiple PSGs are frozen in Phase 2. Furthermore, since each scan group includes inverters for propagating weight inversion signals, the scan cells with inverted weights can be grouped into a few scan chains.

The above observation indicates that the wire length of each scan group can be improved by independently reordering the upper scan cells before the inverter (red dashed line) or the lower scan cells after the inverter (blue dashed line). Because the scan elements of each upper or lower group within the scan group have identical weights, it is expected that the routing overhead can be reduced with minimal impact on the fault coverage, even if physical-aware scan reordering is performed.

The intermediate SCANDEF file must be extracted from the previous Phase 2 or Phase 3 to support the above explanation. Because this file contains the ordering constraint information for each scan cell, the following steps are conducted sequentially to reflect the proposed scan chain structure.

- 1) The scan chains with inverter are identified and partitioned into sub-scan chains based on their inverter.
- 2) The scan chains in each scan group, which have an equivalent weight range, are relabeled to have the identical partition name.
- 3) After the partition name of each scan chain is modified, the resultant SCANDEF file is delivered to the downstream physical-layout tool to reduce the routing overhead further.

The results of this proposed method are presented in the experimental results section.

IV. HARDWARE IMPLEMENTATION

Fig. 16 presents the proposed logic BIST architecture. As can be observed, the architecture consists of three major blocks: a conventional  $n$ -bit linear feedback shift register (LFSR),

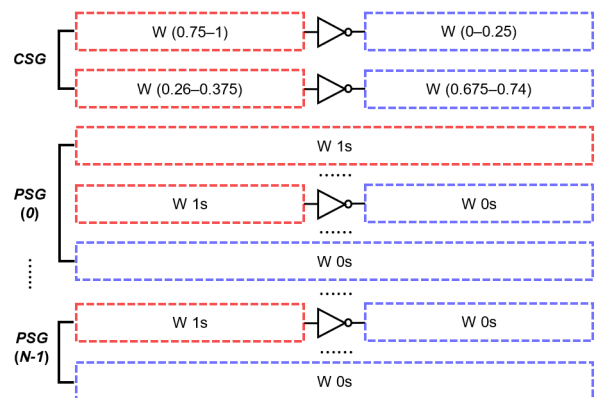


FIGURE 15. Example of scan cell ordering after Phases 1 and 2 are completed.

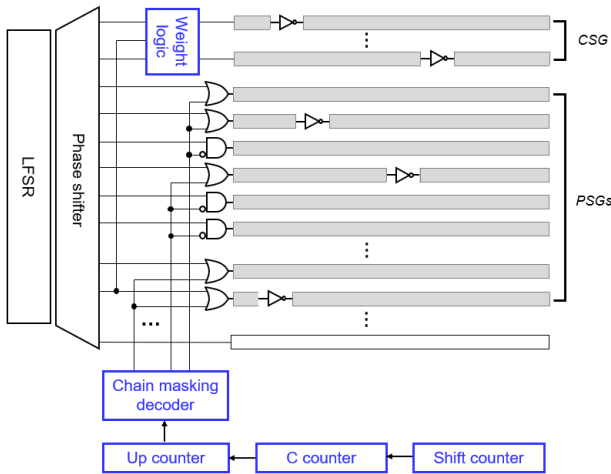


FIGURE 16. Proposed logic BIST architecture.

which generates unbiased pseudorandom patterns, a phase shifter that eliminates linear dependencies, and additional blue colored circuits (*weight-logic*, *shift-counter*, *C-counter*, *up-counter*, and *chain-masking-decoder*) to implement our new scheme.

The *weight-logic* is placed between the phase shifter and the biased scan group (*CSG*). This circuit is configured as combinational gates to provide two types of biased random patterns using unbiased signals from the phase shifter. Additionally, appropriate constant values are sequentially fed into the remaining *PSGs* through two-input AND or OR gates to reduce the switching activity. These constant values are controlled by the following procedure. First, the *shift-counter* iteratively counts the longest scan chain length. In other words, this counter increases incrementally before eventually obtaining the value of the longest chain length. It is then initialized to zero and restarts counting. During this process, the *C-counter* is updated simultaneously with the initialization of the *shift-counter*. Hence, the role of this counter is similar to that of a pattern counter that updates the values per pattern. However, it differs from the typical pattern counter as it adopts its method by repeatedly counting numbers based on *C*. Whenever the *C-counter* reaches a user-defined constant value, it propagates a *carry* to the *up-counter*, allowing it to produce the required input values for the *chain-masking-decoder* (*CMD*) sequentially. The *CMD* is the conventional decoder circuit that converts coded inputs into coded outputs through combinational gates. Because all the masking values are pre-assigned in the decoder, the outputs of the decoder can be changed (or updated) according to the state of the *up-counter*. This means that binary codes allocated to the *CMD* can be sequentially propagated to each *PSG* per *C*.

Fig. 17 shows an example of assigning binary codes to the *CMD* after Phases 1–3 are completed. Let us assume  $T = 30K$  and  $C = 5K$  are determined and that the total number of *PSGs* is five (from  $PSG_c$  to  $PSG_{5c}$ ). The *CSG* is always enabled for all periods to reduce the negative impact on the fault coverage.

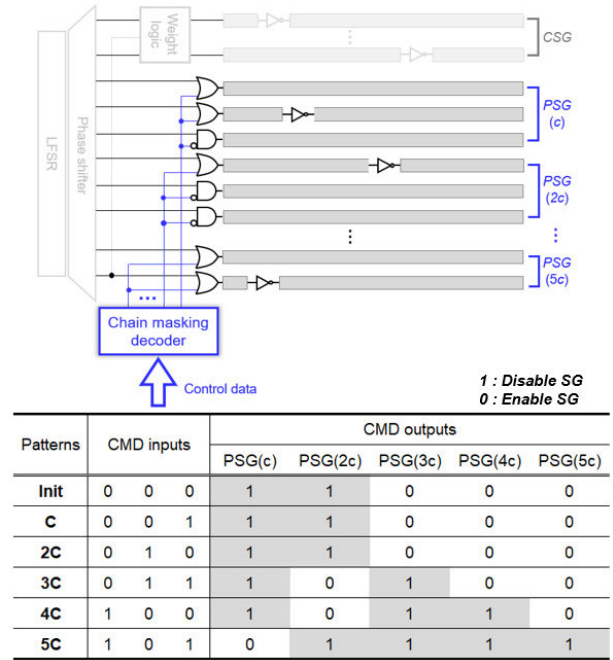


FIGURE 17. Example of truth table assigned to the chain-masking-decoder.

Therefore, control data are not required for disabling the *CSG*. In this example, since the *CMD* sequentially disables or enables *PSGs* for each *C* in the entire test pattern  $T$ , a total of  $T/C = 6$  sets of control data are required. In this case, the total size of the required control data is 30 bits, which can be obtained by multiplying by the total number of *PSGs* and the number of sets of binary codes.

$$CMD_{capacity} \leq \frac{T}{C} \times \left( \frac{T}{C} - 1 \right) \quad (5)$$

Consequently, the truth table size required for scan group disabling can be defined as  $\#PSG \times (T/C)$ , where  $\#PSG$  denotes the total number of *PSGs* and  $T/C$  indicates the number of sets of binary codes. As explained in Section III-B, the maximum  $\#PSG$  is equal to  $(T/C - 1)$ . Therefore, for a given  $T$  and  $C$ , the *CMD* capacity requirement can be obtained using (5).

## V. EXPERIMENTAL RESULTS

In the experiments, the Synopsys EDA toolchain was employed to evaluate each of our algorithms. TetraMAX was used for generating a deterministic pattern set, running fault simulation, and computing the toggling level during scan shift-in and shift-out. Logic synthesis, DFT-insertion, and physical-aware scan reordering were conducted using Design Compiler Graphical (DCG). All procedures were performed using the Synopsys SEAD 32 nm standard library. Additionally, a 40-bit external LFSR with a phase-shifter was implemented in the proposed hardware architecture. The polynomial was  $x^{40} + x^{21} + x^{19} + x^2 + 1$ . For reference, test points were not applied for a fair comparison.

**TABLE 2.** Profile of each of the benchmark circuits.

Benchmark	CUT	Gates	Scan cells	Chain len. ( $L$ )	A-R (%)
ITC'99	b17	25.5K	1.4K	57	14.8
	b18	72.2K	3.3K	133	14.1
	b19	142.3K	6.6K	266	15.1
OpenCores	or1200	48.2K	3.5K	139	25.5

The profile of each circuit is listed in Table 2. The three largest ITC'99 benchmark circuits and an OpenCores benchmark circuit were considered for our experiments. In particular, the ISCAS benchmark circuits were excluded because their gate count and the portion of RPR faults are relatively low. The first and second columns present the name of each benchmark and circuit, and the third column gives the values calculated by dividing the area of all gates by the area of a two-input NAND gate. The fourth and fifth columns indicate the total number of scan cells and the length of the longest scan chain, respectively. The scan chains are balanced according to the length  $L$ . The last column shows the difference between the fault coverage obtained by the ATPG process and that acquired using 30K pseudorandom patterns. All circuits exhibit an average decrease of 17% in terms of fault coverage compared with the ATPG process. Additionally, the scan cells of each circuit were stitched in alphanumeric order using the conventional method.

Hereafter, for brevity, we denote Phase 1, 2, and 3 by P1, P2, and P3, respectively, and the experimental results are detailed for each phase sequentially.

### A. EVALUATION OF COVERAGE-EFFICIENT SCAN GROUPING

Fig. 18 presents the distribution of specified bit usage for finding candidate cells. The histograms were obtained by conducting the following three procedures: 1) extracting a deterministic pattern set for each circuit, 2) sorting the scan cells in descending order by specified bit usage, and 3) removing the scan cells with the unbiased weight range of  $b3$  (0.376–0.624). Fig. 18 reveals that scan cells with high specified bit usage are roughly concentrated in the 8% range for the b17, b18, and b19 circuits. For the OR1200 circuit, most of the scan cells using specified bits are clustered within the 8% range (for reference, a light gray area indicates the distribution of scan cells with the range of  $b3$ ). Based on the above observation, the candidate cells were heuristically determined to be  $m = 8\%$  and were reordered into a dedicated CSG. Additionally, the remaining 92% of the scan cells were used to further reduce the WTM in P2 and P3.

Table 3 lists the result for each factor after P1 is completed. The “FC” rows refer to the fault coverage achieved with 30K random patterns. When the biased random patterns

**TABLE 3.** Comparison of outcomes between the conventional LBIST and Phase 1 ( $m = 8\%$ ).

		b17	b18	b19	or1200
Conv-LBIST	FC (%)	85.01	85.78	84.79	74.45
	Mean WTM (%)	49.44	49.27	49.28	48.84
	Max WTM (%)	55.56	53.18	53.17	52.14
P1	FC (%)	92.72	91.35	90.45	92.87
	$\Delta$ FC (%)	<b>8.32</b>	<b>6.10</b>	<b>6.26</b>	<b>19.83</b>
	Mean WTM (%)	48.74	48.81	48.79	48.82
	Max WTM (%)	55.59	52.93	51.41	52.97

are fed to a small number of candidate cells ( $m = 8\%$ ), the fault coverage improvement rate shows a sharp increase (average 10.13%) compared with the conventional LBIST. Additionally, the “Mean WTM” and “Max WTM” rows indicate the average and maximum toggling levels for the scan cells during the scan shift-in and shift-out, respectively. The row labeled “Mean WTM” of P1 shows that the average toggling level slightly decreases when biased random patterns are applied to a few candidate cells.

To demonstrate whether the  $m$  determined by the P1 algorithm is appropriate, we applied biased random patterns to the top 8%, 16%, and 24% of all the scan cells and then checked the fault coverage improvement rate in each case. As shown in Fig. 19, when biased random patterns are applied to more scan cells ( $m = 16\%$ ,  $m = 24\%$ ), the average improvement rate is less than 1% compared with the identified case with  $m = 8\%$ . This implies that even if only a few of the top scan cells selected by the P1 algorithm are reordered into the CSG, a similar fault coverage gain can be achieved.

### B. EVALUATION OF POWER-EFFICIENT SCAN GROUPING

To evaluate the P2 algorithm, we employed the input configurations ( $T = 30K$ ,  $C = 10K$ , and  $THR = 0.05$ ) and the gate-level netlist acquired in P1 ( $m = 8\%$ ). For reference, if the initial  $C$  is set to be too small compared with  $T$ , the amount of memory capacity and computation time required may be excessive; therefore, we set the initial  $C$  to be approximately  $T/3$ . Additionally,  $THR$  was set to 0.05 so that the algorithm in P2 could promptly stop when the predicted maximum improvement rate for the average toggling level of the next  $C$  was below 5%. As shown in Fig. 20(a), the P2 algorithm checks the  $RD_{max}$  parameter to identify the appropriate  $C$ . Since  $RD_{max}$  calculated at  $C = T/12$  is less than  $THR$ , the algorithm returns the resultant netlist for  $T/12$  and stops. It is worth noting that this algorithm can eliminate many unnecessary iterations and save computation time by not performing the following steps for  $C = T/24$ . Furthermore, power calculation for each successive  $C$  is not

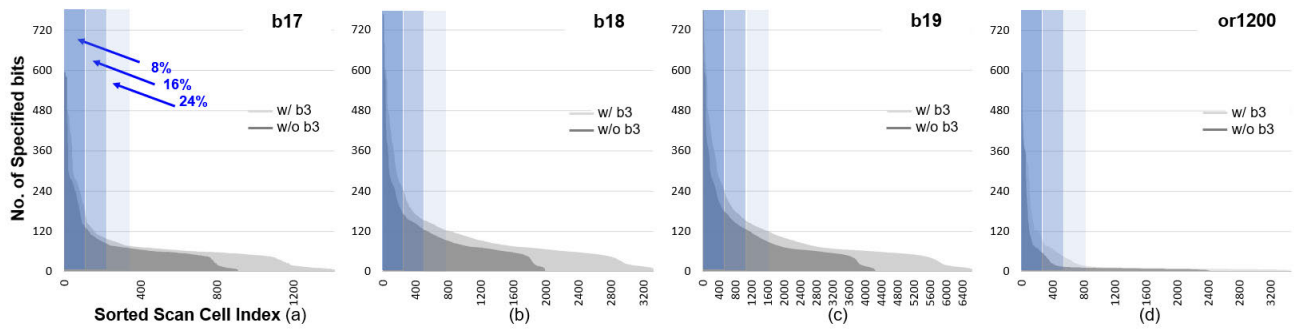


FIGURE 18. Distribution of scan cells with high specified bit usage in the biased and unbiased ranges for each benchmark circuit. (a) b17 (b) b18 (c) b19 (d) or1200.

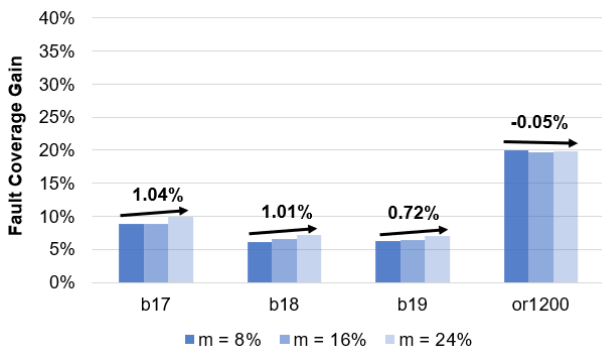


FIGURE 19. Experimental results for the fault coverage improvement rates for  $m = 8\%$ ,  $16\%$ , and  $24\%$ .

required to obtain  $RD_{max}$ . Instead, it is calculated simply using a few parameters during algorithm iterations, which can significantly reduce computation time in finding the solution. Fig. 20(b) shows the correlation between the actual and predicted WTM results. The red line represents the result for the actual mean WTM after P2 algorithm is executed and the blue line is the WTM result predicted by using (4) (in Section III-B). Each dot represents the average value computed from all circuits. It can be seen that the red line is always above the blue line. Hence, because the actual power reduction rate is always less than the predicted one, if the predicted mean WTM (blue line) is worse than expected, then the computation time and the number of iterations can be reduced by promptly terminating the algorithm.

Table 4 lists the fault coverage, mean WTM, and max WTM for ( $C = T/12$ ), which is obtained by P2 algorithm. The row “Mean WTM” of P2 shows that the average toggling level decreased by approximately 49.21% relative to P1 when disabling the scan groups ( $PSGs$ ) at every  $T/12$  interval. Additionally, the row labeled “FC” of P2 shows that the fault coverage of the b17, b18, and b19 circuits is improved compared with P1. This observation suggests that RPR faults may be probabilistically detected by loading constant values equal to the weight of the scan cell for each constant interval. However, for the OR1200 circuit, the number of hard-to-detect faults was relatively small after P1, and even if constant values equal to the scan cell’s weight are fed, it is difficult to detect additional faults. The row labeled “Max WTM”

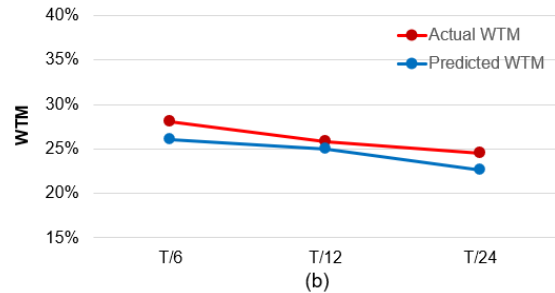
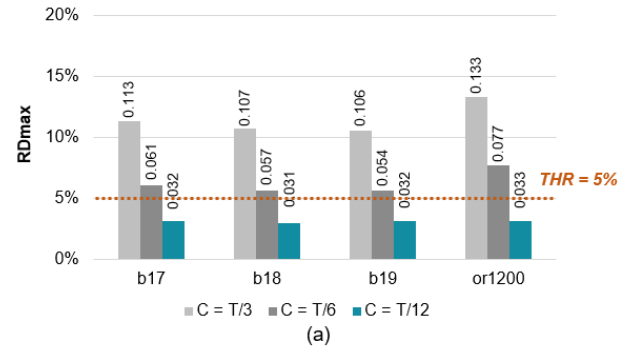


FIGURE 20. Evaluation results of Phase 2 algorithm. (a) Variation of  $RD_{max}$  with a decrease in  $C$ . (b) Correlation between actual and predicted WTM results for different  $C$ .

TABLE 4. Comparison of outcomes between Phase 1 ( $m = 8\%$ ) and Phase 2 ( $C = T/12$ ).

		b17	b18	b19	or1200
P1	FC (%)	92.72	91.35	90.45	92.87
	Mean WTM (%)	48.74	48.81	48.79	48.82
	Max WTM (%)	55.59	52.93	51.41	52.97
P2	FC (%)	95.11	93.22	92.46	92.02
	Mean WTM (%)	26.26	27.71	28.20	16.95
	$\Delta$ Mean WTM (%)	-46.12	-43.23	-42.20	-65.28
	Max WTM (%)	54.36	52.71	51.30	52.39

of P2 indicates that the maximum toggling level is still near 50%. This is because all the scan groups are enabled during the first  $C$ .

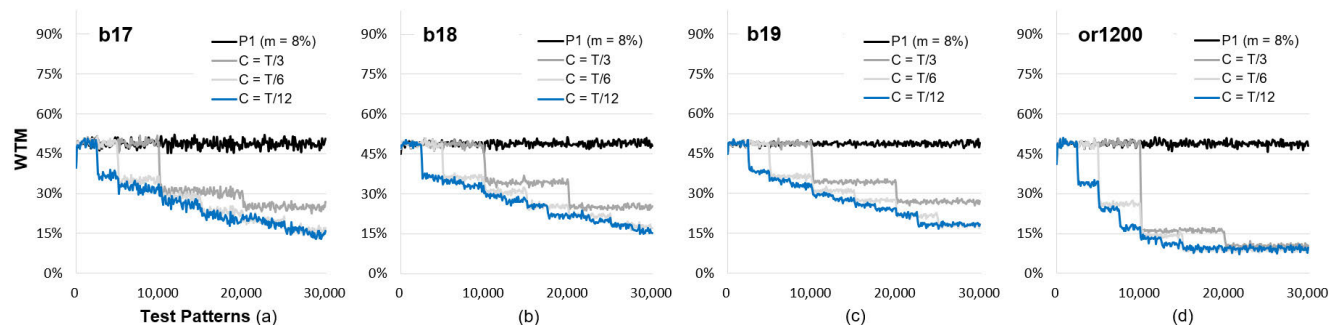
Furthermore, we performed experiments to demonstrate the trend of switching activity for  $C = 10K, 5K,$  and  $2.5K,$  which are equivalent to  $C = T/3, T/6,$  and  $T/12$  (see Fig. 21). For all circuits, the mean WTM is reduced as scan groups are gradually disabled at every successive  $C$ . Additionally, as  $C$  is halved, the mean WTM decreases more than the previous one. For instance, if initial  $C$  is determined to be  $10K,$  scan-group disabling cannot be performed between  $0$  and  $(10K - 1)$ . However, if  $C$  is halved, other scan cells with weights of  $0$  or  $1$  can be identified and disabled in this interval.

**C. EVALUATION OF SCAN GROUP SCHEDULING**

The P3 algorithm was evaluated using the gate-level netlist obtained from P2 ( $C = T/12$ ). Table 5 shows the actual results for each circuit after scheduling the enable time of  $PSGs$  using this algorithm. The ‘‘Mean WTM’’ row of P3 reports that average switching activity is maintained at a level similar to that in P2. The ‘‘Max WTM’’ row reveals that the average reduction rate in max WTM exceeded 41% compared with P2. The ‘‘FC’’ row of P3 gives that there is a slight difference in the results of fault coverage compared with P2. This is because the initial seed values used in each

**TABLE 5. Comparison of outcomes between Phase 2 ( $C = T/12$ ) and Phase 3.**

		b17	b18	b19	or1200
P2	FC (%)	95.11	93.22	92.46	92.02
	Mean WTM (%)	26.26	27.71	28.20	16.95
	Max WTM (%)	54.36	52.71	51.30	52.39
P3	FC (%)	94.42	92.44	91.85	91.63
	Mean WTM (%)	26.21	27.67	28.17	16.94
	Max WTM (%)	32.48	32.52	31.76	26.40
	$\Delta$ Max WTM (%)	<b>-40.25</b>	<b>-38.30</b>	<b>-38.09</b>	<b>-49.61</b>



**FIGURE 21. WTM results for different C. (a) b17 (b) b18 (c) b19 (d) or1200.**

**TABLE 6. Wire length results for the conventional method, ‘‘w/o proposed physical aware reordering,’’ and ‘‘w/ proposed physical-aware reordering.’’**

		b17	b18	b19	or1200
Conv-LBIST	FC (%)	82.62	86.61	84.32	74.57
	Wire-len. (um <sup>2</sup> )	286K	826K	1,616K	524K
w/o Prop.	FC (%)	94.62	92.49	91.72	91.62
	Wire-len. (um <sup>2</sup> )	359K	1,079K	2,125K	598K
	$\Delta$ Wire-len. (%)	<b>20.34</b>	<b>23.46</b>	<b>23.94</b>	<b>12.32</b>
w/ Prop.	FC (%)	93.96	92.42	91.78	90.91
	Wire-len. (um <sup>2</sup> )	313K	889K	1,733K	552K
	$\Delta$ Wire-len. (%)	<b>8.49</b>	<b>7.13</b>	<b>6.74</b>	<b>4.92</b>

$PSG$  in P2 change when the order of the  $PSG$  enable times is scheduled. However, it is only a minuscule difference from the fault coverage that had already improved in P1 and P2, and the P3 algorithm still shows significant improvement compared with the conventional LBIST. Additionally, Fig. 22 shows the changes in the switching activity of scan cells after the P3 algorithm is executed. As time slots with high switching activity are shifted to slots with low switching activity, the global maximum point of the WTM is reduced in comparison with P2.

**D. EVALUATION OF HARDWARE OVERHEAD**

Table 6 presents the results for the total wire length after physical-aware scan reordering. In this experiment, an aspect ratio of  $1.0$  was used for the core area, and cell utilization was  $60\%$ . The ‘‘Conv-LBIST’’ row represents the conventional physical-aware reordering method, in which scan cells are aggressively reordered without any limitation. The ‘‘w/o Prop.’’ row indicates the case where scan reordering is not performed for frozen scan elements through P1 and P2. The last ‘‘w/ Prop.’’ row shows the result after the constraints are

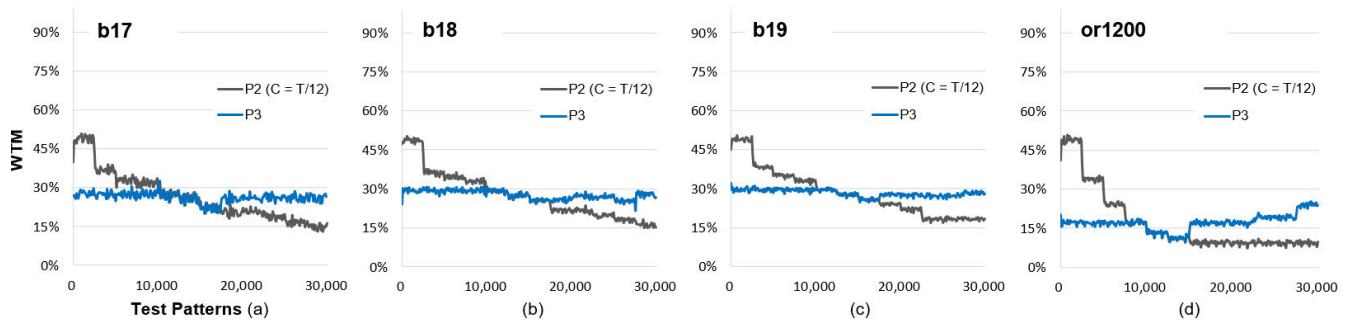


FIGURE 22. WTM result of Phase 3. (a) b17 (b) b18 (c) b19 (d) or1200.

eliminated between scan cells with identical weight range in the same scan group, based on the inverters. The alleviation of the hard ordering constraints reduces the total wire length in comparison to the second case. Additionally, the fault coverage is maintained similarly to the second case. However, as scan reordering can only be applied to scan cells with identical weights in the same scan group, this table indicates that wire length increased to some extent compared with the first case.

The expected maximum size of the control data for scan group disabling was calculated using (5) (Section IV). Because  $T = 30K$  and  $C = 2.5K$  were determined in the previous P2 and P3, the maximum *CMD* capacity was predicted to be 132 bits. However, for each circuit, there existed intervals where scan grouping was not performed; therefore, an average of 105 binary bits, which was lower than the expected value, was assigned to the *CMD*. Also, the average gate count of the generated *CMD* was equivalent to 32 two-input NAND gates, which can be considered as an acceptable level of hardware overhead.

### E. COMPARISON WITH PREVIOUS METHODS

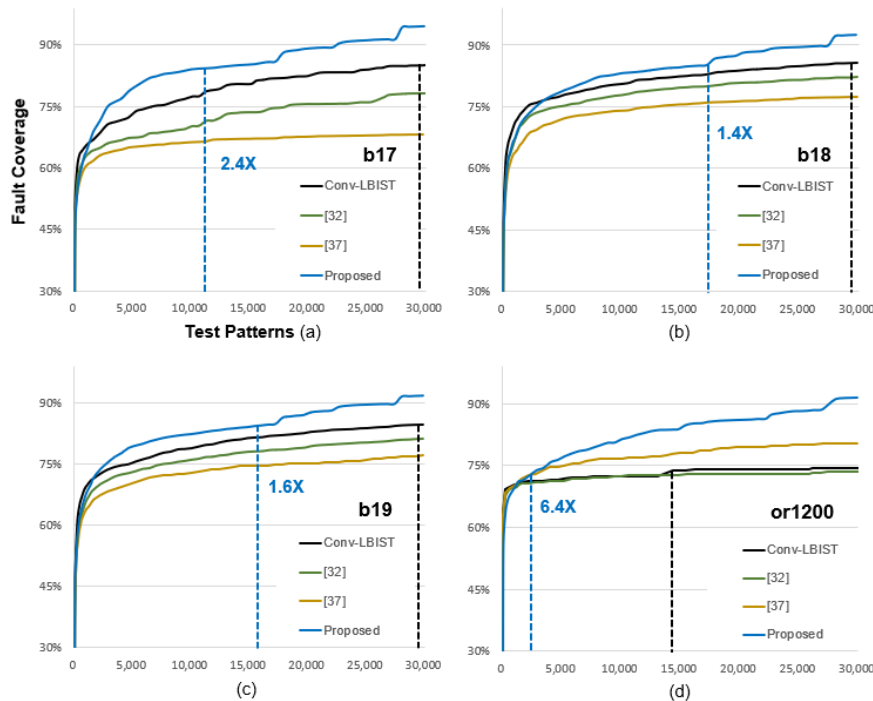
Table 7 shows comparisons of fault coverage and WTM among four cases: the conventional LBIST, [32], [37], and the proposed method (in Section V-C). First, the solution in [32] was designed by setting the configuration of the control register as (25-13), implying that 13 adjacent binary-one bits are loaded to a 25-bit control register to generate a toggling level similar to that in the proposed method. Additionally, for a fair comparison, the scan chain structure in the circuit, the size of the LFSR, and the phase-shifter structure used for evaluation were implemented identically. Similarly, the solution in [37], namely the LCA, was also evaluated under identical conditions. This solution is capable of producing low transition patterns close to 25% toggling level on average, similar to the proposed method. In the table, compared with [32] and [37], the proposed method results in similar enhancement in terms of mean and max WTM and exhibits the best results with respect to fault coverage based on 30K random patterns. In other words, based on an average power improvement rate of approximately 50%, average fault

TABLE 7. Comparison of fault coverage and WTM among the conventional LBIST, [32], [37], and the proposed method.

		b17	b18	b19	or1200
Conv-LBIST	FC (%)	85.01	85.78	84.79	74.45
	Mean WTM (%)	49.44	49.27	49.28	48.84
	Max WTM (%)	55.56	53.18	53.17	52.14
[32]	FC (%)	78.24	82.28	81.23	73.49
	Mean WTM (%)	26.75	25.99	25.92	26.18
	Max WTM (%)	31.72	29.58	35.10	35.46
[37]	FC (%)	68.14	77.29	77.12	80.48
	Mean WTM (%)	25.91	25.82	26.21	25.77
	Max WTM (%)	32.31	30.58	28.81	29.52
Prop.	FC (%)	94.42	92.44	91.85	91.63
	$\Delta$ FC (%)	<b>9.97</b>	<b>7.20</b>	<b>7.69</b>	<b>18.75</b>
	Mean WTM (%)	26.21	27.67	28.17	16.94
	$\Delta$ Mean WTM (%)	<b>-46.99</b>	<b>-43.84</b>	<b>-42.84</b>	<b>-65.32</b>
	Max WTM (%)	32.48	32.52	31.76	26.40
	$\Delta$ Max WTM (%)	<b>-41.54</b>	<b>-38.85</b>	<b>-40.27</b>	<b>-49.37</b>

coverage improvement rates of 15% and 18% are achieved compared with the methods from [32] and [37], respectively. This suggests the possibility of consuming relatively low hardware resources such as test points and memory elements for detecting additional RPR faults. Additionally, compared with conventional LBIST, fault coverage and switching activity are significantly improved within a limited test application time by up to 11% and 50%, respectively, on average.

As shown in Fig. 23, the fault coverage gradient was evaluated by considering the aforementioned circuits containing



**FIGURE 23.** Fault coverage for conventional LBIST, [32], [37], and the proposed method. (a) b17 (b) b18 (c) b19 (d) or1200.

large numbers of undetected faults compared with the ATPG process. In these graphs, the proposed method (indicated by the blue curve) exhibits the best performance in terms of fault coverage with an improved WTM. Furthermore, compared to the base point at which fault coverage is saturated in the conventional LBIST (indicated by the black vertical dashed line), the average test application time of the proposed method is improved by as much as 2.9 times on average (indicated by the blue vertical dashed line). This suggests that depending on how the targeted fault coverage is determined, the test application time can be further reduced while maintaining the enhanced mean and maximum toggling level.

## VI. CONCLUSION

In this paper, we proposed a new method for significantly improving both fault coverage and switching activity with limited test application time and few hardware resources. Experimental results obtained for the ITC'99 and OpenCores benchmark circuits with high RPR fault rates demonstrated that our method reduces both target factors compared with the classical method and the recently proposed solutions. Furthermore, our simple architecture could be implemented with an acceptable amount of area and low routing overhead.

It is noteworthy that these key factors can be improved concurrently. Our approach suggests the following. First, there is a possibility for reducing the test time further while maintaining low switching activity, depending on the outcomes of our approach. For instance, if the fault coverage is improved over a particular ASIL target requirement within

the targeted test time, then the test time can be further reduced as long as the required fault coverage is met. Second, additional hardware resources can be reduced. The fact that fault coverage has already been improved compared to the conventional works implies that hardware costs associated with test points or memory elements can be saved. In other words, as the number of undetected faults from random patterns relatively decreases through WGS, the additional hardware overheads to detect remaining faults can be minimized even if the desired fault coverage is not achieved. In future research, a more advanced method based on the proposed scheme is expected to achieve high performance and low-test power using a small number of test points and deterministic test data.

## REFERENCES

- [1] C. Mathas, "The price tag of automotive electronics: What's really at play?" EDN, Tech. Rep., Sep. 2017. [Online]. Available: <https://www.edn.com/the-price-tag-of-automotive-electronics-whats-really-at-play/>
- [2] A. Ahmad, "Automotive semiconductor industry—Trends, safety and security challenges," in *Proc. 8th Int. Conf. Rel., Infocom Technol. Optim. (Trends Future Directions) (ICRITO)*, Jun. 2020, pp. 1373–1377.
- [3] N. Mukherjee, D. Tille, M. Sapati, Y. Liu, J. Mayer, S. Milewski, E. Moghaddam, J. Rajsiki, J. Solecki, and J. Tyszer, "Test time and area optimized BrST scheme for automotive ICs," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2019, pp. 1–10, Paper 10.1.
- [4] *Road Vehicles-Functional Safety*, Standard ISO26262, International Organization for Standardization, Jan. 2011.
- [5] P. D. Hortensius, R. D. Mcleod, W. Pries, D. M. Miller, and H. C. Card, "Cellular automata-based pseudorandom number generators for built-in self-test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 8, pp. 842–859, Aug. 1989.
- [6] Y. Sun, S. K. Millican, and V. D. Agrawal, "Special session: Survey of test point insertion for logic built-in self-test," in *Proc. IEEE 38th VLSI Test Symp. (VTS)*, Apr. 2020, pp. 1–6.



- [7] L. Chen and S. Dey, "Software-based self-testing methodology for processor cores," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 3, pp. 359–380, Mar. 2001.
- [8] P. Georgiou, X. Kavousianos, R. Cantoro, and M. S. Reorda, "Fault-independent test-generation for software-based self-testing," *IEEE Trans. Device Mater. Rel.*, vol. 19, no. 2, pp. 341–349, Jun. 2019.
- [9] M. Psarakis, D. Gizopoulos, E. Sanchez, and M. S. Reorda, "Microprocessor software-based self-testing," *IEEE Des. Test. Comput.*, vol. 27, no. 3, pp. 4–19, May 2010.
- [10] F. Reimann, M. Glaß, J. Teich, A. Cook, L. R. Gómez, D. Ull, H.-J. Wunderlich, P. Engelke, and U. Abelein, "Advanced diagnosis: SBST and BIST integration in automotive E/E architectures," in *Proc. Design Automat. Conf. (DAC)*, Jun. 2014, pp. 1–9.
- [11] S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern generation for a deterministic BIST scheme," in *Proc. IEEE Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 1995, pp. 88–94.
- [12] R. Dorsch and H.-J. Wunderlich, "Accumulator based deterministic BIST," in *Proc. Int. Test Conf.*, Oct. 1998, pp. 412–421.
- [13] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 776–792, Jun. 2004.
- [14] E. J. McCluskey, "Built-in self-test techniques," *IEEE Des. Test Comput.*, vol. 2, no. 2, pp. 21–28, Apr. 1985.
- [15] P. Girard, L. Guillier, C. Landrault, S. Pravossoudovitch, and H. J. Wunderlich, "A modified clock scheme for a low power BIST test pattern generator," in *Proc. 19th IEEE VLSI Test Symp. (VTS)*, Apr. 2001, pp. 306–311.
- [16] S. Gerstendörfer and H. J. Wunderlich, "Minimized power consumption for scan-based BIST," in *Proc. Int. Test Conf. (ITC)*, Sep. 1999, pp. 77–84, Paper 4.1.
- [17] I. Pomeranz and S. M. Reddy, "3-weight pseudo-random test generation based on a deterministic test set for combinational and sequential circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 7, pp. 1050–1058, Apr. 1993.
- [18] F. Muradali, V. K. Agarwal, and B. Nadeau-Dostie, "A new procedure for weighted random built-in self-test," in *Proc. Int. Test Conf. (ITC)*, Sep. 1990, pp. 660–669.
- [19] A. Jas, C. V. Krishna, and N. A. Touba, "Hybrid BIST based on weighted pseudo-random testing: A new test resource partitioning scheme," in *Proc. 19th IEEE VLSI Test Symp. (VTS)*, Apr. 2001, pp. 2–8.
- [20] D. Xiang, X. Wen, and L.-T. Wang, "Low-power scan-based built-in self-test based on weighted pseudorandom test pattern generation and reseeding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 942–953, Mar. 2017.
- [21] A. S. Abu-Issa, "Energy-efficient scheme for multiple scan-chains BIST using weight-based segmentation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 3, pp. 361–365, Mar. 2018.
- [22] L. Li and K. Chakrabarty, "Test set embedding for deterministic BIST using a reconfigurable interconnection network," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 9, pp. 1289–1305, Sep. 2004.
- [23] H.-G. Liang, S. Hellebrand, and H.-J. Wunderlich, "Two-dimensional test data compression for scan-based deterministic BIST," in *Proc. Int. Test Conf.*, Nov. 2001, pp. 894–902.
- [24] B. Kaczmarek, G. Mrugalski, N. Mukherjee, J. Rajski, Ł. Rybak, and J. Tyszer, "Test sequence-optimized BIST for automotive applications," in *Proc. IEEE Eur. Test Symp. (ETS)*, May 2020, pp. 1–6.
- [25] N. A. Touba and E. J. McCluskey, "Bit-fixing in pseudorandom sequences for scan BIST," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 4, pp. 545–555, Apr. 2001.
- [26] K.-H. Tsai, J. Rajski, and M. Marek-Sadowska, "Star test: The theory and its applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 9, pp. 1052–1064, Sep. 2000.
- [27] W. Kang, H. Lim, and S. Kang, "Scan cell reordering algorithm for low power consumption during scan-based testing," in *Proc. Int. SoC Design Conf. (ISOCC)*, Nov. 2014, pp. 300–301.
- [28] S. Pathak, A. Grover, M. Pohit, and N. Bansal, "LoCCo-based scan chain stitching for low-power DFT," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 11, pp. 3227–3236, Nov. 2017.
- [29] S. Seo, Y. Lee, H. Lim, J. Lee, H. Yoo, Y. Kim, and S. Kang, "Scan chain reordering-aware X-filling and stitching for scan shift power reduction," in *Proc. IEEE 24th Asian Test Symp. (ATS)*, Nov. 2015, pp. 1–6.
- [30] Y. Z. Wu and M. C. T. Chao, "Scan-cell reordering for minimizing scan-shift power based on nonspecified test cubes," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 16, no. 1, pp. 10:1–10:29, Nov. 2010.
- [31] C. Zoellin, H. J. Wunderlich, N. Maeding, and J. Leenstra, "BIST power reduction using scan-chain disable in the cell processor," in *Proc. Int. Test Conf. (ITC)*, Oct. 2006, pp. 1–8, Paper 32.3.
- [32] M. Filipek, Y. Fukui, H. Iwata, G. Mrugalski, J. Rajski, M. Takakura, and J. Tyszer, "Low power decompressor and PRPG with constant value broadcast," in *Proc. Asian Test Symp. (ATS)*, Nov. 2011, pp. 84–89.
- [33] M. Filipek, G. Mrugalski, N. Mukherjee, B. Nadeau-Dostie, J. Rajski, J. Solecki, and J. Tyszer, "Low-power programmable PRPG with test compression capabilities," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 6, pp. 1063–1076, Jun. 2015.
- [34] M. Tehranipoor, M. Nourani, and N. Ahmed, "Low transition LFSR for BIST-based applications," in *Proc. 14th Asian Test Symp. (ATS)*, 2005, pp. 138–143.
- [35] M. Nourani, M. Tehranipoor, and N. Ahmed, "Low-transition test pattern generation for BIST-based applications," *IEEE Trans. Comput.*, vol. 57, no. 3, pp. 303–315, Apr. 2008.
- [36] F. Liang, L. Zhang, S. Lei, G. Zhang, K. Gao, and B. Liang, "Test patterns of multiple SIC vectors: Theory and application in BIST schemes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 614–623, Apr. 2013.
- [37] M. Omaña, D. Rossi, E. Beniamino, C. Metra, C. Tirumurti, and R. Galivanche, "Low-cost and high-reduction approaches for power droop during launch-on-shift scan-based logic BIST," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2484–2494, Aug. 2016.
- [38] G. Mrugalski, J. Rajski, Ł. Rybak, J. Solecki, and J. Tyszer, "Star-EDT: Deterministic on-chip scheme using compressed test patterns," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 4, pp. 683–693, Apr. 2017.
- [39] Y. Liu, N. Mukherjee, J. Rajski, S. M. Reddy, and J. Tyszer, "Deterministic stellar BIST for automotive ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 8, pp. 1699–1710, Aug. 2020.
- [40] E. Moghaddam, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer, and J. Zawada, "Logic BIST with capture-per-clock hybrid test points," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 6, pp. 1028–1041, Jun. 2019.
- [41] S. Milewski, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer, and J. Zawada, "Full-scan LBIST with capture-per-cycle hybrid test points," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2017, pp. 1–9, Paper 10.3.
- [42] S. Wang, Y. Higami, H. Takahashi, H. Iwata, and J. Matsushima, "Automotive functional safety assurance by POST with sequential observation," *IEEE Des. Test. Comput.*, vol. 35, no. 3, pp. 39–45, Jun. 2018.
- [43] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," in *Proc. 18th IEEE VLSI Test Symp.*, Apr. 2000, pp. 35–40.



**KWONHYOUNG LEE** received the B.S. degree in electrical engineering from Kwangwoon University, Seoul, South Korea, in 2012. He is currently pursuing the M.S. degree with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul.

Since 2012, he has been a DFT Engineer with Samsung Electronics, Hwaseong, South Korea. His current research interests include design for testability, logic BIST, and low power testing.



**SANGJUN LEE** received the B.S. degree in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2018, where he is currently pursuing the combined Ph.D. degree with the Department of Electrical and Electronic Engineering.

His current research interests include design for testability, scan-based testing, and low power testing.



**JONGHO PARK** received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2020, where he is currently pursuing the combined M.S. degree with the Department of Electrical and Electronic Engineering.

His current research interests include design for testability, low power testing, and logic BIST.



**INHWAN LEE** received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2020, where he is currently pursuing the M.S. degree with the Department of Electrical and Electronic Engineering.

His current research interests include design for testability, low power testing, and test scheduling.



**SUNGHO KANG** (Senior Member, IEEE) received the B.S. degree in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1986, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 1988 and 1992, respectively.

He was a Research Scientist with the Schlumberger Laboratory for Computer Science, Schlumberger Inc., Austin, and a Senior Staff Engineer in semiconductor systems design technology at Motorola Inc., Austin. Since 1994, he has been a Professor with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul. His current research interests include VLSI/SOC design and testing, design for testability, design for manufacturability, and fault-tolerant computing.

• • •