A New Procedure for Weighted Random Built–In Self–Test

Fidel Muradali, Vinod K. Agarwal and Benoît Nadeau-Dostie*

VLSI Design Laboratory, McGill University, Montréal, Québec, Canada
* Bell–Northern Reserach (Corkstown), Nepean, Ontario, C anada

Phone: (514) 398-7136
Fax: (514) 398-4470
e-mail: agarwal@pike.ee.mcgill.ca

Abstract

A procedure for determining weight distributions and implementing weighted random pattern generation in built–in self–test circuits is examined. Test lengths orders of magnitude shorter than those encountered in conventional pseudorandom testing are recorded.

1. Introduction

In an attempt to address the unreasonable test times encountered using conventional pseudorandom testing, this paper describes a procedure for incorporating weighted random pattern generation into self-testing circuits.

At the root of most common test pattern application mechanisms used today is stored pattern testing, pseudorandom pattern generation and the inevitable hybrid of both. Stored pattern testing involves the application of specific test vectors, each of which provides an incremental level of coverage. Externally applied, this approach may require large storage capability and expensive test units. The technique is straightforward and suitable for many *present* testing needs. However, considering that the specifications of a test unit are static, upper limits are physically imposed on such variables as test frequency, the number of I/O channels and the size of input (pin) buffers. On the other hand, circuit attributes are dynamic, that is various technology advancements can result in higher operating speeds, higher pin counts, and progressively larger test sets. This conflict ultimately implies costly test equipment upgrades or replacement. Therefore, it is questionable if long term economical testing is feasible with a standard stored pattern approach [Bas89].

Uniform random pattern or pseudorandom testing relaxes the functional and memory requirements of the external tester used. Often, relatively simple sequential circuits, such as linear feedback shift registers (LFSRs) [Gol67][Bar87] or cellular automata (CA) [Hor89], can be used to pseudorandomly generate test inputs such that there is an equal probability of assigning a 1 or 0 value to an input. The intention is that after a large number of these "pseudorandom"[1] vectors, a reliable level of fault coverage will be achieved. Unfortunately, experience has shown that with current circuit densities, an excessive number of test patterns may be needed to attain this goal. Moreover, only a relatively small number of the initial vectors cover a large portion of the detectable single stuck-at faults, leaving the majority of the sequence to be wasted in an attempt to detect random pattern resistant faults. In fact, most of the time penalty associated with random pattern testing is due to the application of these unnecessary patterns.

An obvious alternative is to use a joint test strategy – apply a reasonable number of pseudorandom test patterns and supplement this with stored pattern testing. However, it has been found that in many cases, the size of the stored test set needed nears 70% of the full deterministic set [Bas89]. Thus, this approach does not quite address the problem of limited storage in external testers.

The idea of incorporating test circuits on-chip, i.e. that of built–in self–test (BIST), offers numerous advantages, such as increased test portability, more efficient low level (probe) tests and easier diagnosis of failed chips at the board level. Of course, there are shortcomings which must be considered, for example an increase in area, and potential speed degradation if test circuitry is inserted in a critical path.

A refinement of conventional pseudorandom testing is the application of a non-uniform distribution of 1's to 0's as test inputs. This idea of *biased* or *weighted* random pattern (WRP) generation can result in a much higher rate of coverage than that of uniform random pattern testing alone. One drawback though is that proposed schemes to determine weights are quite complicated and the associated on-chip weighted pattern generators have, so far, been much more area "hungry" than their uniform random counterparts [Wun87][Sch75][Brg89].

This paper examines a built–in self–test (BIST) imple-

[1] In the course of this text, "pseudorandom" or "random" patterns refers to those which are pseudorandomly generated with a device such as an LFSR [Bar87].

1990 International Test Conference CH2910-6/0000/0660$01.00 © 1990 IEEE

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

mentation of a WRP generator for full serial scan-path circuits. Although multiple weight sets can be used to minimize the overall test length [Wun88][Wai88][Lis87], in a BIST environment, the high cost of implementing different weight distributions must be balanced with a generated test length which is deemed to be acceptable[2]. For this reason complete test pattern generation is performed using one uniform weight and one non-uniform weight distribution of fairly coarse resolution. Experimental results demonstrate that the performance of this generation scheme is orders of magnitude better than generic random testing alone.

The test methodology involves serially scanning in a test pattern and scanning out the corresponding test response to a compaction unit. Using such an approach makes the suggested BIST solution extendible to external testing. The single stuck-at fault assumption is used, however the analysis can be extended to any model which does not require a specific ordering of input test patterns (e.g. bridging, multiple stuck-at, etc.).

The text is divided into two sections: first, an ad-hoc simulation-based method of determining the characteristic weight sets is outlined; next, a distributed WRP generator is proposed for local generation of the dually weighted sequence. This generator design is intended for BIST and is suitable for automated design. At various stages, performance results are given in order to help evaluate the procedure.

2. Weight Set Estimation

This section discusses a new method of designing a weight set. A flowchart of the entire procedure is given in figure 1.

2.1 Previous Work

The idea of using weighted generation of test patterns is not particularly new. In some earlier work ([Sch75]) intended for large-scale integration (LSI), the relative weight of an input node is adaptively assigned based on the amount of internal switching activity caused by exercising that input.

With respect to VLSI circuits, [Sia88] proposes an alternate adaptive algorithm which reduces the prohibitive amount of simulation needed in [Sch75] to evaluate nodal activity. Instead of randomly toggling single input pins, most of the initial information is gathered using a functional test set which already exists for design verification purposes. By simulating each pattern of this set, the amount of activity per vector is assessed from observing the circuit inputs in conjunction with the corresponding output states. This information is used to develop an initial weight set from which weighted test patterns are generated and simulated. If any new vectors are found, the estimated weight set is iteratively refined using a similar process. A functional test set may not always be a proper choice for the prime source of information, especially if it provides poor fault coverage. In such cases, the initial *information* may be biased resulting in the final test sequence being geared to test only a portion of the circuit.

In an algorithm described in [Eic87], all possible paths from an output terminal to an accessible input are traced and an initial weight is calculated as a function of the number of circuit inputs feeding the blocks along the trace-back paths. Weighted simulation is then performed and deterministic test pattern generation (DTPG) is used to find test patterns for any faults left undetected. Additional weight sets are derived from the DTPG results. Because of differences in input circuitry to blocks along the trace-back paths, the estimation of the initial weight may be less than optimal in circuits with fanout.

In [Lis87], a test generation system is designed which uses testability measures ([Brg84]) to calculate an input weight set by minimising a cost function which balances CPU time and simulated test length. Patterns are generated according to the assigned weight set and fault simulation is done until a threshold rate of fault coverage is reached. Using the resulting reduced undetected fault list (thus, for simulation purposes, a reduced representation of the circuit), the process repeats to find supplemental weight distributions.

Recently, [Wun88] developed a detection probability-based tool to determine a user-specified number, n, of weight sets. Using detection probability estimates, a fault set is partitioned into n subsets such that the overall test length is minimized. A weight set is estimated for each fault grouping. The method is somewhat sensitive to detection probability accuracy, and the size and contents of the initial fault list. For example, if the contribution of redundant faults is not neglected, the calculated detection probabilities of this class of fault would corrupt intermediate test length estimates and could result in inaccurate weight sets and a pessimistic overall test length projection [Mur90].

Regardless of the inherent inaccuracies of the respective estimation processes, acceptable performance of weighted generation in each case suggests that the weight sets do not have to be very accurate. In a practical sense, since thousands of test patterns are generated, the weight must simply represent enough information so that its performance is superior to that of uniform random pattern testing.

2.2 Proposed Ad Hoc Method

As a departure from much of the previous work, the weight set estimation method discussed in the rest of this section relies neither on signal probability calculations nor on explicit analysis of the circuit structure. The conjecture is that sufficient circuit information for this purpose is already contained in a pre-determined test set, and can be extracted

[2] Production decisions may be based on a tradeoff of various factors e.g. test time, fault coverage, production quota, and development time, to name a few.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

using simulation techniques. In fact since the weight estimation method is based on the circuit's simulated behaviour during test mode, the circuit can be considered a functional "black box" during the method. Because fanout and redundant faults are not explicitly considered in a functional circuit representation, they should not directly contribute to the accuracy of the final weight set. Nonetheless, the existence of fanout and redundancies do, as usual, impact the computational effort of simulation tasks performed.

As an aid to the ensuing discussion, a flowchart of the entire scheme is provided in figure 1. The alphabetic tags associated with the various steps are referred to in the subsection(s) dealing with that stage of the process.

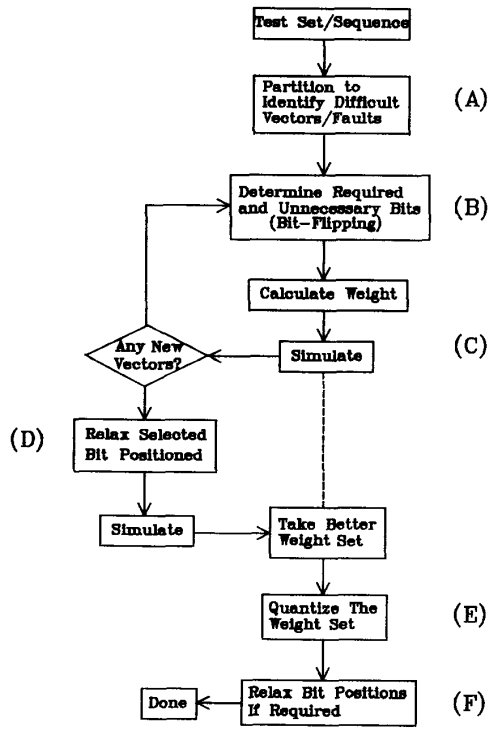

Figure 1 Flowchart of the Weight Estimation Method

The first step (A) is to identify the required test vectors which complement the conventional random test patterns for archieving a very high fault coverage. This can be done by tracking an associated random pattern detection profile (see figure 2 solid curve) and partitioning the curve at the point at which the rate of coverage begins to decrease by a user-defined threshold amount. Typically this might occur in the "knee" region of the curve. The groups of vectors found on either side of the partition point form two subsets, each of which defines a different weight.

In the steeper sloped region to the left of the partition, faults classified as random **easy** are quickly detected in a relatively short test length. Since the performance of random patterns is already acceptable in this region, a uniform weight of 0.5 (random) is used to cover these faults. As the test length approaches infinity, random **difficult**[3] faults are found at a comparatively slower rate. This is characterised by the long flattened tail region to the right of the partition. Because the vectors occurring within this tail segment are more resistant to random pattern generation, a non-uniform weight set is determined to cover them (and thus their associated faults) more rapidly. The expected effect of using such a pair of weights is shown in figure 2.

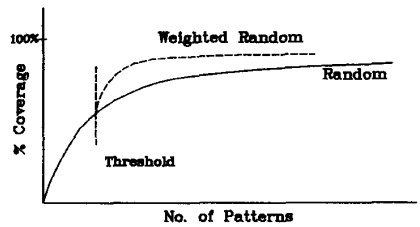

Figure 2 Expected effect of 2 weight sets

2.2.1 Processing Tail Vectors (B)

In many cases it may be unrealistic to perform fault simulation until all difficult faults are detected. Instead, a cut-off limit may be imposed on the rate of coverage, after which simulation is terminated. In such an instance, although it is possible that not all difficult faults are covered by the test sequence, if the rate of coverage at the cut-off is chosen to be sufficiently small so that most of the difficult faults are found, the tail vectors can still be used to determine an initial non-uniform weight set. Since, aside from the circuit itself, the vectors for the detected difficult faults represent the only available information for determining the weight set, the estimated weight set will be geared to cover the detected difficult faults (called *target* faults). ATPG tools or a method such as that described later in this section can be used to determine supplemental tail vectors for missed faults.

The pool of vectors which detect target faults is averaged per bit to provide the necessary information needed to calculate the non-uniform weight. Each test vector contains information concerning the circuit faults which it detects. However, some of this information may be redundant since some faults are repeatedly detected by subsequent tail vectors.

In order to formulate a weight set characteristic of the target faults, an attempt is made to reduce, if not remove, any redundant information contained within each tail-end vector and isolate the bit assignments useful in detecting

[3] Note that the qualitative classification of a fault as "difficult" or "easy" is always with respect to a specific type of generator, be it random or biased according to a specific weight set; unless specified as otherwise, "difficult" will refer to uniform random generation

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

faults. The first task is accomplished by using a fault dropping scheme, demonstrated in figure 3, to assign to each tail vector (V) a list of unique target faults (t) which excludes those faults detected by any previous vector.

Vector	Original Fault List	Assigned Fault List
V_1	$t_1\ t_2\ t_3\ t_6$	$t_1\ t_2\ t_3\ t_6$
V_2	$t_1\ t_3\ t_4$	t_4
V_3	$t_1\ t_2\ t_3\ t_5$	t_5

Figure 3 Reducing the contribution of multiple detection of faults

Next, we notice that for each tail vector, V_i, not all bit assignments are instrumental in detecting its associated faults. Any simulation tools such as Tulip [Maa90] can be used to provide a partial estimate of the necessary bits per vector. ATPG tools can also be configured to provide some of this information.

In the experiments performed a *bit flipping* algorithm was used to determine the needed bits per vector. This is introduced here :

- Given an n bit tail vector V_j, create n new unique vectors $(v_1, ..., v_i, ..., v_n)$ each of whose Hamming distance from V_j is 1. This is done by inverting the ith bit of V_j to create v_i.
- Simulate each vector v_i with respect to the target faults associated with V_j.
  - if the coverage decreases then bit i is needed

    else bit i is marked "don't care" (unnecessary)

The "filtering" process described above results in a sparse test set in which only the needed bit assignments per tail vector are retained. Using this test set representation, a weight set which can cover the set of target faults can be estimated by calculating the ratio of the number of 1s to 0s within each bit position (input position). A generator defined as such, attempts to produce a test sequence containing vectors similar to (ideally identical to) that of the filtered test set.

Vector	Before Bit Flipping	After Bit Flipping
V_1	1 1 1 0 0 1	x 1 1 x x x
V_2	0 1 1 1 0 1	0 1 x 1 x x
V_3	1 1 0 0 0 0	1 1 x x x 0
V_4	0 1 0 0 0 1	0 1 0 x x x
weight	.5 1 .5 .25 0 .75	.33 1 .5 1 .5 0

Figure 4 Weight Set Estimation

In figure 4, a weight set is calculated for a sample set of vectors before and after extraction of the needed bits. Notice that that the removal of the unnecessary bits (shown as 'x' after bit flipping) results in a much different weight for the first, fourth, fifth and sixth bit positions. Before bit flipping is performed, the definite assignments to the unnecessary bit positions act as "noise" which corrupt the weight estimate. This is apparent, for example, in the sixth bit position – before bit flipping the weight is 0.75 but it is found that a 0 bias is more appropriate. Filtering off these assignments should result in a more accurate weight set.

2.2.2 Experimental Results

Before proceeding further, we describe some results on weight estimation and its performance. The test generation scheme is evaluated using five scannable circuits of varying size[4]. The complexity of their topology is indicated in table 1.

NETLIST	#Faults	# Inputs	# Outputs	# Gates	# Levels
C6941	5673	247	250	3780	55
C9389	8075	725	786	5044	53
C11657	9836	615	687	6541	72
C30989	28572	1668	1668	16299	44
C35002	32985	1464	1578	16820	45

Table 1 General Circuit Information

The first stage of the process is to determine an initial test set and isolate the difficult faults. This is done by simulating[5], with a collapsed fault set, the order of 1 million random patterns and partitioning the sequence at the point where less than roughly 30 faults were detected in a window of 3000 consecutive random patterns[6]. As mentioned before, the partition threshold is flexible. The results are shown in table 2. For each circuit, two test lengths and their respective fault coverages, in terms of the number of irredundant faults left undetected, are given. First, in column II, the test length required to reach the partition point (i.e. those which cover the easy faults) is shown. Column III lists the fault coverage of this short sequence. The maximum number of random patterns simulated (L_o) and its associated coverage are given in columns IV and V respectively. Finally, the number of **target difficult** faults identified in the test sequence after the partition point (i.e. after the test length of column II) is shown in column VI. This fault pool provides information used in the bit–flipping process.

[4] These circuits are precursors to the ISCAS-89 Benchmark circuits.

[5] The fast fault simulator used is Tulip [Maa90] and redundancies were later found using [Raj90].

[6] Although the initial test set was determined using simulation, it is stressed that it is not important how the test set was developed.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

I NETLIST	II R.P. Test Length	III # Undet Faults	IV Max. R.P. Test Length (L_o)	V # Undet Faults	VI # Target Faults (V - III)
C6941	5.5K	751	3M	7	744
C9389	12K	508	0.2M	0	508
C11657	5.3K	517	1M	94	423
C30989	13K	1954	2M	41	1913
C35002	15K	370	1M	33	337

Table 2 Initial (random pattern) Circuit Data

The next step is to isolate the difficult vectors and use bit–flipping to determine which bits are needed. Using this information, a weight set is then calculated and a set of 40K WRPs is simulated (table 3). This limit on the WRP test length is chosen to satisfy approximate test time restrictions[7] but in practice, of course, the upper bound can vary. Column III (L_{wrp}) of table 3 is the point in this WRP test sequence after which coverage of the difficult faults does not increase significantly. L_{rp} (col. II) is the number of extra uniform random patterns required to cover the easy faults not yet detected. The coverage of the pair of weight sets is indicated, in column IV, by the number of faults left undetected after applying $L_{wrp} + L_{rp}$.

I NETLIST	II L_{rp}	III L_{wrp}	IV # Undet. ($L_{wrp} + L_{rp}$)	V # New Found	VI # Targets Missed	VII # Undet. ($L_{wrp} + L_{rp} + L_o$)
C6941	5.5K	38K	8	7	8	0
C9389	12K	34K	0	0	0	0
C11657	5.3K	34K	24	77	7	17
C30989	13K	40K	31	33	23	8
C35002	15K	37K	21	22	10	12

Table 3 Initial WRP data

Test length improvements ranging from 1 to 3 orders of magnitude are recorded. For example, more than 2 million pseudorandom patterns were needed to test C30989 with a coverage of 41 undetected faults, while 31 undetected faults remain after a joint sequence of 53K patterns. Also, although a few target faults are missed, a number of previously undetected ones are found (col. V). Recall that these newly detected faults could have required a number of random patterns far in excess of the maximum amount originally simulated (table 2). In fact, if weighted random patterns are applied after the maximum radom pattern test set, the coverage is quite high as shown in column VII – $(L_{wrp} + Lrp + L_o)$.

[7] Another measure of this would be to more specifically consider the scan chain length and limit the number of bits in the test sequence.

2.2.3 Improving the Estimated Weight Set (C and D).

Referring to figure 2, as expected, within the WRP region there is again an extended tail segment where WRP difficult faults[8] are detected. This region may be excessively long for one or more of the following interrelated reasons:

- test vectors for new faults are slowly being found
- characteristics of all faults are not sufficiently represented by one non–uniform weight set
- processed vectors with many retained bits dominate the average
- many processed vectors of similar form dominate the average

One could potentially use steps A and B in an iterative manner to estimate a new weight set from the WRP difficult faults of the first weight set. However, since we would like to limit the number of non-uniform weight sets to just 1, we propose two modifications to alter the original weight set to obtain a better fault coverage.

- Update the initial pool of target faults.
- Relax selected input biases.

These modifications are ad hoc and iterative in nature.

Phase 1 - Update Target Pool

The effectiveness of the method depends on how well the pool of target faults, defined in section 2.2.1, represents all the existing difficult faults. It is possible that the initial biased generator will detect some extra faults which are not detectable using realistic random test lengths. In the modified method these new test vectors are included into the target pool and the weight set is re-estimated. This process of iterative WRP generation and re-estimation of the weight set could be repeated until no new faults are detected.

Phase 2 – Relax Pin Biases

In an attempt to further reduce the weighted test length, the biases of selected pins are relaxed. After a suitable weight set is found in Phase 1, the circuit is re-simulated using the weighted generator and only the target pool of faults is considered. Again, by noting the progression of coverage, there will be a small group of vectors which are found after a long WRP sequence. The assumption is that these few WRP difficult vectors strongly differ from the generator bias at several positions. A partial remedy is to force some of these conflicting pin biases to 50%. This compromise should not adversely affect the test length if the number of positions modified is small.

The method of selecting bit positions to be relaxed is rather straightforward. According to a user-undefined thresh-

[8] Recall again that the qualitative classification of a fault as "difficult" or "easy" is always with respect to a specific type of generator, be it random or biased according to a specific weight set.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

old criteria, vectors found in the end of the WRP tail region are stripped from the WRP test set. An example of such a threshold is to strip all vectors found after the point where no fault is detected by 1000 consecutive WRPs.

Using steps A and B, a new weight set is estimated from the stripped WRP vectors and is compared to the original weight set. Bit positions which differ by a user-defined amount (e.g. 0.7), are relaxed. For example :

Original Weights	1	0	.2	.7	.8	.2	.1
New Weights	1	0	.4	.8	.1	.9	.3
Modified Weights	1	0	.2	.7	**.5**	**.5**	.1

Note that this is a potentially volatile step because disturbing the generator bias (i.e. the original weights) may negatively affect the detection of WRP easy faults. As a result, it is recommended that this phase be performed only once and the number of positions relaxed be small.

2.2.4 Further Experimental Results

I NETLIST	II L_{rp}	III L_{wrp}	IV # Undet. ($L_{rp} + L_{wrp}$)	V # New Found	VI # Target Missed	VII # Bits Relaxed
C6941	5.5K	33K	**4**	0	4	5
C9389	12K	4K	**0**	0	0	6
C11657	5.3K	25K	**0**	17	0	10
C30989	13K	40K	**30**	0	22	16
C35002	15K	29K	**12**	4	5	6

Table 4 Refined WRP Data

Continuing with the circuits and test sets described in section 2.2.2, Table 4 is obtained after refining the weight set in the manner described above. Here, the number of *new faults* detected does not include those found in the initial WRP run. As expected, there is a general improvement in the weighted test length (col. III); in each case accompanied by an increase in coverage. Notice that for C30989, the WRP test length is the 40K limit. If ten thousand more weighted patterns are simulated, the coverage is 22 undetected faults, thus illustrating the possible tradeoff between test time and coverage. A summary of the performance of the complete generation scheme in reducing the random pattern test length otherwise needed to achieve equivalent coverage, is shown in table 5.

Relaxing an input's bias to 0.5 also reduces the implementation overhead since a modified scan cell is not needed at that position. In all the cases, test length did not suffer, thus it could be beneficial to investigate the maximum number of inputs which could be relaxed without seriously increasing the WRP test length and coverage.

NETLIST	Refined WRP Data L_{rp}	L_{wrp}	# Undet Faults	Equiv. Rand. Pat. Test Len.	Test Length Reduction
C6941	5.5K	33K	**4**	4M	99%
C9389	12K	4K	**0**	0.5M	92%
C11657	5.3K	25K	**0**	>30M	>99%
C30989	13K	40K	**30**	4M	98%
C35002	15K	29K	**12**	>5M	>99%

Table 5 Weighted Testing v.s. Random Testing

3. Hardware Implementation

One of the major concerns when building an on-chip generator is to minimize the area overhead without severely compromising test quality and testability of the BIST circuitry. Two interdependent issues are involved in the proposed implementation strategy :

- The hardware used to generate patterns according to a pair of weight sets must be simple and transparent to the CUT. It must also be general enough to be used in any scan circuit testable with one non–uniform weight
- The weight set determined must be adapted to suit the generator.

The suggested approach is to redesign specific scan cells so that from the pattern generation point of view, the CUT appears testable with a uniformly distributed pseudorandom sequence.

3.1 Local Generation of Dually Weighted Test Patterns

A Distributed BIST Generator provides test vectors according to **both** the uniform and determined weight sets. It consists of a generating scan chain (GSCAN) and an on-chip autonomous random pattern generator such as a CA or LFSR. The global effect is that each generator output (CUT input) can be made to be dually weighted – generating patterns according to the bias of the source generator, or according to a hardwired weight (figure 5).

GSCAN is an altered scan chain which serves (apart from its other functions) as an interface between a random pattern generator and a WRP testable circuit. At a number of times proportional to the ratio of the WRP test length to the random test length, the random input at selected pin positions is locally converted to a weighted circuit inputs of specific biases. This is done by replacing standard scan elements by modified blocks which perform the conversion. These new cells are themselves generic and can be included in a cell library for automated design and layout.

Since the weight set is a mapping of bias values to input pins, the scan cells to be replaced are those which correspond to a bias other than 0.5.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

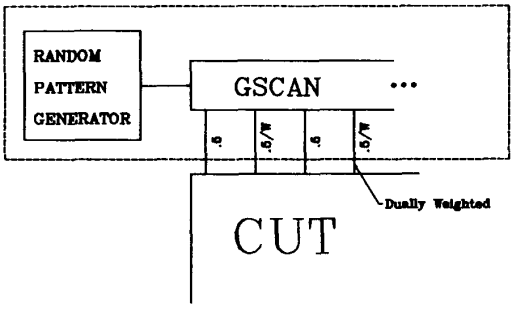

Figure 5 Distributed Generation of Patterns according to 2 weight sets

3.1.1 Modified Cells

The logical block diagram of the circuit for locally generating a biased input sequence, is shown in figure 7. It differs from a regular scan element (figure 6) in that there are :

- n-1 possible extra input lines from **previous** scan elements.
- Extra combinational logic (WLOGIC) to generate the biased stream.
- Output multiplexing to choose between biased streams.
- Independence between the scan output and the input to the CUT.
- A global control line (WT-SEL).

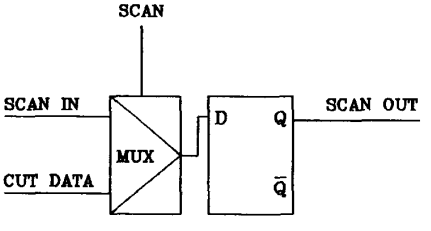

Figure 6 Generic Scan cell

The means of generating a weighted sequence exploits the output signal probability of logic gates. Extra combinational logic (shown as WLOGIC in figure 7) combines n random input streams, one from the resident flip flop and $n-1$ from the $n-1$ previous cells, to give a weighted output stream. For example if WLOGIC is an AND gate and n is two, the bias of the weighted sequence is around 0.25.

In general, the output multiplexer and WLOGIC can be represented by a small, functionally equivalent, combinational circuit. The GSCAN cell which conditionally generates a bit stream of bias 0.25 is shown in figure 8. It is a

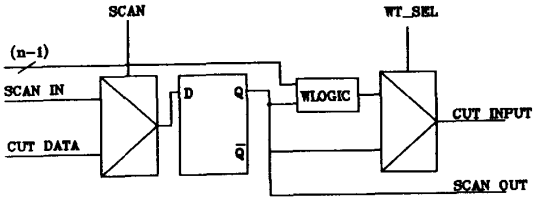

Figure 7 Scan cell modified for WRP generation

trivial task to design similar cells for biases of 1, 0 and 0.75. More finely tuned weights such as 0.125 and 0.875, etc., can be generated if values from greater than 1 neighbour are combined (i.e. n≥3). In such cases, however, the cost of routing might become prohibitive.

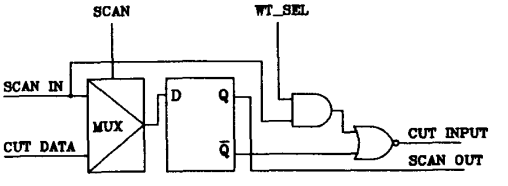

Figure 8 GSCAN cell for a bias of 0.25

3.2 Mixed Generation/Application of Uniform and Weighted Patterns

WT-SEL is a control line used to select between the weighted bit stream generated locally and the contents of a GSCAN cell's resident D-flip flop (D-FF). The signal can be applied externally or, in a true BIST sense, generated by logic connected to the RPG. Furthermore, the signal can be encoded such that weighted and non-weighted test pattern application is done in separate test sessions, or encoded as a *weighted control* so creating a *mixed-weighted* generation scheme in which weighted and non-weighted patterns are randomly applied in the same test interval. This is shown in figure 9.

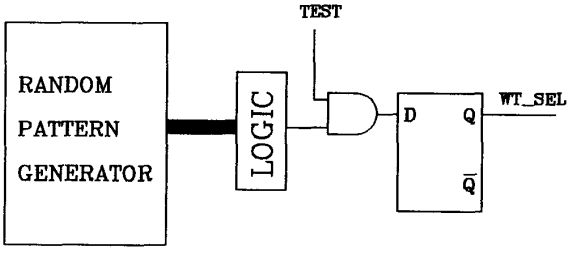

Figure 9 Block Diagram of WT-SEL Generation

Here, the signal assumes 2 'modes' of operation which are outlined in table 6 and described below.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

MODE	TEST	DESCRIPTION
W	1	WT-SEL = weighted sequence
0	0	WT-SEL = 0

Table 6 Modes of operation for WT-SEL

W - Weighted Mode

When TEST=1, WT-SEL is itself a weighted sequence whose weight reflects the proportional sizes of the WRP and random pattern test lengths. Whenever WT-SEL=1, the corresponding test vector applied to the CUT belongs to the biased random test set, and when WT-SEL=0, the corresponding vector belongs to the uniform random test set. In another words, WT-SEL is used to choose either the biased stream generated or the random sequence scanned in from the RPG. This technique is possible because the order in which the individual test vectors (weighted vs. random) are applied is not critical when testing combinational circuits for the single stuck-at faults.

0 - Mode

In this mode, WT-SEL is permanently 0 and all the vectors applied are of uniform random type only.

It is a simple task to integrate other modes of operation to satisfy other desired scan functions.

3.2.1 Area Penalty

Compared to a regular scan element, the area overhead of a redesigned scan cell in which the weighted stream is 0.25, 0.75 or 1.0 is :

- a pair of 2 input gates.
- the extra routing inside the cell.

In the case of a 0 bias, a simple NOR (4 transistors) is needed. It is of course clear that if pin biases are chosen such that inputs are needed from more than one neighbours, the overhead and layout complexity will increase due to the routing from progressively distant cells.

With reference to a conventional random pattern BIST implementation, the extra area penalty of this scheme is due to the contribution of each modified scan cell, the logic needed to generate WT-SEL and the additional interconnect required to globally distribute this control signal.

3.3 Quantization of Pin Biases

Thresholds are established to force the individual biases of the estimated weight set to implementable values. The criteria should ensure that:

- The overall features of the experimental bias, especially the extreme values, are retained.
- The levels are chosen to facilitate minimum area overhead.

The latter point implies a tradeoff between design complexity, fault coverage and test time. For example, the most attractive levels in terms of area and routing are 0, 0.25, 0.5, 0.75 and 1. However there is a potential for a loss of fault coverage when only these values are used because extreme pin biases, such as 0.05 and 0.95, are forced to 0.25 and 0.75 respectively. This will, in most cases, result in an increase in test length or a possible loss of coverage if the test size is limited. One solution, of course, is to use more quantization levels but this results in higher overhead.

In general, the thresholds and number of quantization levels used depend on what is deemed acceptable production-wise. It will be shown experimentally in the next section that biases restricted to 0, 0.25, 0.5 0.75 and 1 are indeed sufficient to attain acceptable fault coverage. The results of including levels 0.125 and 0.875 are also included for completeness.

3.4 Experimental Results

Table 7 shows the results of using the bias levels <0, 0.125, 0.25, 0.5, 0.75, 0.875, 1> and table 8 for the levels <0, 0.25, 0.5, 0.75, 1>.

NETLIST	L_{rp}	L_{wrp}	# Undet. Faults
C6941	0.5K	40K	7
C9389	11.5K	6K	0
C11657	5.3K	37.5K	4
C30989	13.5K	39K	54
C35002	15K	38.4K	17

Table 7 Quantization Runs for 7 bias levels

NETLIST	L_{rp}	L_{wrp}	# Undet. Faults
C6941	0.4K	39K	36
C9389	11.5K	6K	0
C11657	5.3K	35K	8
C30989	11.5K	39.5K	73
C35002	15K	30.8K	25

Table 8 Quantization Runs for 5 bias levels

The result of using a joint scheme with 7 bias levels, in terms of both coverage and test length, is much better than that of uniform random generation alone. In a more practical sense, the coarser weight set of 5 levels uses only the less complex GSCAN cells and also produces a significant improvement. The implementation overhead (beyond scan) for each circuit of this latter case is approximately 3%. This is calculated by using an industrial standard cell library to estimate the size of each circuit and to this amount, the extra area needed in order to modify the scan chain is added.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

3.5 LFSR–Based Scheme

In the previous experimental analysis, software random pattern generation (therefore almost no correlation between generated input bits) was used to evaluate the potential of a dually weighted generation scheme and examine the effect of using a coarse weight set. In order to evaluate the **mixed–weighted** implementation scheme, this section presents the results of simulating a model of the distributed test structure using a 32-bit LFSR (table 9) and 5 bias levels. The performance is compared to that of 'ideal' uniform random pattern testing implemented in software.

For each circuit a 70 thousand mixed-weighted pattern test sequence (Test Len.) is generated with a weighted contribution indicated by the bias of the WT-SEL signal. The number of undetected faults at the 50 thousand and 70 thousand mark is shown along with the random pattern (R.P.) test length needed for equivalent coverage.

NETLIST	WT_SEL_{bias}	Mixed-Weighted Test Len.	# Undet Faults	Equiv R.P. Test Len.
C6941	75%	50K	19	1.9M
		70K	15	1.9M
C9389	13%	50K	4	120K
		70K	1	125K
C11657	75%	50K	23	15M
		70K	13	>30M
C30989	63%	50K	95	1M
		70K	55	1.7M
		150K	23	5M
		200K	18	>5M
C35002	37%	50K	58	144K
		70K	28	2.5M

Table 9 Results for the modeled BIST Implementation with 32 bit LFSR

Higher fault coverage is expected if the entire analysis (partitioning fault sets etc.) were performed using an LFSR. It has also been found from experience that changing the LFSR seed during test mode positively affects the rate of coverage. More detailed experimental results on this aspect are contained in [Mur90].

According to the modified scan cell design, by permuting the bits at neighbouring input sites to form weighted stimuli, it is expected that some correlation between neighbouring input bits is created during weighted pattern application. In fact, in special scan chain orderings [Mur90], some input states are not generatable during weighted mode. However, the results of table 9 indicate that this correlation, as well as that inherent to the LFSR, does not have a serious negative impact on overall test performance. Furthermore, by observing the test vectors for the difficult faults missed, the sites for which the missing input state is critical can be identified. If the corresponding modified scan cells are replaced by generic ones (i.e. the bias is relaxed to 0.5) the problem is eliminated with little effect to the test length, provided that the number of positions relaxed is small. Further investigation into the impact of correlation introduced by modifying the scan chain and due to LFSR generation is required.

4. Conclusions

It is proposed that a pseudorandom sequence and a single weighted random sequence be used to efficiently implement built–in self–test in a large integrated scan–circuit which would otherwise need an excessive pseudorandom test length. Based on fast fault simulation tools, a method of determining the weight set and the approximate pseudorandom and weighted random test lengths is suggested.

By modifying specific scan cells, the BIST hardware conditionally generates the weighted stream *locally*, at specific input sites. A *weighted control* signal is used to regulate the proportion of weighted and pseudorandom inputs.

Apart from demonstrating that in the cases examined, one weight set was sufficient for a notable decrease in test time, it was also noticed that a very coarse weight set (i.e. restricting biases to 0, 0.25, 0.5, 0.75 and 1) provides acceptable results. Using finer resolution within the weight set usually results in a slightly higher coverage but at the expense of a much higher area overhead.

REFERENCES

[Bar87] P.H. Bardell, W.H. McAnney, J. Savir, *Built-In Self-Test for VLSI*, Wiley-Interscience, New York, 1987

[Bas89] R.W Bassett et al., "Low Cost Testing of High Density Logic Components", *Proc. ITC-89*, pp.550-557, Washington, DC, September 1989.

[Brg89] F. Brglez, C. Gloster, G. Kedem, "Hardware-Based Weighted Random Pattern Generation for Boundary Scan", *Proc. ITC-89*, Wahsington, DC, pp.264-273, 1989.

[Eic87] E.B Eichelberger et al., "Weighted Random Pattern Testing Apparatus and Method", *U.S. patent No. 4,745,355, Aug. 18 1987.*

[Fed86] X. Fedi and R. David, "Some Experimental Results From Random Testing of Microprocessors", *IEEE Trans. Inst. & Meas.*, Vol. IM-35, No. 1, pp.78-86 March 1986.

[Gol67] S.W. Golomb, *Shift Register Sqeuences*, Holden-Day Inc., San Francisco CA, 1967.

[Hor89] P.D. Hortensius, R.D. McLeod, W. Pries, D.M. Miller, and H.C. Card, "Cellular Automata-Based Pseudorandom Number Generators for

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.

Built-In Self-Test", *IEEE Trans. Comp. Aided Design*, Vol. 8, pp. 842-859, Aug. 1989.

[Lis87] R. Lisanke, F. Brglez, A. Degeus and D. Gregory, "Testability-Driven Random Test-Pattern Generation", *IEEE Trans. CAD*, Vol. CAD-6, No.6, Nov. 1987.

[Maa90] F. Maamari and J. Rajski, "A Method of Fault Simulation Based on Stem Regions", *IEEE Transactions on Comuter-Aided Design*, Vol. 9, No. 2, February 1990, pp. 212-220.

[Mur90] F. Muradali, "A New Procedure for Weighted Random Built-In Self-Test", *Master of Engineering Thesis*, Department of Electrical Engineering, McGill Universty, Montreal, March 1990.

[Raj90] J. Rajski and H. Cox, "A Method to Calculate Necessary Assignments in Algorithmic Test Pattern Generation", *to appear in Proc. ITC-90*, Washington, DC, September 1990.

[Sch75] H.D. Schnurmann, E. Lindbloom, R.G. Carpenter, "The Weighted Random Test Pattern Generator", *IEEE Trans. Comp.*, Vol.C-24, pp. 695-700, July 1975.

[Sia88] F. Siavoshi, "WTGPA: A Novel Weighted Test-Pattern Generation Approach for VLSI Built-In Self-Test", *Proc. ITC-88*, pp. 256-262, Washington, DC, September 1988.

[Wai88] J.A. Waicukauski, V.P. Gupta and S.T. Patel, "Fault detection Effectiveness of Weighted Random Patterns", *Proc. ITC-88*, pp. 245-255, Washington DC, Sept. 1988.

[Wun87] H.-J. Wunderlich, "Self-Test Using Unequiprobable Random Patterns", *Proc. FTCS-17*, Pittsburgh, 1987.

[Wun88] H.-J. Wunderlich, "Multiple Distributions for Biased Random Test Patterns", *Proc. ITC-88*, pp. 236-244, Washington DC, Sept. 1988.

Authorized licensed use limited to: Siemens AG GBS B&S SOL GMS. Downloaded on September 13,2024 at 14:03:20 UTC from IEEE Xplore. Restrictions apply.