AN EFFICIENT SEGMENTED RANDOM ACCESS SCANARCHITECTURE WITH TEST COMPRESSION

Maddumage Karunaratne[1] and Bejoy G. Oomman[2]

[1]Department of Electrical Engineering, University of Pittsburgh, Johnstown, PA, USA
[2]Genesys Testware/Broadcom Inc., San Jose, CA, USA

ABSTRACT

Integrated circuit (IC) chip designs relying on Random Access Scan (RAS) architecture for post-production structural tests typically provide lower test power dissipation, test data volume, and test application time compared to the classical serial scan-based Design for Test (DFT) methodology. However, previous RAS schemes incur high signal routing and test area overheads relative to the serial scan way. Unlike serial scan schemes, previous RAS schemes have not been effectively combined with test compression to further reduce test application time and test data volume. Authors have already formally documented a locally addressed (segmented) and compressed Segmented RAS (SRAS) architecture with low area overhead and test application time. This paper describes the SRAS architecture in more detail and provides comparative experimental results. Area overhead is reduced using test access hierarchy (segmented), while adding compression to RAS lowers the test application time.

Also presented is another enhancement to incorporate a scan channel multiplex block at hierarchy segments which helps drastically decrease the area and routing overhead of the original architecture to practically implementable levels on commercial circuits. The extra Segment Data Multiplexor (SDM) blocks reduce the area overhead of other components by the multiplexing factor, and the reduction in overall area is significant based on experimental data.

Test data compression and auto addressing of segments are achieved by transmitting a seed address to select segments with auto-increment or auto-decrement capability followed by either single cell selection or entire leaf cell segment selection. To further reduce the area overhead and test power, this architecture is enhanced to contain multiple channels at a cost of increased overall test application time with no increase in test data volume. Results of applying the enhancements to a large circuit with one level of intermediate segments with each of them having 256 leaf segments are presented in the paper with and without multi-channel multiplexing for comparison.

KEYWORDS

Design for Test, Random Access Scan, Data Compression, Signal Routing

1. INTRODUCTION

The main advantage of serial scan technique is the relatively small area occupied by the test circuitry added to facilitate testing IC chips after production. This test technique is ideal if all the flip-flops in scan chains are to be controlled (loaded) with stimulus data and their response data observed (unloaded) for comparison with expected values generated by automatic test pattern generation tools (ATPG) at design stage. However, for large circuits, serial scan method is not ideal since only a small fraction of the flip- flops needs to be loaded or unloaded in most test patterns. Since all the flip-flops in a scan chain are clocked in every cycle of a load or unload, such circuits dissipate much higher power during actual test application compared to functional operations. Consequently, test application time (TAT) cannot be minimized without over-

designing the power-delivery network. To mitigate such issues, various approaches have been proposed and practiced, such as loading constant values into serial scan cells that are not required in the current test pattern. However, all flops in the circuit under test (CUT) are clocked during shifting, and the clock network is a significant contributor to chip power. Therefore, tests can still dissipate high power even after mitigation.

RAS [1~9] techniques reduce test power by clocking only the specific scan cells required for the current test pattern. Ando [1] first introduced the RAS technique, which was later adopted by others [2~3]. As thename implies, RAS allows access to any randomly chosen flip-flop in CUT, and changes only that element.Therefore, the power dissipation is extremely low relative to serial scan schemes. In attempts to lower overall TAT and data volume, additional techniques within RAS, such as test vector ordering based on pattern value distances and test output response compression using multiple-input signature registers (MISR) have been employed [4]. However, unknown, or unpredictable (X) data values would make MISRresults invalid and test coverage low. Total test time will still be high.

Another team [5] describes a test vector compression combined with an iterative scan design technique while employing RAS row and column address decoding architecture. They employed two flavors of RAScells based on probability of having a 0 or 1 using ATPG vector data, along with compressing test data using coding schemes for adjacent 0's against 1's. However, their test flow may not be practical for largecommercial design teams that lack resources and time required to iterate design changes after ATPG has been performed before the design sign-off. Relying on a data driven DFT to achieve production level testcoverage may not be acceptable for such design teams. A cocktail approach on RAS to achieve low powerand overall high-test efficiency has been described [6] earlier. It incorporated the conventional RAS architecture and a cyclic random scan test scheme where output response of a scan cell group was taken as seed for the next random pattern. It is an overall test strategy with more efficient random test generation rather than improving RAS technique itself. Despite all the advantages, RAS schemes significantly increase IC chip area due to the additional logic circuitry and wiring to address each individual scan element.

To mitigate routing overhead, one team [7] employed a scheme similar to static random-access memory in their progressive RAS architecture reducing both area and routing overhead compared to their previous work [4]. They progressively activated each row in the RAS grid structure by shifting the decoded row selection vector, while the decoded column selection vector selected one column per write operation, but all columns for read operation with read data feeding a MISR. However, unknown, or unpredictable (X) values would make MISR results invalid and test coverage low. Another group [8] designed a toggle scanflip-flop (TRAS) eliminating two global signals (i.e. scan-in and scan-enable) required in typical RAS structures. Toggling the target flip-flop effectively loads the required value, while captured value is shifted out via a bus structure when subsequent flip-flops are accessed. Using a grid architecture, they have mitigated routing overhead of conventional RAS to some extent. However, with toggling being the only way to set a value, loading a test vector would require loading new row/column addresses or shifting those already loaded in both directions which drastically adds to test time.

Another paper [9] introduces a layout-based design approach named localized random-access scan (LRAS)to eliminate the global test enable signal while localizing row and column enable signals. RAS scan cells are clustered into blocks based on the estimated layout data gathered from initial design stages. Each block is organized into a RAS structure and selected by decoding a block access bus while smaller uneven row and column address bus lines are routed to all the clustered

blocks for local decoding. The captured scan test results still need to be propagated out as conventional RAS does from each block which still makes the overall test time high and routing less than optimal.

Compared to serial scan, RAS techniques reduce power dissipation during load and unload procedures. However, power reduction during capture procedures is not addressed. System-On-Chip (SOC) circuits are typically separated into several design blocks, each of which represent a physical block separately implemented to overcome capacity limitations of software tools for physical implementation. Some RAS techniques localize 2-D physical grid layouts to each physical block of the SOC. However, this approach does not fully mitigate the high wiring area overhead of large physical blocks. Use of MISR for compression without having a mechanism to observe individual capture data degrades test coverage in the presence of unknown (X) values for large circuits which typically contain a few circuit blocks that would need precise calculation/simulation of signal values. Test automation tools are incapable of providing such precision and thus unknown (X) values are inevitable on such blocks.

The authors have introduced [10] hierarchical, compressed random access scan (CRAS-N) architecture, and a more constrained version named Segmented Random Access Scan (SRAS) which reduces test data volume (TDV), test application time (TAT) and routing overhead. In particular, the CRAS-N DFT technique provides a multi-dimensional addressing scheme without upper limits on the number of hierarchy levels in the test access structure of the design. SRAS simplifies the implementation of CRAS-N by keeping the number of scan cells or groups of scan cells addressed at each level of test hierarchy thesame. It also constrains the number of scan cells directly addressed to be a power of 2 for easy implementations.

The rest of this paper is organized as follows. Section 2 presents the CRAS-N and SRAS architectures while individual subsections divulge details and block diagrams of individual components in those two architectures. Additionally, subsection 2.6 compares the area overhead of different SRAS schemes with commercially available serial scan compression scheme for a representative test circuit. Section 3 providesa brief conclusion.

2. COMPRESSED RAS-N AND SRAS DFT SCHEMES

The CRAS-N scheme is described next having a 2-level hierarchy identified as a CRAS-2 scheme, where design is partitioned as shown in Figure 1 with an example partition of 4 blocks. Each top and lower-level block contains its own Demultiplexer-Decompressor-Multiplexer-Compressor (DDMC) circuit block while the top level DDMC (marked #5 in Figure 1) interfaces chip level scan pins to lower level DDMC blocks. DDMC-2 decodes Row Address (RA) and generates signals to select one or more of DDMC circuit blocks in the four (4) quadrants. The Column Address (CA) is broadcast to DDMCs 1-4, and a Column Address Decoder (CAD) in each DDMC generates select signals for one or more scan cells in a respective quadrant. ATPG tools control CA and RA in each test pattern. Decoded CA lines are routed only within a single quadrant while decoded RA lines are routed from DDMC #5 to the quadrant level DDMC blocks thereby limiting the routing congestion. This CRAS-2 DFT technique reduces wiring area overheads of the CRAS DFT technique. Typically, RA, CA MODE signal, scan input (SI) and scan output(SO) are serially accessed to minimize the number of test pins at the top (i.e. IC chip) level.

Figure 2 is an abstract block diagram of a design under CRAS-3 scheme with 3 levels of hierarchy wherethe circuit has been partitioned into 4 sub-circuits with each sub-circuit having its own DDMC block marked A - D in Figure 2. Each sub-circuit has been further partitioned into sub-sub-circuits with each sub-sub-circuit having its own DDMC block marked 00-03, 04-07, 08-11 and 12-15 in Figure 2. A top level DDMC block marked “$” is used to interface at chip level

scan pins to lower level DDMC circuit blocks. DDMC $ decodes a Page Address (PA), not shown in Figure 2, and generates select signals for the DDMC blocks in the four quadrants A-D. Row address, RA is broadcast to each DDMC block in the four (4) quadrants A - D. Each DDMC block decodes RA and generates select signals for those DDMC blockswhich are in the respective sub-quadrants, i.e., DDMCs 00-03, 04-07, 08-11 and 12-15. The Column Address, CA (not shown) is also broadcast to DDMCs 00-03, 04-07, 08-11 and 12-15. Each of DDMCs 00-03, 04-07, 08-11 and 12-15 blocks decodes CA to generate select signals for scan cells in the respective sub-quadrant. ATPG tools control CA, RA and PA for each test pattern.

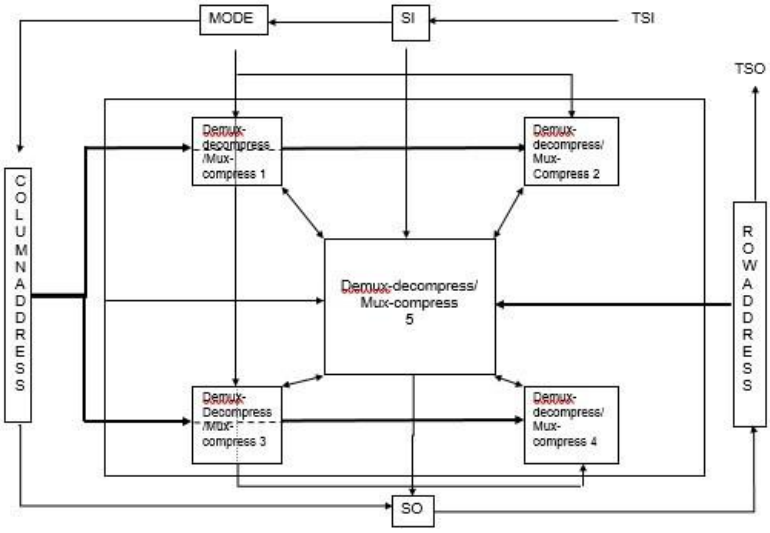

Figure 1. Block diagram of CRAS-2 architecture

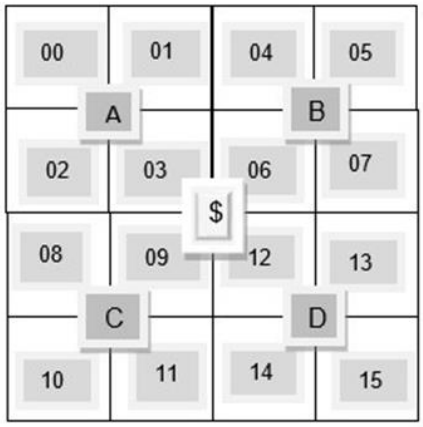

Figure 2. Block diagram of CRAS-3 architecture

In the CRAS-2 scheme (Figure 1), decoded CA lines need to traverse only within a single quadrant. However, in CRAS-3 scheme (Figure 2), decoded CA lines traverse only within a single sub-quadrant. Also, in CRAS-3 scheme, decoded RA lines traverse only the area occupied by a single quadrant while decoded PA lines are routed from the center ($) to quadrant level DDMC blocks, thereby limiting maximum wire lengths and congestion. This demonstrates that CRAS-3 scheme further reduces wiring area overhead of CRAS-2 DFT scheme. As explained, this new CRAS-N scheme provides a multi- dimensional addressing scheme without upper limits (of N) on the number of dimensions in the logical structure of the design. A hierarchy having N

levels of DDMC blocks is referred to herein as CRAS-N. Also, to maintain flexibility, the number of DDMC blocks at each hierarchy level need not be the same. However, for implementation convenience in semiconductor technology, the number of blocks at each level of the hierarchy and the number of storage elements under the lowest block in the hierarchy are made the same. That number is referred to as *segment_size*. Such constrained implementations of CRAS-N schemes are named as Segmented, Random Access Scan (SRAS) technique. Hence, for a circuit with a given number of scan cells, the number of levels in the test access hierarchy and the *segment_size* are inversely related.

As depicted in Figure 3, CRAS-3 technique can alternatively be represented as a graph with vertices (rectangles in Figure 3) denoting DDMC blocks, and edges (straight lines) denoting the hierarchical relationship of DDMC blocks. This represents an SRAS-3 scheme in the form of a tree structure hierarchy having three (3) levels, and a segment size of 4 which implies that CRAS blocks are arranged in 4 units at each hierarchy level. Leaf Scan Groups (marked LSG0 through LSG15) correspond to DDMC blocks marked 00 through 15 in Figure 2. Since the *segment_size* is 4, each LSG block in Figure 3 contains four (4) scan cells, which are numbered 0 – 63. Also, the Intermediate Scan Groups (marked ISG0 - ISG3) correspond to DDMC blocks marked A through D in Figure 2 while the Top Scan Group (marked TSG) corresponds to the top level DDMC block marked $ in Figure 2. The CA, RA, PA, and MODE inputs are not shown in Figure 3. The top level ISG in the hierarchy is named a Top Scan Group (TSG). The hierarchical addressing scheme can be represented by a rooted, full, N-ary tree, where N is the *segment_size*, while the *dimension_count* is the depth of the tree (i.e., the number of levels in the hierarchy). The cell area overhead of SRAS technique can further be traded off against routing area overhead by optimizing *dimension_count* and *segment_size* for different types of designs.

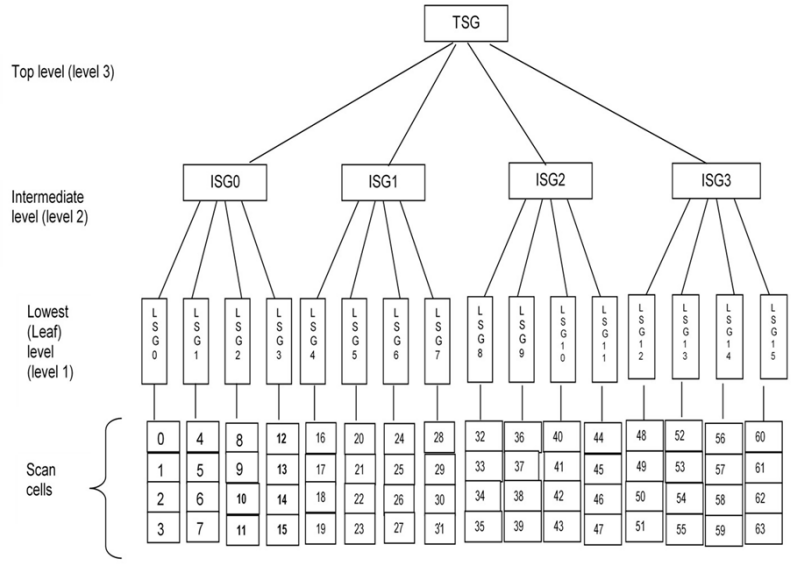

Figure 3. Hierarchy diagram of CRAS-3 segment 4

In an ATPG tool flow, only a small fraction of scan cells needs to be loaded (unloaded) in most scan test patterns. However, in the few starting ATPG test patterns, almost all the scan cells require load/unload. In such a case, RAS architectures fare worse than serial scan DFT in test application time (TAT) since each address of scan cell needed to be shifted in. This is mitigated by adding an Address Increment (AI) mode of operation in which the scan cell address (for example, CA, RA, and PA for CRAS-3) is incremented automatically, without external control, in

each load or unload cycle. In AI mode, a new scancell value is loaded (unloaded) in each clock cycle similar to serial scan method. This mode provides the lowest test time for initial ATPG vectors in which almost all the scan cell values are specified.

In the CRAS-N architecture, multiple compression and decompression modes are implemented in DDMC blocks of Figure 1 to further reduce TAT and TDV over single level compression-decompression schemes.Hence, in SRAS architecture, an LSG, an ISG and a TSG have seven (7) modes of operations: (a) normal(Functional); (b) Address Increment (AI); (c) Test Bypass (TB); (d) One Address (ON); (e) Less Than or

Equal Address (LE); (f) Greater Than Address (GT); and (g) All Except Address (AE). The ON, LE, GTand AE modes of operation are collectively referred to as *shift* modes which can be used to enable ATPG software to choose the most efficient scan-compression method for each test pattern it generates. The selected compression method should prevent any unknown value (X) from being tolerated to make test signatures consist only of known values. If a high percentage of scan cells needs to be loaded (unloaded)in a particular test, AI mode is enabled, and the internal scan address register can be incremented from a minimum to a maximum value while applying the scan input data and observing the scan output data at chip pins in each test clock cycle. This mitigates the test application time penalty of RAS compared to serial scan for loading/unloading a large percentage of the scan cells.

If a test vector focuses only on a well partitioned portion of the circuit, the rest of the circuit is kept in theTest Bypass (TB) mode to reduce shift length which reduces TAT and test power. To load/unload only one scan cell, the scan cell address and scan input data is shifted in, and the scan output data is shifted out. This is One Address (ON) mode operation. The LE mode is selected under several constraints. All the scan cells requiring load need the same value (all 0 or all 1). All addresses of those cells are less than the targeted scan cell address, ADDR, and their current data values (to be unloaded) do not contain any unknowns (X) as verified by the ATPG tool. Then the largest address of all the controlled/observed scan cells and scan input data is shifted in, and the exclusive-OR of all the states of all the scan cells at an address that is less than or equal to the specified address (ADDR) is shifted out (unload). This LE mode sets or resets all scan cells whose address is less than or equal to ADDR (load). In contrast, GT mode setsor resets all scan cells whose address is greater than ADDR (load).

If all the scan cells to be controlled in a test pattern need the same value (i.e., all 0 or all 1), and if only one SRAS scan cell in an LSG has an unknown value in a particular test, the address of that scan cell withX value and the scan input data is shifted in, and the exclusive-OR of all the states of all scan cells except that the one with ADDR is shifted out (unload). This All Except Address (AE) mode of operation sets or resets all scan cells except the one with ADDR (load). For simplicity, the compression mode is also shifted in/out along with scan address and load/unload data values. Each shift operation can correspond to a load, unload, or simultaneous load-unload operation which reduces TAT. However, this is not always possible due to conflicting scan addresses and compression mode between consecutive patterns. In addition, multiple load/unload operations may be needed to load and unload all relevant scan cells in a test pattern in the presence of unknown (X) values. For example, if scan cells with address value 10 through 20 are unknown (X), and all other scan cells need to be observed in the current test pattern, GT mode with addressvalue 20 is applied to observe scan cells 21 and above, followed by LE mode with address 9 is used to observe scan cells 0 to 9. Also, cells requiring different load values need multiple load operations for thattest pattern. In the worst case for TAT, ON mode would be applied to load or unload each specified scancell in a test pattern.

2.1. SRAS Implementation

The implementation of CRAS-N and SRAS schemes is described next. Normal flip-flops are replaced by SRAS cells. As described earlier, the SRAS hierarchy consists of LSGs, ISGs and TSGs blocks. An LSG includes: (a) a Segment Selector Block (SSB); (b) a Segment Accumulator Block (SAB); (c) an optional Segment Data Multiplexor (SDM) and (d) a Segment Control Block (SCB). In LSG block, SSB selects one or more scan cells from a group of *segment_size* (= 4 in Figure 3) to load/unload based on its segment address and compression mode inputs. In LSG, SAB accumulates values from a *segment_size* SRAS cell group while SCB controls the operation of that cell group. The optional SDM block reduces the area overhead of SSB/SAB by partially multiplexing/demultiplexing SRAS cell data before being fed to/from the SSB/SAB. In essence, SCB, SSB, optional SDM, and SAB implement DDMC blocks in Figure 1. A SRAS cell group, an SSB, an SAB, an SCB, and an optional SDM implement a single LSG block. An SSB, an SAB, an SCB, an optional SDM, and a group of LSGs implement an ISG under the hierarchy scheme in Figure 3. The TSG block contains a group of ISGs, an SSB, an SAB, an optional SDM and an SCB. Here, SSB, SCB, SAB, and optional SDM together constitute a Segmented Random Access Scan Controller (SRASC). The same SRASC can implement LSG, ISG and TSG in a particularly large design. SRASC, when custom implemented, reduces area overhead, and is reusable in many designs.

Table 1. Optimized RAS scan cell port list

No	Pin	In/out	Description	Function
1	D	In	Functional data	Functional data in
2	CLK	In	Functional Clock	Functional data capture clock
3	SI	In	Scan In	Test data in
4	TCK	In	Test Clock	Test data capture handling clock
5	Q	Out	Functional output	Functional data out
6	CS	In	Cell Select	Random access scan (RAS)
7	OE	In	Observe Enable	Optionally disable test output for power reduction
8	CO	In	Control Override	Optional override cell selects for circuit initialization
9	OO	In	Observe Override	Optionally override cell select for signature accumulation
10	OS	In	Output Select	Optionally bypass SRAS cell for race paths
11	TQ	Out	Test Output	Test data out

Figure 4 depicts a unique RAS scan cell optimized for SRAS architecture with special features to support silicon debug. Its ports are listed in Table 1 showing a separate test clock (TCK) and a select (CS) line. It also has optional signals such as observe enable (OE), control override (CO), observe override (OO), and output select (OS) to reduce power dissipation and facilitate debug. However, for ease of implementation, a commonly available multiplexed scan D flip-flop (Figure 5) with a capture enable input (CE) may be used as the RAS cell. It reduces efforts for SRAS implementation on real world designs since it is readily available in commercial standard cell libraries.

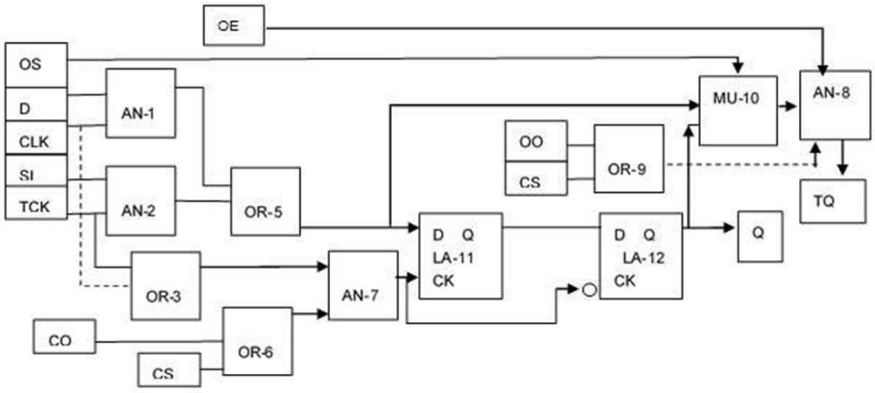

Figure 4. RAS scan cell block diagram

A multiplexed scan D flip-flop with capture enable has lower area overhead than an optimized SRAS scan cell. Ports of the cell are listed in Table 2. It consists of two multiplexers and a flip-flop which captures D input when CE=1 and CS=0 and captures SI when CS=1. The cell holds state, Q value when CE=0 and CS=0. Note that the ports of the multiplexed scan D flip-flop with capture enable are a subset of the optimized SRAS cell except for the CE signal which is 1 during functional operation (and scan capture) but stays 0 otherwise.

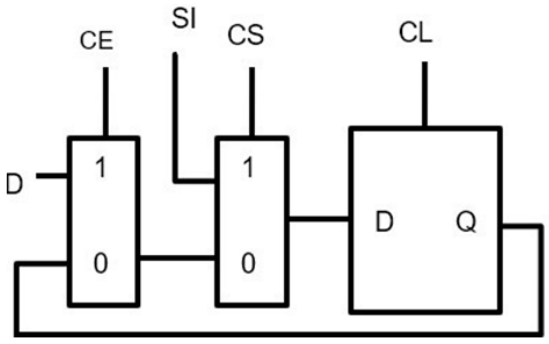

Figure 5. Multiplexed D flop with capture enable port.

Table 2. Ports of multiplexed scan flip-flop

No.	Port	Direction	Description
1	CLK	Input	Clock signal
2	D	Input	Data Input
3	Q	Output	Data Output
4	SI	Input	Scan Input Data
5	CS	Input	Scan Capture Enable
6	CE	Input	Data Capture Enable

2.2. Segment Selector Block (SSB)

The detailed SSB is shown in Figure 6, while Table 3 lists its ports and functions. An SSB implements a test control path in a TSG/ISG/LSG hierarchy while an SAB implements a test data path in the TSG/ISG/LSG path. The SCB contains all the control registers for different modes of TSG/ISG/LSG operations, switches test input from higher to lower hierarchy levels, and further switches test output data from lower to higher hierarchy levels. The left-most AND block in

Figure 6 is not needed if optimized SRAS cell is used since TQ of a de-selected cell has value 0. In Figure 6, the COM signal selects one SRAS cell using the ORTREE block and compresses multiple SRAS cell data using the XORTREE. The INV port driven by the SCB optionally inverts multiplexer output to produce "one cold data" word (such as 1101111) at SELD outputs. The INV signal to SSB produces SELD output that selects all but one SRAS cell and executes the exclusive-or operation on all SRAS cell data excluding the addressee cell. The Decoder block, multiplexer and ORTREE block implement a data compressor corresponding to the DDMC in Figure 1.

Table 3. Ports of Segment Selector Block (SSB)

No	*Pin*	*Description*
1	ENA	Input bit; Selector Block Output Enable signal
2	SAD	Segment Address Register, an encoded bit vector
3	COM	Input bit, Compress mode signal (together with INV, provides address modes)
4	INV	Input bit; Invert decode signal (together with COM, provides address modes)
5	SELD	Segment Select output, a decoded bit vector of an SAD register value

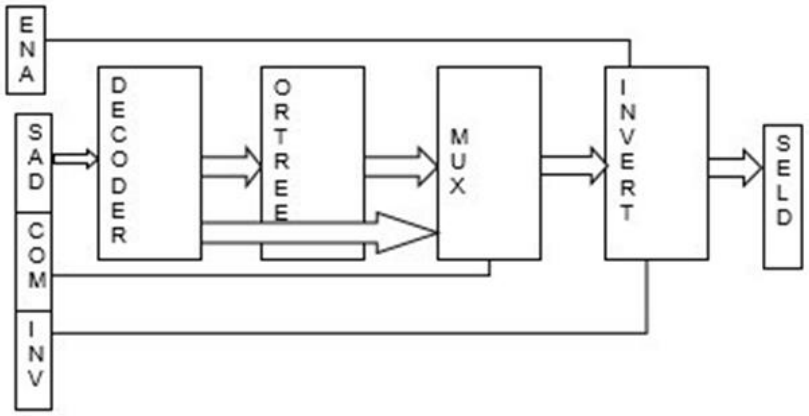

Figure 6. Segment Selector Block (SSB) diagram

2.3. Segment Accumulator Block (SAB)

The SAB is shown in Figure 7, while Table 4 lists its ports and functions. Signal TQ is the test data output vector from scan cells in LSG, or ISG provided scan output data to be sent to chip output via TSG. The SAB accumulates TDO data from lower hierarchy levels (SRAS, LSGs or ISGs) into a single bit value, and feeds it to next higher level (ISG or TSG). Reduction-XOR compresses multiple SRAS scan cell TQ outputs to a single bit for TDO output port. The SAB implements LE, GT, or AE operational modes. ENA signal may disable TDO to be 0 value using Test Bypass (TB) so the circuit can operate in normal functional mode. As seen in Figure 7, Reduction-XOR block acts as a data compressor, and Reduction- OR acts as multiplexer logic. They correspond to DDMC in Figure 1.

Table 4. Ports and Functional Modes of SAB

No.	*ENA* (output enable signal)	*INV* (invert decode signal)	*COM* (compress mode signal)	*TDO* (1-bit scan data output signal)	*Modes*
1	0	X	X	0	Functional/Test Bypass (TB)
2	1	0	0	OR all TQ	One Address (ON)/ Address Increment (AI)
3	1	0	1	XOR all TQ	Less than or Equal (LE)
4	1	1	0	XOR all TQ	All Except (AE)
5	1	1	1	XOR all TQ	Greater Than (GT)

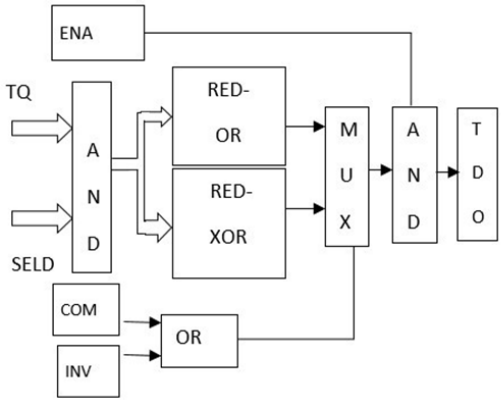

Figure 7. Segment Accumulator Block (SAB)

2.4. Segment Control Block (SCB)

Figure 8 shows SCB while Table 5 lists its ports. All are single bit lines, except for the shaded arrows. The functions of the SCB is to: (a) provide AI mode by configuring Segment Address Data (SAD) to operate as part of a distributed scan address counter; (b) control SSB and SAB using shift registers accessible fromSCB in the parent ISG or from chip pins; (c) daisy chain the shift registers of all SCBs in LSGs/ISGs thathave a common ISG parent; (d) optionally splice the daisy chained shift registers of all SCBs in LSGs/ISGs immediately below it in the hierarchy using the unused SI register in the ISG as a control bit to implementTest Bypass (TB) mode; and (e) turn off functional clocks labelled as CLKO output port in unused LSGsor ISGs (i.e., LSGs or ISGs that are not pulsed in the current test pattern) to reduce power dissipation during a capture procedure with the help of a clock gating signal (SCG) and a clock gating cell (CG). The rightmost bit in a 3-bit SSC register is referred to as the SCG, which controls the CG cell.

In AI mode, each scan cell is consecutively addressed using a distributed counter comprised of SAD counters in all SCBs of the circuit. A carry input signal (CIN) vector from LSGs/ISGs in the immediatelylower hierarchy of this SCB is applied as input to a logical-OR block since each bit in CIN enables the address counter in individual child blocks. An address increment input (AIN) signal which is a global enable signal for the AI mode; an enable (ENA) signal which is active when the parent ISG selects the LSG/ISG containing the SCB; and output from the OR circuit are applied as inputs to SAD counter in SCB. For SRAS scheme, *segment_size* is chosen as 2^N where N is the SAD width. The SAD in the SCBfeeds the SAD port of SSB as in Figure 8.

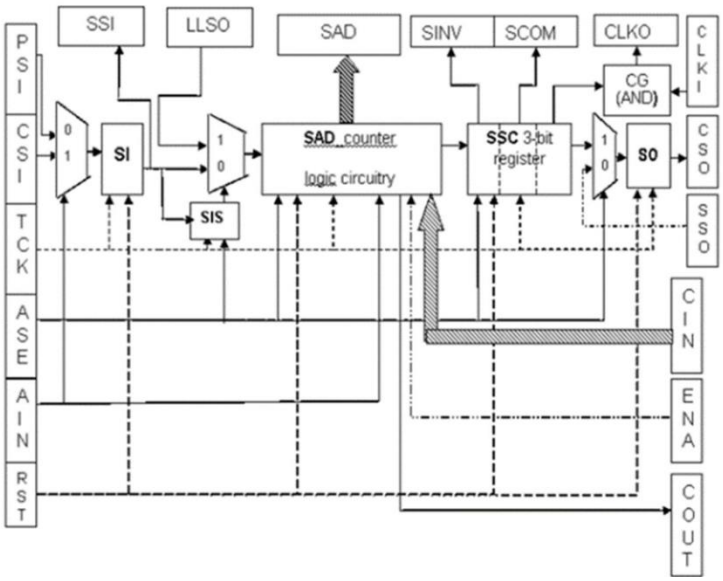

Figure 8. Segment Control Block (SCB) diagram

When the SAD counter in the first LSG reaches the last count value of all 1's, it generates a COUT pulse incrementing the SCB in its parent ISG which disables the first LSG and enables the next LSG. This process repeats until the last LSG under the first ISG completes counting (all 1), upon which the SCB in the first ISG generates a COUT pulse to its parent ISG. At that point the SAD counter in the SCB of the second level ISG increments disabling the first ISG and enabling the next ISG. This process repeats until all the SAD counters in all SCBs in all LSGs finish counting. This architecture enables the construction of a distributed scan address counter by interconnecting SCBs in a hierarchical tree structure as illustrated in Figure 3.

Table 5. Ports of Segment Control Block (SCB)

No.	***Pin***	***Direction***	***Description***
1	TCK	Input	Test clock
2	CSI	Input	Channel Scan Input
3	CSO	Output	Channel Scan Output
4	RST	Input	Test reset global input
5	SSI	Output	Segment Scan Input
6	SSO	Input	Segment Scan Output
7	ASE	Input	Address Shift Enable signal
8	AIN	Input	Address Increment signal
9	CIN	Input	Address count enable bit vector
10	SAD	Output	Segment Address Data vector
11	SINV	Output	Segment Invert Data
12	SCOM	Output	Segment compression mode signal
13	COUT	Output	Address Count enable output
14	LLSO	Input	Last Leaf Scan Out
15	ENA	Input	Block enable
16	PSI	Input	Previous Segment Input
17	CLKI	Input	Functional clock input
18	CLKO	Output	Functional clock output

2.5. Segment Data Multiplexor Enhancement (SDM)

We present below another enhancement to incorporate a Segment Data Multiplexer (SDM) block at the hierarchy scan group segments (LSG, ISG and TSG) which drastically decreases area overhead of the original architecture to practically implementable levels on commercial circuits. The added area of SDM is more than compensated for by the test area reductions in SSB and SAB. The worst-case TAT might increase by the multiplexing factor since the scan cell group in LSB is now accessed with time multiplexing. A power of 2 (2, 4, 8, 16, 32 etc.) is suggested for the multiplexing factor for ease of implementation. The SDM block is detailed in Figure 9 while Table 6 describes its ports. The use of an SDM enables SAB/SSB width to be reduced by a factor of 2^{MSBW}, where MSBW is the width of the SADMport of the SDM. The area of the SAB and SSB is correspondingly reduced. Figure 10 shows an LSG with 2-way SDM controlling multiplexed scan flip-flops with CE feeding through an enhanced SAB. Figure 11 illustrates the placement of an SDM between SSB, SAB, and the scan cell array inside LSG, which reducesdata vector width of both SSB and SAB by the multiplexing factor.

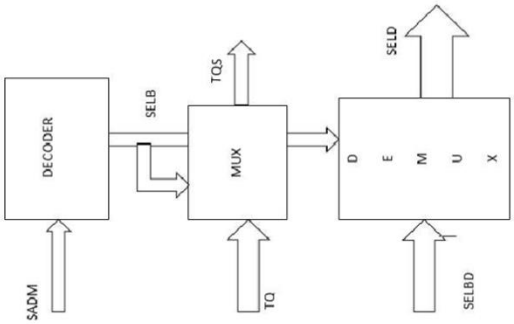

Figure 9. Segment Data Multiplexor (SDM)

For example, for an SRAS-8 architecture, where 256 scan cells are controlled by one LSG, the area overhead of the SAB/SSB can be reduced by a factor of 8 by using the 3 most significant bits of the Segment Address Data (SAD) to control the SDM and using only the least significant 5 bits of the SAD to control the SAB/SSB. This area decrease is offset by the area of the SDM itself. The SSB is the largestblock in architecture due to the presence of the reduction XOR logic blocks. This provides a way to achieve a significant net reduction in the overall area overhead of SRAS architecture by introducing the SDM into LSG/ISG blocks. A corresponding reduction in the overhead of test compression circuits is possible by employing a smaller XOR network. In the SRAS-8 architecture with SDM-3, a minimum of 8 unload operations are needed to observe all 256 scan cells under an LSG, since the XOR based data reduction factor in SSB is only 32 ($=2^5$). This can be achieved without an SDM, but with a single unload operation since a 256 (2^8) bit XOR network is available in the SSB.

Table 6. Ports of Segment Data Multiplexor (SDM)

No.	Port	Description
1	SADM	Most significant 1-4 bits of Segment Address
2	TQ	Test Data
3	TQS	Selected Test Data
4	SELBD	Decoded select lines within selected bank
5	SELD	Global decoded select lines

Figure 11 shows the hierarchical interconnections among ISGs and LSGs for the same 4x4 segment SRAS scheme for which the LSG was shown in Figure 9. Note that there are 2 levels of ISGs in this implementation corresponding to a total dimension of the test access to be 4 (LSG, ISG1, ISG2, TSG). The address increment (AI) mode of operation using the distributed address counter can be studied by tracing the CIN and COUT connections in Figure 11.

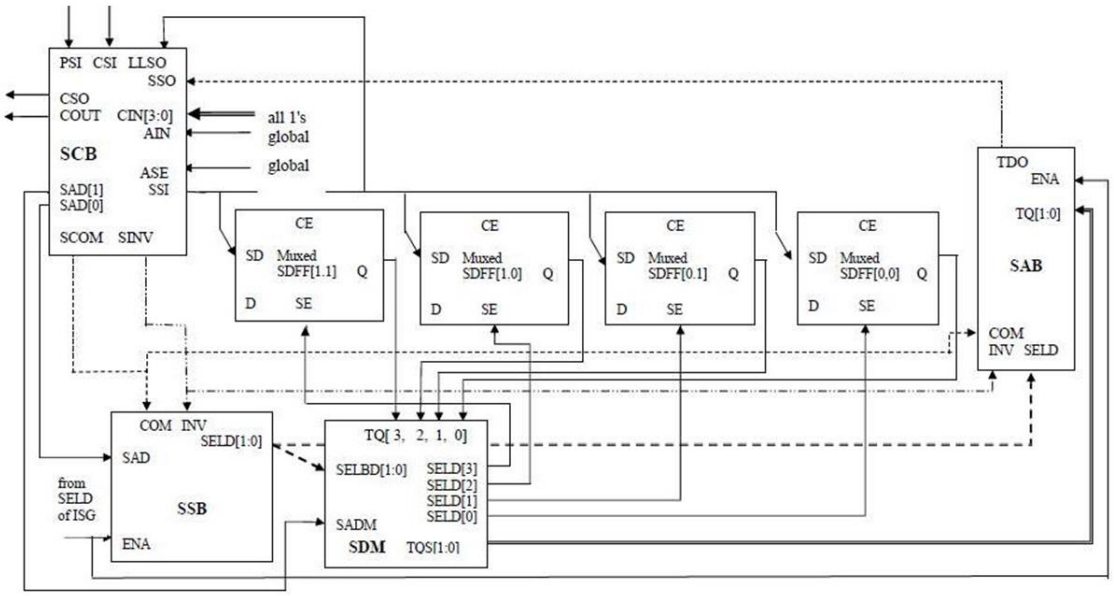

Figure 10. LSG with 2-way SDM connected to enhanced SAB and multiplexed scan cells.

2.6. Comprehensive analysis of the CRAS schemes

A comparative analysis of the area overhead of CRAS scheme with various serial scan technologies is presented in this section. A Register Transfer Level (RTL) description of a 256x256 random access flip- flop array was used as a test circuit. The RTL description of the design is shown in Figure 12. An SRAS- 8 scheme with one LSG per 256 scan cells, was implemented for this design. Also, a CRAS-8 was implemented for the test circuit in only two levels of hierarchy (LSG and ISG) along with a single TSG block without an SDM. Another experiment was performed with the enhancements described in this paperusing an 8-way SDM scheme named CRAS-8-8. The area overhead of CRAS-8 and CRAS-8-8 were compared with non-scan, serial scan, and compressed serial scan implementations. The scan overhead is atypically high since ram_256x256 is mostly comprised of flip-flops. Therefore, the combinational logic area was normalized to be four times sequential area of the non-scan circuit as is typically the case. Industrystandard compressed serial schemes and scan insertion tools were used in the design synthesis flow. In the serial scan compression experiments, a compression ratio of 32 was used (8 scan channels).

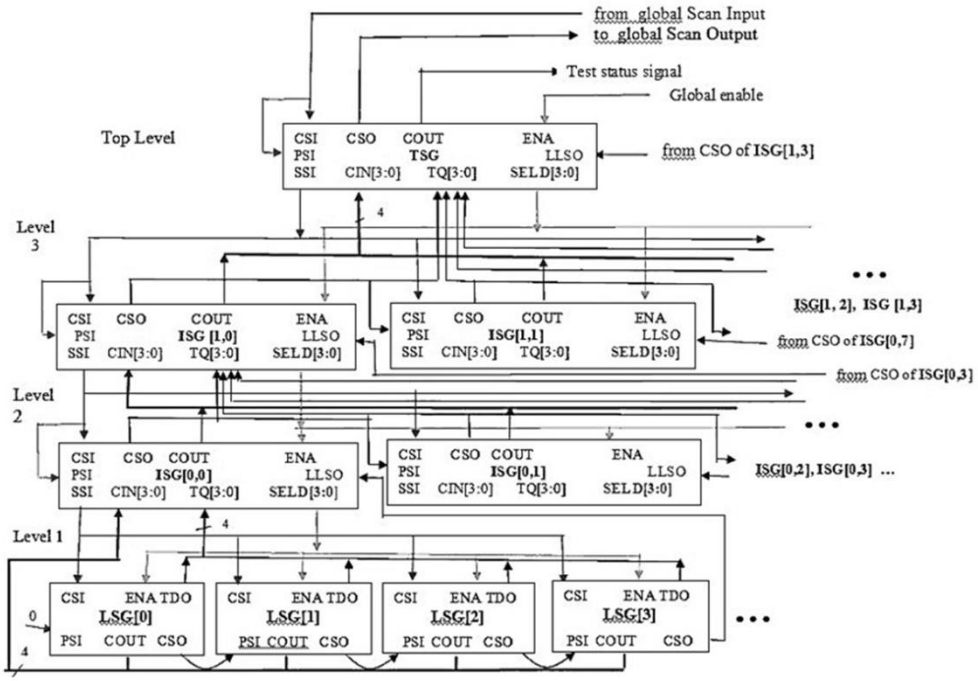

Figure 11. Hierarchical interconnection diagram of TSG, ISGs, and LSG blocks.

Table 7 provides the experimental data. The normalized area overhead is 72.65% for CRAS-8, and 40.6% for CRAS-8-8. Area overhead is still very high for RTL based implementations, but 8 channel CRAS is effective in reducing area overhead by nearly half. We expect that custom implementation of the SRASC(Segmented Random Access Controller containing SSB, SCB, SDM and SAB) to reduce the area overheadof SRAS to levels comparable to compressed serial scan.

Table 7. Area overhead comparison of scan schemes for a register file of 256x256 flops

Area (normalized)	Non-scan	Serial Scan	Serial scan Compress1	Serial Scan Compress2	CRAS-8	CRAS-8-8
Combinational	85740	88151	89062	93142	373591	214893
Non combinational	102236	127795	127795	128552	185767	180637
Normalized Combinational	408944	411355	412266	416346	696795	538097
Total area	187976	215946	216857	221664	559358	395530
Total normalized area	511180	539150	540061	544868	882562	718734
Raw area overhead %	0	14.88	15.36	17.92	197.56	110
Normalized area overhead %	0	5.47	5.64	6.59	72.65	40.6

```
module ram_256x256( output  [255:0] dout, input   [255:0] din, input   [7:0]   add, input    clk, we);
reg  [255:0] array[255:0];  integer      i;  genvar i;
generate
  for(i=0; i < 256 ; i=i+1) begin : FLOP_ARRAY
    always @(posedge clk) begin
      if((we == 1) && (add == i))array[i] <= din;
    end
  end
endgenerate
assign dout = array[add];
endmodule
```

Figure 12. Verilog description of RTL circuit under test

3. CONCLUSION

After thoroughly surveying different RAS architectures available in the published literature, a new CRAS-N/SRAS-N architecture to eliminate the high routing area overhead of traditional RAS was proposed. The implementation of SRAS-N architecture was described next in this paper. Finally, results comparing the area overhead of SRAS-8 architecture were presented after using commercially available scan compression tools on a simple experimental circuit. Careful selection of the segment size (256 in the experiment) and multiplexer width (8 in the experiment), along with custom implementation of SRASC was used to reduce area overhead of SRAS to levels comparable to compressed serial scan (about 20% overhead). The dream of providing the benefits of RAS (low test power and reduced test application time) to mainstream IC designers while keeping area overhead comparable to the latest compressed serial scan techniques is closer to reality with this work.

REFERENCES

[1] Ando, H, (1980) “Testing VLSI with Random Access Scan”, Digest of Computer Society International C o n f e r e n c e (COMPCON), pp. 50-52.

[2] Wagner, K, (1983) “Design for Testability in the Amdahl 580”, Digest of Computer Society International C o n f e r e n c e (COMPCON), pp. 384-288.

[3] Ito, N., (1991) “Automatic Incorporation of on-Chip Testability Circuits”, Proceedings of Design Automation Conference (DAC), pp. 529-534.

[4] Baik, D, et al, (2004) “Random access scan: A solution to test power, test data volume and test time,” Proceedings of VLSI Design, pp. 883-888.

[5] Hu, Y, et al,(2006) "Compression/Scan Co-Design for Reducing Test Data Volume, Scan-in Power Dissipation and Test Application Time", IEICE Transactions on Information and Systems, Vol. E89-D, No. 10, pp. 2616-2625.

[6] Lin, C , e t a l , (2 0 0 5) “ A Cocktail Approach on Random Access Scan toward Low Power and High Efficiency Test,” Proceedings of International Conference on Computer-Aided-Design (ICCAD), pp. 94-99.

[7] Baik, D, and Saluja, K, (2005) “Progressive Random Access Scan: A simultaneous solution to test power, test data volume and test time”, Proceedings of International Test Conference (ITC), no. 15.2, pp. 1-10.

[8] Mudlapur, A, et al, (2005) “A Random Scan Architecture to Reduce Hardware Overhead”, Proceedings of International Test Conference (ITC), no. 15.1, pp. 1-9.

[9] Hu, Y, (2008) “Localized Random Access Scan: Towards Low Area and Routing Overhead,” Proceedings of Design Automation Conference (ASPDAC), pp 565-569.

[10] Oomman, B, and Karunaratne, M (2016) “Design-for-test techniques for a digital electronic circuit,” US patent #9423455, August

[11] Johnson, S, (1967) “Hierarchical Clustering Schemes”, Psychometrika, Vol. 32, No. 2, pp. 241-254.
[12] Van de Goor, A (1991) “Testing Semiconductor Memories: Theory and Practice”, Chichester, UK: John Wiley & Sons, Inc.,

AUTHORS

Maddumage Karunaratne earned his Ph.D. degree in electrical engineering from the University of Arizona, Tucson, AZ. Before joining Pitt-Johnstown in 2004, he gained fourteen years of industry experience working in the semiconductor industry performing software development, application engineering, design, testing and verification of digital integrated circuits. His research interests are in design and test automation, power analysis, and software development. He has authored numerous publications and holds several US patents. He can be reached at maddu@pitt.edu.

Bejoy Oomman founded Genesys Testware test automation and consulting firm specializing in embedded memory test and self-repair. Over 30 years, he has been working in engineering and management positions in various test automation and systems design companies as a design and test engineer. Mr. Oomman has a MS in Computer Engineering from the University of Massachusetts. He can be reached at bejoygoomman@aim.com.