Localized Random Access Scan: Towards Low Area and Routing Overhead

Yu HU, Xiang FU, Xiaoxin FAN

Key Laboratory of Computer System and Architecture
Institute of Computing Technology, CAS
Beijing 100080, CHINA
e-mail: {huyu, fuxiang, fanxiaoxin}@ict.ac.cn

Hideo FUJIWARA

Graduate School of Information Science
Nara Institute of Science and Technology (NAIST)
8916-5 Takayama, Ikoma, Nara 630-0192, JAPAN
e-mail: fujiwara@is.naist.jp

Abstract: Conventional random access scan (RAS) designs, although economic in test power dissipation, test application time and test data volume, are expensive in area and routing overhead. In this paper, we present a localized RAS architecture (LRAS) to address this issue. A novel scan cell structure, which has fewer transistors than the multiplexer-type scan cell, is proposed to eliminate the global test enable signal and to localize the row enable and the column enable signals. Experimental results on ISCAS'89 and ITC'99 benchmark circuits demonstrate that LRAS has 54% less area overhead than multiplexer-type scan chain based designs, while significantly outperforms the state-of-the-art RAS scheme in routing overhead.

I. Introduction

Scan design is one of the most widely used design-for-testability (DFT) techniques, which reduces the complexity of automatic test pattern generation (ATPG) for sequential circuits. However, scan design faces the challenges of long test application time and high test power dissipation because of its serial shifting nature. To cope with these problems, various approaches have been proposed, such as scan structure alteration, test data compression and blocking logic insertion. Among these approaches, random access scan (RAS) designs [1]-[9] have been proved capable of simultaneously reducing test time, test data volume and test power.

RAS was firstly proposed by Ando [1] and then applied in Amdahl 580 by Wagner [2] and in Fujitsu VP-2000 by Ito [3]. Unlike scan chain design that test data are serially shifted in and out of scan chains, RAS is similar to the random access memory (RAM) design that each memory element can be randomly and uniquely addressed. In RAS, only one scan flip-flop is toggled at any clock cycle of loading test stimuli, thus test power dissipation is drastically reduced. To reduce test application time and test data volume, test vector ordering and X-identification techniques [4][5], compression/scan co-design approach (CSCD) [5], and deterministic scan reseeding [7] method were proposed.

Although RAS designs are economic in test application time, test data volume and test power dissipation, their hardware overhead, especially routing overhead, are prohibitively high. Contrary to the regular structure of RAM, scan flip-flops of RAS are randomly distributed all over the circuit, hence row-enable signals and column-enable signals, which can be short word lines and bit lines in RAM, now become long global wires in RAS.

To cope with the routing overhead of RAS, two approaches were proposed recently. Progressive random access scan (PRAS) [8] utilized a structure similar to static random access memory (SRAM), which helped to achieve smaller area overhead and routing overhead than the structure of their previous work [4]. Authors of [9] designed a toggle scan flip-flop structure (Toggle RAS for short) that could eliminate two global signals "Scan-In" and "Scan-Enable" that were used in the conventional RAS structure, thereafter reduced the routing overhead. However, no experimental results were given by [9] to show routing overhead of Toggle RAS.

While [8] and [9] tried to reduce routing overhead in some extend, they did not directly address the irregular structure of RAS. As row-enable signals and column-enable signals connected to scan flip-flops are globally routed, there are tens of such long wires for a circuit with thousands of flip-flops. Therefore, shorten the wire length of these enable signals is very critical to reduce routing overhead. Moreover, if refer to the physical DFT synthesis flow using scan chains, we can see scan flip-flops are reordered according to the layout information [10]. Consequently, the length of stitching wires between scan flip-flops is reduced an order of magnitude. The reorder step in the DFT flow highlights the necessary to consider the layout information.

In this paper, we propose a layout-aware design approach named as Localized Random Access Scan (LRAS) to eliminate the global test enable signal and to localize the row and the column enable signals. The proposed solution adopts two-pass DFT synthesis flow. First, physical synthesis is performed on Register-Transfer-Level (RTL) description of the circuit. During this procedure, the coordination of flip-flops is dumped out. Next, flip-flops are grouped based on their coordination by an agglomerative clustering algorithm. Afterwards, the flip-flops are replaced with LRAS scan cells while the flip-flops within a group are assigned to

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

their group address decoder. Finally, we synthesize the modified RTL codes of the circuit for conducting placement and routing. By considering layout information in the synthesis flow, the DFT designer is able to successfully meet testability goals with minimum hardware penalty.

The remainder of the paper is organized as follows. In the next section, we analyze the conventional RAS architecture to show the causes of high hardware overhead. In Section 3, we present the general architecture of LRAS and structure of the scan cell. Section 4 gives an agglomerative clustering algorithm to group LRAS scan flip-flops. In Section 5, experimental results obtained on the ISCAS'89 and ITC'99 benchmark circuits are reported and discussed. Finally, Section 6 concludes the paper.

II. Preliminaries

Fig. 1 illustrates a conventional RAS architecture and its scan cell structure. In general, RAS has two major parts: the one is address decoders, which usually contains a row address decoder and a column address decoder, the other part is scan cells organized as a two-dimensional array. Addresses are input to decoders through column address (*CA*) bus and row address (*RA*) bus. Row enable signal (*RE*) and column enable (*CE*) signal, as well as test enable (*TE*) signal, together determine whether a scan cell is written a test stimuli bit through scan-input (*SI*) or read out a test response bit through scan-output (*SO*).

Unlike scan chain design that the scan-output of a cell is the scan-input of its successor cell, which makes *SI* and *SO* share one wire in the middle of a scan chain, RAS needs two dedicated wires for *SI* and *SO*. Except *TE*, there are other two enable signals, *RE* and *CE*, connected to a RAS scan cell. Roughly speaking, RAS has three types of signals contributing to long wire length: *SI*/*SO*, *RE* and *CE*, which is the cause for prohibitively high routing overhead. Moreover, compared with multiplexer-type scan cell used in the conventional scan chain design, each RAS scan cell is augmented with a 12-transistor multiplexer and a 4-transistor NAND gate, which makes it 1.57 times larger than a 28-transistor multiplexer-type scan cell. High hardware overhead prevents RAS from practical utilization.

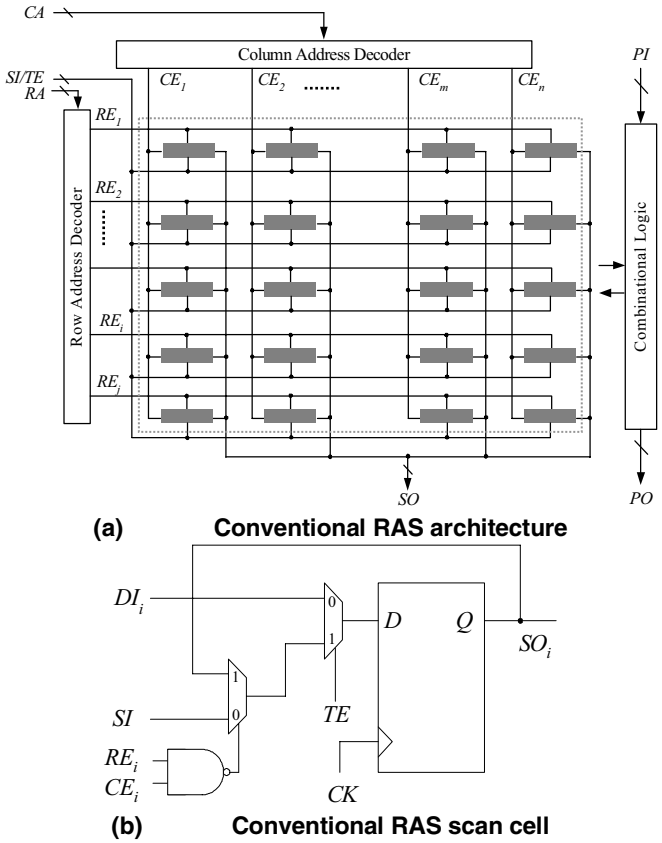


Fig. 1. Conceptual structure of conventional RAS

III. Localized Random Access Scan

A. LRAS Architecture

The LRAS architecture, just as the name implies, can effectively reduce routing overhead by localizing previously long *RE* and *CE* signals in small blocks or modules, and by eliminating the global *TE* signal. In addition, LRAS multiplexes *SI* and *SO* to the signal *SIO*, thus further reduces routing overhead.

Fig. 2 shows an example of LRAS architecture. The floorplan of a chip is partitioned to several blocks which might contain various numbers of scan cells. Variant blocks can have various numbers of scan cells, such flexibility is very important because in some cases, e.g. a System-on-Chip (SOC) with analog/mixed signal modules and RAMs distributed on the chip, the density of flip-flops in various geometric blocks may be very different. Each block has a row address decoder (*RAD*) and a column address decoder (*CAD*). To select a scan cell in a block, the block address code shall be input to the block decoder (*BD*) through block address (*BA*) bus, meanwhile, *RA* bus and *CA* bus transfers the row address code and the column address code of the scan cell in the block to *RAD* and *CAD*, respectively. Notice that all of *BA*, *RA*, *CA* and *SIO* pins can reuse function Input/Output pins, and *SIO* pins must be bi-directional.

As illustrated in Fig. 3, when the block-select signal (*BSEL*) is asserted, *RAD* and *CAD* decode the addresses on *RA* and *CA*, hence corresponding RE_u and CE_v (u>0 and v>0) are asserted, otherwise, *RAD* and *CAD* will assert the reserved RE_0 and CE_0 enable signals. *CE* and its inversion

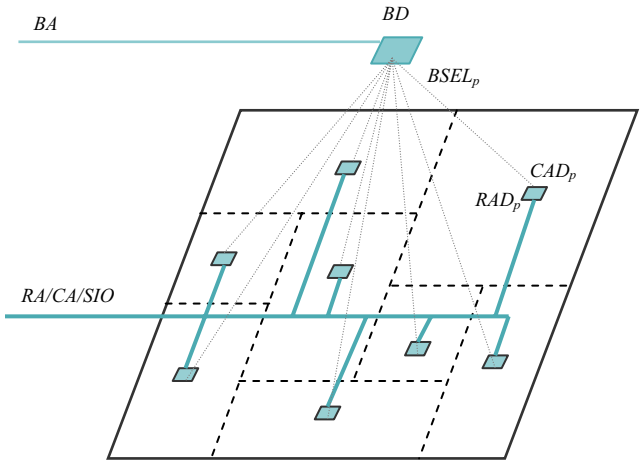


Fig. 2. LRAS architecture

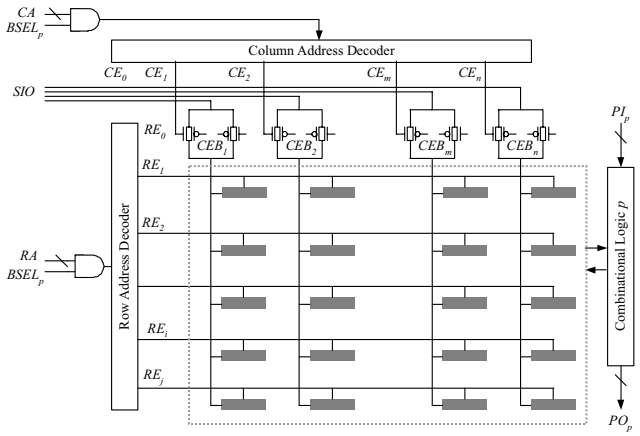


Fig. 3. Block structure

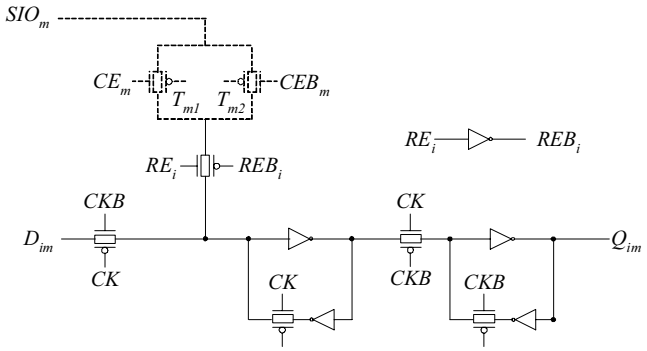


Fig. 4. LRAS scan cell structure

signal *CEB* control which of the transmission gate pair can connect the *SIO* signal to a column of scan cells.

Compared with conventional RAS, LRAS has *BA*, *BSEL* and more address decoders, however, the number of blocks is small, e.g. it is from 4 to 8 in our experiments conducted on large ISCAS'89 and ITC'99 benchmark circuits. LRAS is still benefited from adding a top address decoder *BD* to localize *RE* and *CE*.

B. LRAS Scan Flip-Flop

Fig. 4 shows the structure of LRAS scan cell. LRAS scan cell augments the conventional positive edge triggered D flip-flop (DFF) with a transmission gate and an inversion gate. Test stimuli bit and test response bit are transferred by the bi-directional signal SIO_m, which is routed to scan cells of column *m*. As only two test signals: RE_i and SIO_m are needed for a scan cell, so the routing overhead is reduced.

Assume a transmission gate consists of 2 transistors, an inverter 2 transistors, and a multiplexer 12 transistors, then the area overhead for each LRAS scan cell versus multiplexer-type scan cell is 20/28=71.4%. Except less transistors, LRAS scan flip-flop does not insert logic in the signal propagation path, thus it has less performance penalty comparison with multiplexer-type scan cell.

TABLE 1
Operation modes of the LRAS scan cell

RE	*CE*	*CK*	mode
0	X	↕	Function
1	1	1	Write
1	0	1	Read
0	X	1	Hold

LRAS scan cell has four operation modes as shown in Table 1. In Function mode, the clock signal *CK* transits at certain frequency. As RE_i=0, the flip-flop is disconnected with SIO_m, and D_{im} is propagated to Q_{im}. In Write mode, *CK* is kept high, as soon as RE_i and CE_m are asserted, test stimuli bit on SIO_m is transferred through transmission gate T_{m1} to the scan cell. In Read mode, since RE_i =1, the states of scan cells in row *i* are simultaneously read out to *SIO* bus through transmission gate T_{m2}. In Hold mode, as RE_i=0 and *CKB*=1, the input of the master stage of flip-flop is high impedance so the scan cell holds its previous state.

C. Test Application Time Analysis

In LRAS, the read operation is conducted half cycle before writing test stimulus bit to the first cell whose column address is smallest among the row of cells that will flip their states. Therefore, in the case of testing stuck-at faults, the number of cycles to test a LRAS-designed circuit is the sum of writing cycles and capture cycles:

$$T_{LRAS} = \sum_{p=1}^{P}\sum_{l=1}^{g}\left(s_l^p + 1\right)$$

where *P* denotes the number of test patterns, *g* is the number of blocks, s_l^p is the number of scan cells that flip their states to load the *p*th test stimulus into the *l*th block.

It is known that test application time of a chain-based scan design can be given by

$$T_{Chain} = P\left(1+\left\lceil\frac{s}{h}\right\rceil\right)+\left\lceil\frac{s}{h}\right\rceil$$

where *h* is the number of scan chains.

To assign approximate number of test pins for LRAS and scan chain designs, the relationship of *h* and *g* should be

$$\begin{aligned}2h+1 &= BA+RA+CA+SIO\\ &= \lceil\log_2 g\rceil + \max_{l\in(1,2,\cdots,g)}\left(\log_2\left(\left\lceil\sqrt{s_l}\right\rceil+1\right)\right)\\ &+ \max_{l\in(1,2,\cdots,g)}\left(\log_2\left(\left\lceil\frac{s_l}{\left\lceil\sqrt{s_l}\right\rceil}\right\rceil+1\right)\right)+\max_{l\in(1,2,\cdots,g)}\left(\log_2\left\lceil\frac{s_l}{\left\lceil\sqrt{s_l}\right\rceil}\right\rceil\right)\end{aligned}$$

where the left term represents the number of test pins which are scan-inputs, scan-outputs and *SE*, for the scan chain based design; The right term denotes the total width of address buses and *SIO* bus for the LRAS design. Notice the added "1" in the second and the third right terms is for reserving RE_0 and CE_0 when design *RA* and *CA*.

For an instance, assume *s* is from 500 to 5000, *g*=8, s_l^p =0.1**s*, Fig. 5 shows the test application time of LRAS and scan chain based designs.

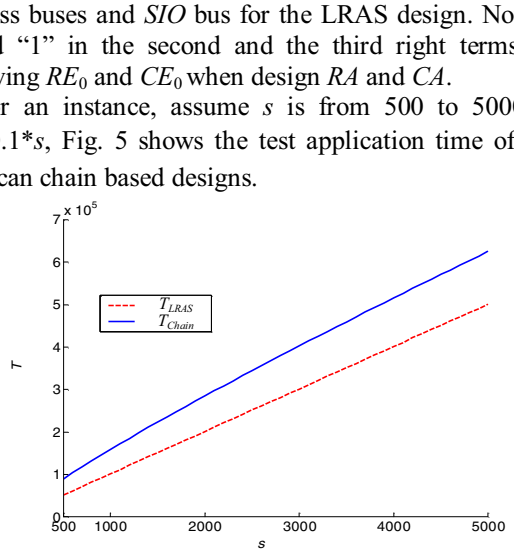


Fig. 5. Comparison of test application time

TABLE 2
Experimental results on hardware overhead

Circuit Name			s13207	s15850	s35932	s38417	s38584	b17	b20	b22
# Flip-flops			638	534	1728	1636	1426	1414	490	703
Our Work	LRAS	g	6	8	7	6	6	6	8	6
		DR (%)	47.7	44.5	49.5	48.8	47.5	47.4	37.3	49.1
		#Pins	23	21	33	32	31	32	20	26
	MSC	#Chains	11	10	16	15	15	15	9	12
		#Pins	23	21	33	31	31	31	19	25
	Area Overhead (um^2)	Plain	51379	52560	158483	145217	141179	262596	114963	162717
		LRAS	55367	57572	168367	153420	146357	270865	119433	167659
		MSC	60913	62689	181501	172191	176871	281423	121483	172071
	Routing Overhead (um)	Plain	107680	93697	401246	269303	397856	990035	363601	545274
		LRAS	166617	148592	580726	423805	547971	115000	458664	657837
		MSC	148824	133169	529897	390178	554190	114252	441389	647701
[8]	Area Overhead (%)	PRAS	9.5	7.9	8.1	7.2	6.5	4.5	5.4	4.6
		MSC	8.0	6.4	8.2	7.2	6.4	4.4	4.2	4.0
	Routing Overhead (%)	PRAS	25.7	20.2	25.1	23.0	19.0	8.7	10.7	10.4
		MSC	21.1	16.1	20.5	18.5	15.1	6.8	8.3	8.1

D. Testing LRAS

The DFT logic of LRAS shall be tested before testing the circuit-under-test. Since the access mechanism of LRAS is similar to that of SRAM, memory test techniques can be utilized to test LRAS. However, scan flip-flops of LRAS are distributed all over the circuit; the scan flip-flop density of LRAS is much lower than the memory cell density of SRAM. Therefore, memory faults such as state coupling faults rarely happen in LRAS. The test requirement of LRAS is relaxed. Simple memory built-in self test algorithm, e.g. MATS++ [11] can be used to test the LRAS DFT logic

IV. Heuristic Clustering Algorithm

In the two-pass LRAS DFT synthesis flow, a crucial step is to group flip-flops. Given the coordination of flip-flops, we employ an agglomerative hierarchical clustering algorithm (AHCA) [12] to group flip-flops. Firstly, a distance matrix (*DM*) is obtained by calculating Manhattan distance between flip-flops, e.g. $d_{ij}=|x_i-x_j|+|y_i-y_j|$, and then following steps are executed:

Step 1: Assign each flip-flop to a cluster. So that if there are s flip-flops, there are now s clusters, each containing just one flip-flop. Let the distances between the clusters the same as the distances between flip-flops.

Step 2: Find the closest pair of clusters c_a and c_b, and merge c_a and c_b into a single cluster c_{ab}, so that the number of clusters decreases by one.

Step 3: Update *DM*. Calculate the Manhattan distance of c_{ab} between other clusters, Remove the rows and columns representing of c_a and c_b from *DM*, while insert a row and a column to represent c_{ab}.

Step 4: Repeat steps 2 and 3 until all flip-flops are clustered into two clusters or Distance Ratio (*DR*) is beyond a user-given threshold.

We define *DR* as the ratio of average Manhattan distance between scan cells and their *CAD*/*RAD* address decoders to average Manhattan distance between scan cells and *BD* address decoder. *DR* indicates the extent of scan cells concentrated to the address decoder. Lower *DR* indicates higher concentration, thus shorter wire length of *RE*. The formula of calculating *DR* is given by

$$DR=\frac{1}{g}\sum_{l=1}^{g}\frac{\sum_{i=1}^{s_l}\left(|x_{li}-x_{l0}|+|y_{li}-y_{l0}|\right)}{s_l}\Bigg/\frac{\sum_{i=1}^{s}\left(|x_i-x_0|+|y_i-y_0|\right)}{s}$$

where g denotes the number of blocks, s_l is the number of scan cells in the ith block, s is the total number of scan cells in the circuit.

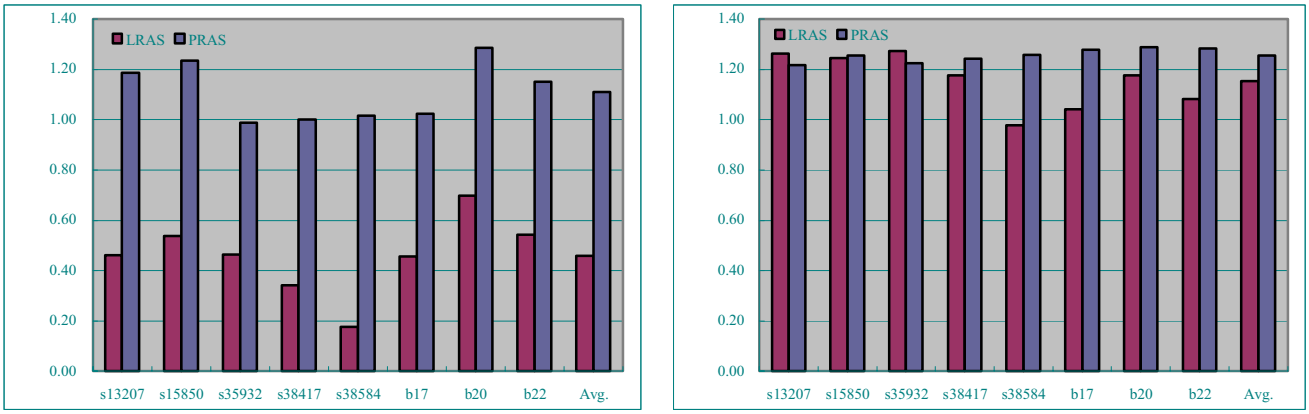
In Step 3, usually the distance between two clusters can be the maximum distance between flip-flops of each cluster, the minimum distance or the mean distance between flip-flops of each cluster. Here we use the mean distance to be the distance of two clusters.

After grouping, flip-flops within a group are further assigned to form rows and columns, and then address decoders are generated. Finally, we synthesize the modified RTL codes of the circuit for conducting placement and routing.

V. Experimental Results

We have conducted experiments on five large ISCAS'89 and three ITC'99 benchmark circuits. A 0.18um digital CMOS technology was used. For the multiple-scan-chain (MSC) versions of these circuits, scan chains were optimally reordered by Astro. Area overhead was estimated by Design Compiler, while routing overhead was estimated by Astro. Design Compiler and Astro are from Synopsys. The other algorithms, e.g. AHCA, were implemented in Matlab. All experiments were conducted on four 1.8GHz XEON processors running Linux.

Table 2 shows the experimental results of hardware overhead for eight circuits. The first and second rows list names of benchmark circuits and numbers of flip-flops in each circuit.



(a) Normalized Area Overhead Increase **(b) Normalized Routing Overhead Increase**

Fig. 6. Hardware overhead comparisons of LRAS against PRAS

In the entry of "Our Work", the "LRAS" rows contain circuit structure information designed with LRAS. The row of "g" shows the number of flip-flop groups in the circuit. The row of "DR (%)" gives the distance ratio after grouping scan flip-flops. The row of "#Pins" shows the number of pins used for testing the circuit. Below "LRAS", the "MSC" rows contain circuit structure information designed with multiple scan chains. "#Chains" is the number of scan chains in the circuit. "#Pins" is the total number of scan-input pins, scan-output pins, and the *SE* pin. "Area Overhead (um^2)" rows present the total area reported by Design Compiler. Here "Plain" means the original benchmark circuits without DFT. In rows of "Routing Overhead (um)", the total wire lengths reported by Astro are presented.

As PRAS in [8] is a typical state-of-the-art RAS scheme and it provides experimental results on area and routing overhead, so we compare LRAS with PRAS. According to [8], the area overhead of PRAS and MSC was the fractions of the number of transistors used in scan circuitry over total number of transistors in the circuit. Similarly, the routing overhead was the fraction of scan routing length over total routing length for the circuit. The rows of "Area Overhead (%)" and "Routing Overhead (%)" lists the experimental results shown in [8].

To compare the hardware overhead of LRAS and PRAS, we normalized the increased hardware overhead by

Normalized Hardware Overhead Increase

$$= \begin{cases} \dfrac{LRAS - Plain}{LRAS} \Big/ \dfrac{MSC - Plain}{MSC}, & LRAS \\ \dfrac{PRAS\ Overhead}{MSC\ Overhead}, & PRAS \end{cases}$$

For an instance of s13207, the normalized area overhead increase of LRAS and PRAS are given by

$$\frac{55367-51379}{55367} \Big/ \frac{60913-51379}{60913} = 0.46 \quad \text{and} \quad \frac{9.5}{8.0} = 1.19$$

here 0.46 means LRAS has 54% less area overhead than multiplexer-type based MSC, while 1.19 indicates PRAS has a 19% increase in area than MSC.

Fig. 6 illustrates normalized hardware overhead increase of LRAS and PRAS. From sub-figure (a), we can see for all experimental circuits, LRAS averagely has a 54% reduction on area overhead than MSC, while PRAS has an 11% increase than MSC. From sub-figure (b), we can see LRAS outperforms PRAS for the larger circuits from s38417 to b22, although in the case of smaller circuits s13207 and s35932, routing overhead increase of LRAS is 5% more than that of PRAS. In general, routing overhead increase of LRAS is 11% less than that of PRAS.

VI. Conclusions

In this paper we presented a scan architecture called Localized Random Access Scan (LRAS) that can effectively reduce hardware overhead. The proposed LRAS architecture utilizes the layout information by grouping scan cells according to their coordination and then generating two layers of address decoders. An agglomerative hierarchical clustering algorithm was employed to optimally cluster flip-flops. Experimental results on ISCAS'89 and ITC'99 benchmark circuits confirmed that LRAS was capable of producing lower routing overhead circuits than the state-of-the-art RAS scheme, and it had 54% less area overhead than multiplexer-type scan chain based designs.

As LRAS has no impact on ATPG, the fault coverage is the same as scan chain DFT. In the future work, we will optimize the clustering algorithm to further reduce the routing overhead. Meanwhile, since LRAS only modifies the flip-flop access fashion from flat to hierarchy, it shall have merits of other RAS architectures, such as short test application time and low test power dissipation. We will demonstrate such merits in our future work.

Acknowledgments

This work was supported in part by 21st Century COE (Center of Excellence) Program "Ubiquitous Networked Media Computing", in part by JSPS (Japan Society for the Promotion of Science) under Grants-in-Aid for Scientific Research B (No. 15300018), in part by National Natural Science Foundation of China (NSFC) under grant No. 60633060, 90607010, 60576031, and in part by National Basic Research Program of China (973) under grant No. 2005CB321604 and 2005CB321605.

References

[1] H. Ando, “Testing VLSI with Random Access Scan”, *Digest of Computer Society International Conference (COMPCON)*, 1980, pp. 50-52.

[2] K. Wagner, “Design for Testability in the Amdahl 580”, *Digest of Computer Society International Conference (COMPCON)*, 1983, pp. 384-288.

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

[4] D. Baik, S. Kajihara, and K.K. Saluja, “Random access scan: A solution to test power, test data volume and test time,” *Proceedings of VLSI Design*, 2004, pp. 883-888.

[5] K.T. Le, D.H. Baik, and K.K. Saluja, “Test Time Reduction to Test for Path-Delay Faults using Enhanced Random-Access Scan”, *Proceedings of VLSI Design*, 2007, pp. 769-774.

[6] Y. Hu, Y.-H. Han, X.-W. Li, H.-W. Li, and X.-Q. Wen, "Compression/Scan Co-Design for Reducing Test Data Volume, Scan-in Power Dissipation and Test Application Time", *IEICE Transactions on Information and Systems*, 2006, Vol. E89-D, No. 10, pp. 2616-2625.

[7] S.-P. Lin, C.-C. Lee and J.-E. Chen, “A Cocktail Approach on Random Access Scan toward Low Power and High Efficiency Test,” *Proceedings of International Conference on Computer-Aided-Design (ICCAD)*, 2005, pp. 94-99.

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

[9] A.S. Mudlapur, V.D. Agrawal, and A.D. Singh, “A Random Scan Architecture to Reduce Hardware Overhead”, *Proceedings of International Test Conference (ITC)*, 2005, no. 15.1, pp. 1-9.

[10] S. Makar, “A Layout-Based Approach for Ordering Scan Chain Flip-Flops”, *Proceedings of International Test Conference (ITC)*, 1998, pp. 341-347.

[11] A.J. van de Goor, “Testing Semiconductor Memories: Theory and Practice”, Chichester, UK: John Wiley & Sons, Inc., 1991.

[12] S.C. Johnson, “Hierarchical Clustering Schemes”, *Psychometrika*, 1967, Vol. 32, No. 2, pp. 241-254.