

System-On-a-Chip Test Data Compression and Decompression with Reconfigurable Serial Multiplier

S.Sivanantham^{*}, Padmavathy M[#], Divyanga S[#], Anitha Lincy P V[#]

ASIC Design Laboratory, School of Electronics Engineering
VIT University

Vellore – 632014, Tamilnadu, India

^{*}ssivanantham@vit.ac.in

[#]{padma2407, divyngasiva, anitha.lincy}@gmail.com

Abstract — One of the increasingly difficult challenges in testing System-On-a-Chip (SoC) is dealing with the large amount of test vectors that must be stored in the tester and transferred between the testers. The test data bandwidth between the tester and the SOC is a bottleneck that can result in long test times when testing complex SOCs that contain many cores. Hence a test data compression and decompression scheme using reconfigurable multipliers has been presented in this paper. This scheme stores test vectors as a product of two deterministic vector seeds and uses the intermediate states of the multiplication of the vector seeds using the reconfigurable serial multiplier, to realize the test vector. Since multipliers are one of the widely used components in many SOCs, this method reduces significant hardware by exploiting the advantage of using the existing circuitry in the circuit under test for decompression. It provides a twofold advantage by reducing the amount of test data that needs to be stored on the tester and reducing the time for transferring test data from the tester to the circuit-under-test resulting in better encoding efficiency. Linear decompression with free variables is used. The encoding efficiency could be further explored with high percent of free variables. This scheme works best for circuits within built multipliers and significant reduction in hardware is observed. Experimental results obtained with ISCAS'89 benchmark shows the efficiency of this scheme on reduction of test data volume as well as the test time.

Keywords— *VLSI Testing, reconfigurable multiplier, system-on-a-chip, linear feedback shift register, design for testability, linear decompression.*

I. INTRODUCTION

Test data volume that must be transferred between the tester and the chip is a major problem encountered in the testing of system-on-a-chip (SOC) designs. A typical SOC consists of several intellectual property (IP) blocks, each of which must be exercised by a large number of pre-computed test patterns. The increasingly high volume of SOC test data is not only exceeding the memory and I/O channel capacity of commercial automatic test equipment (ATE) but it is also leading to excessively high testing times. Test data compression techniques provide means to reduce the test data storage requirements on the tester and the test data bandwidth requirements on the tester and the test data bandwidth requirements between the tester and chip, thereby allowing less expensive testers to be used as well as reducing time. The general setup used to test the system-on-chip using automatic test equipment is shown in Fig. 1.

A number of test data compression schemes such as statistical coding, run-length coding, Huffman coding, LZ77 [2], frequency directed codes [6] have been developed using a variety of techniques to reduce the storage on the tester and also to reduce the test data bandwidth between the tester and the chip. Also many techniques such as [7-8], [14] were proposed to reduce the test power in scan based testing. The adaptive test clock with low transition [15] is used to reduce the test application time. The compression utilizes the fact that the majority of bits in most test cubes are unspecified, which increases the probability of mutual compatibility and provides a high degree of freedom while solving the linear equations and specifies the compression ratio. A very popular test vector compression technique is based on linear expansion [1] where test data is decompressed on chip using only linear operations. This includes linear feedback shift registers (LFSR) reseeding, linear expansion networks consisting of XOR gates and seed overlapping [7]. Commercial tools for compressing test vector TestKompress from Mentor graphics and Smart BIST [4] from IBM/Cadence are based on linear expansion circuits. Every decompressed bit is represented as a linear combination (modulo 2) of the stored compressed data. The stored bits which are free variables can be assigned any value (0, 1). The values are assigned such that when shifted into the scan chains through linear expansion networks, they can reproduce the desired test cubes. The choice of decompression network is vital as the encoding efficiency, which is defined as the ratio of specified bits to the number of bits stored on the tester (i.e. the compressed data) largely depends on it.

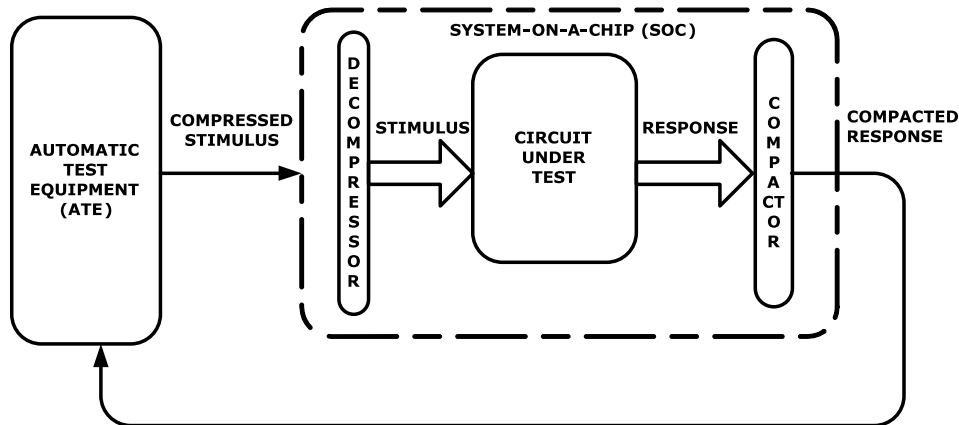


Fig. 1. Set-up for testing System-on-a-Chip (SoC) using Automatic test equipment

The number of free variables can be fixed [9] or variable [4], [11] which is determined according to the complexity. In order to guarantee a reduction in the overall test time, the compression and decompression process should not add additional delay and area to the circuit.

The test data compression allows some additional hardware before and after the scan chain. The extra hardware decompresses the test set which comes from the tester. It also stores the data in a compressed format on the tester [5]. To achieve maximum compression the don't care bits must be filled with '1's or '0's. But the way of filling the don't care bits depends upon the nature of code. Also the don't care bits are filled to reduce the test power [8, 12]. However in linear compression technique the don't care bits need not be filled with either '0's or '1's instead the test vector is formed to become a test cube and from the specified bits in the test cube seeds values are formed and stored on the tester hence achieving greater compression.

II. LINEAR DECOMPRESSORS

One very economical option is to use the existing components in the circuit under test (CUT) to test other parts of the circuit. Since multiplication is very commonly used and is an expensive operation [10] rapid improvements of VLSI technologies have been used to increase the reliability and the speed of the multiplication units embedded in the circuit. This is particularly important for specialized chips supporting multiplication intensive operations, such as in digital signal processing and computer graphics. The reconfigurable multiplier architecture proposed in [13] is generally used to reduce the power dissipation. Also multiplier with computation sharing were introduced in [3] for signal processing application where it demands for floating-point operations. But our approach to reduce the test data demands reconfigurable serial multiplier. Bit serial multipliers are area efficient in contrast to N^2 adder requirement for bit parallel multipliers. The serial multipliers can be easily configured to function as a cyclic code generator, perform multiplication in GF (2), and function as an LFSR in addition to functioning as a pure binary multiplier. For circuits with a built-in multiplier unit, test data decompression can be performed using the multiplier without having to modify the scan chains. Linear decompressor consisting of only XOR gates and flip-flops, expand the data coming from the tester to fill the scan chains. A test vector Z can be compressed by a particular linear decompressor if and only if there exists a solution to a system of linear equations, $AX=Z$, where A is the characteristic matrix of the linear decompressor and X is a set of free variables stored on the tester (which can be assigned the value '0' or '1'). A figure of merit for linear decompressor is "encoding efficiency", which is defined as

$$\text{Encoding efficiency} = \frac{\text{Specified bits in Test set}}{\text{Bits stored on Tester}} \quad (1)$$

III. DECOMPRESSION USING RECONFIGURABLE SERIAL MULTIPLIER

The decompression of deterministic test vectors on chip is done using a reconfigurable serial multiplier. Fig. 2 shows the architecture of a reconfigurable serial multiplier. The deterministic vectors are stored as compressed "seeds" and are decompressed by multiplying the seeds together in the reconfigurable serial multiplier for partial products that are processed with simple arithmetic operations and concatenated to obtain the actual test vector. For an $(n*n)$ test matrix, the objective is to find two n -bit numbers A and B which will be used as seeds, such that the states of the multiplier while multiplication can be used in the shift-and-add algorithm to generate the test matrix. A set of linear equations are formed for the given test vector and solved using Gauss Jordan elimination method to obtain the test vector.

For a linear decompressor, finding an assignment for the free variables that will encode a particular test cube can be done by solving the system of linear equations for the specified bits in the test cube. Consider the example where the test vector is converted into a (4*4) test matrix. The vectors *A* and *B* are obtained which when multiplied using the reconfigurable multiplier implementing the shift and add algorithm provides the desired test matrix.

$$\begin{bmatrix} t14 & t13 & t12 & t11 \\ t24 & t23 & t22 & t21 \\ t34 & t33 & t32 & t31 \\ t44 & t43 & t42 & t41 \end{bmatrix} = \begin{bmatrix} 1 & x & x & 1 \\ 1 & 0 & 0 & x \\ 1 & 0 & 1 & 1 \\ x & 0 & x & x \end{bmatrix} \tag{2}$$

$$A = [a_4 \ a_3 \ a_2 \ a_1] \tag{3}$$

$$B = [b_4 \ b_3 \ b_2 \ b_1] \tag{4}$$

$$v_{ij} = a_i \cdot b_j \tag{5}$$

The partial products of the multiplier generated while computing *A*B* is matrix in Eq.5. The states of the multiplier vary during each clock cycles depending on the shift and add operation. Depending on these operations the seed values (*A*&*B*) are calculated using Eq. 6. The shifted matrix is given by the Eq.5.

$$\begin{bmatrix} & & & v14 & v13 & v12 & v11 \\ & & & v24 & v23 & v22 & v21 \\ & & v34 & v33 & v32 & v31 & \\ v44 & v43 & v42 & v41 & & & \end{bmatrix} \tag{6}$$

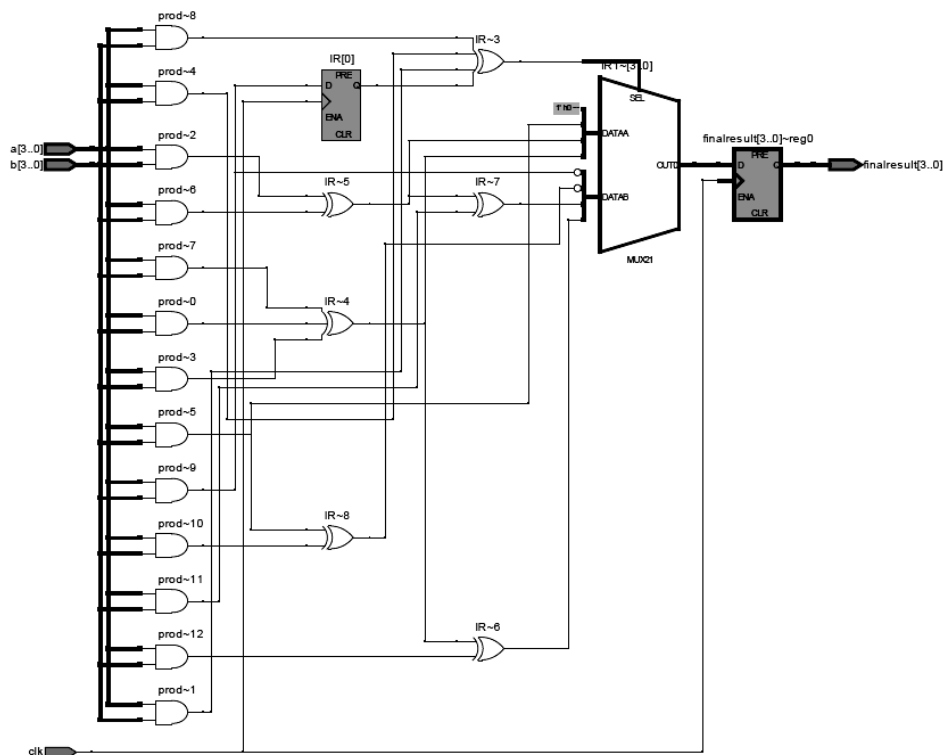


Fig. 2. Reconfigurable Serial Multiplier Architecture

$$\begin{bmatrix} t14 & t13 & t12 & t11 \\ t24 & t23 & t22 & t21 \\ t34 & t33 & t32 & t31 \\ t44 & t43 & t42 & t41 \end{bmatrix} = \begin{bmatrix} v14 & v13 & v12 & v11 \\ v24 & v14^{\wedge}v23 & v13^{\wedge}v22 & v12^{\wedge}v21 \\ v34 & v24^{\wedge}v33 & v14^{\wedge}v23^{\wedge}v32 & v13^{\wedge}v22^{\wedge}v31 \\ v44 & v34^{\wedge}v43 & v24^{\wedge}v33^{\wedge}v42 & v14^{\wedge}v23^{\wedge}v32^{\wedge}v44 \end{bmatrix} \tag{7}$$

The states of the multiplier vary during each clock cycles depending on the shift and add operation. Depending on these operations the seed values (*A* & *B*) are calculated. From (6) the objective is to find *A* & *B* by filling '0' and '1' which should be equal to Eq.1. The Eq. 7 can be written as $Mx=y$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} v11 \\ v12 \\ v13 \\ v14 \\ v21 \\ v22 \\ v23 \\ v24 \\ v31 \\ v32 \\ v33 \\ v34 \\ v41 \\ v42 \\ v43 \\ v44 \end{bmatrix} = \begin{bmatrix} t11 \\ t12 \\ t13 \\ t14 \\ t21 \\ t22 \\ t23 \\ t24 \\ t31 \\ t32 \\ t33 \\ t34 \\ t41 \\ t42 \\ t43 \\ t44 \end{bmatrix} \tag{8}$$

Only the equations where the bits are specified has to be solved hence the modified matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} v11 \\ v12 \\ v13 \\ v14 \\ v21 \\ v22 \\ v23 \\ v24 \\ v31 \\ v32 \\ v33 \\ v34 \\ v41 \\ v42 \\ v43 \\ v44 \end{bmatrix} = \begin{bmatrix} t11 \\ t14 \\ t22 \\ t23 \\ t24 \\ t31 \\ t32 \\ t34 \\ t34 \\ t43 \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} t11 \\ t14 \\ t22 \\ t23 \\ t24 \\ t31 \\ t32 \\ t34 \\ t34 \\ t43 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

The above equations are solved according to Eq.4. The Gauss-Jordan Elimination method is applied in serial multiplier on Eq. 8 to obtain the vector seeds A and B .

$$t_{14} = 1 \Rightarrow v_{14} = 1, a_1 = 1, b_4 = 1$$

$$t_{11} = 1 \Rightarrow v_{11} = 1, b_1 = 1$$

$$t_{21} = 1 \Rightarrow v_{21} = 1, a_2 = 1$$

$$t_{31} = 1 \Rightarrow v_{31} = 1, a_3 = 1$$

$$t_{23} = 0 \Rightarrow v_{23} = 1, b_3 = 1$$

$$t_{22} = 0 \Rightarrow v_{22} = 1, b_2 = 1$$

$$t_{43} = 0 \Rightarrow v_{43} = 1, a_4 = 1$$

For the above example the seed values are obtained as $A=1111$ and $B=1111$. The two numbers A and B which are called the seeds represent the compressed version of the test vector matrix.

A. Computation of Efficiency and Operand Sharing

A figure of merit for linear decompressor is encoding efficiency, which is defined as “specified bits in test set/Bits stored on tester”. Also the prospects of operand sharing can be taken into account for further encoding efficiency. As we see many test vectors possess unspecified bits and these bits can be taken in any combination to form test vectors that have the same operands i.e. greater the number of unspecified bits greater is the sharing. Also, there are distinct test cubes that share some common operands and hence contribute for much greater encoding efficiency as compared to the former. Here in the above example, the compression for a given test vector is from n^2 bits to $2n$ bits therefore resulting a compression efficiency of 50%. Also the encoding efficiency is obtained as 1.25.

IV. EXPERIMENTAL RESULTS

The compression scheme was developed for circuits with built-in serial multipliers by developing the pivotal matrix solving the same using Gauss Jordan elimination method. The test patterns are generated using ATALANTA ATPG tool for ISCAS’89 full scan sequential circuits. The fault simulation was done using FSIM tool. The decompression architecture was modeled using Verilog HDL and has been used to compress the test vectors for various benchmark circuits. The functional verification was performed with Cadence ncsim and the entire architecture was synthesized using Cadence Encounter RTL Compiler with TSMC 180nm, 1.8V CMOS technology.

Table I show the encoding efficiency of two ISCAS’89 benchmark circuits. The column I indicate the name of the circuit. The column 2-4 in Table I gives the number of Primary inputs, number of equivalent gates and number of primary outputs for the circuits. The column 5 indicates the number of pattern with don’t care bits generated using ATALANTA ATPG tool and corresponding fault coverage is given in column 6. The last column provides the encoding efficiency of the scheme which is computed using Eq. 1. We have achieved an encoding efficiency of 24.74% and 54.4% for s1196 and s1238 circuits respectively. It is evident from the Table I that if the number of bits specified in a test cube is small and when a larger multiplier unit is used then higher encoding efficiency will be achieved. On the other hand if a densely specified test cubes are used then smaller multiplier units provide better encoding efficiency. The decompression scheme for the same is generated using the reconfigurable serial multiplier that decompresses vector seeds to obtain the original test vector. The operand sharing possibility of the proposed scheme in the bench mark circuits have been observed based on the compression results of the circuits. It is seen that the same operands are obtained as the compressed one of seeds for different test cubes.

TABLE I. EXPERIMENTAL RESULTS ON ENCODING EFFICIENCY FOR ISCAS’89 BENCHMARK CIRCUITS

Circuit Name	# PIs	# gates	# POs	#Test Patterns	Fault coverage	Encoding efficiency
S1196	32	529	32	1242	100%	24.74%
S1238	32	508	32	1286	94.91%	54.40%

TABLE II. COMPARISON OF AREA AND POWER REPORTS WITH VARIOUS COMPRESSION/DECOMPRESSION SCHEME (SYNTHESIZED WITH TSMC180NM, 1.8V CMOS STANDARD CELL LIBRARY)

Method	Power			Area (um ²)
	Leakage Power (nW)	Dynamic Power (nW)	Total power (nW)	
Compression using Gauss Jordan Elimination	8.642	282889.18	282897.82	4078
Decompression using GF Multiplier	13.122	76509.82	76522.94	722
Decompression using Serial Multiplier	1.466	31707.61	31709.08	393

The compression and decompression designs have been optimized for power and synthesized. The decompression in the proposed scheme using serial multiplier has been compared for area and power with the method using GF(2) field multiplier and the results are presented in Table II. The results clearly show that the power and the area are reduced to a greater extent when a serial multiplier is used for decompression process. It can be seen that about 58.55% of power reduction and about 45.5% of reduction in area has been achieved with the proposed scheme.

V. CONCLUSION

The tests data compression/decompression scheme using a reconfigurable serial multiplier is presented in this paper. It provides a twofold advantage by reducing the amount of test data that needs to be stored on the tester and reducing the time for transferring test data from the tester to the circuit-under-test resulting in better encoding efficiency. Linear decompression with free variables is used. The encoding efficiency could be further explored with high percent of free variables. This scheme works best for circuits within built multipliers and significant reduction in hardware is observed.

REFERENCES

- [1] Dutta, A.; Rodrigues, T.; Touba, N.A.; , "Low cost test vector compression/decompression scheme for circuits with a reconfigurable serial multiplier," *VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on* , vol., no., pp. 200- 205, 11-12 May 2005
- [2] Wolff, F.G.; Papachristou, C.; , "Multiscan-based test compression and hardware decompression using LZ77," *Test Conference, 2002. Proceedings. International* , vol., no., pp. 331- 339, 2002
- [3] Sivanantham,S., Jagannadha Naidu, K., Balamurugan, S. & Bhuvana Phaneendra, D . 2013 "Low Power Floating Point ComputationSharing Multiplier for Signal Processing Applications", *International Journal of Engineering and Technology*, vol. 5, no. 2, pp. 979-985.
- [4] Krishna, C.V.; Touba, N.A.; , "3-stage variable length continuous-flow scan vector decompression scheme," *VLSI Test Symposium, 2004. Proceedings. 22nd IEEE* , vol., no., pp. 79- 86, 25-29 April 2004
- [5] Sivanantham, S.; Manuel, J.P.; Sarathkumar, K.; Mallick, P.S.; Perinbam, J.R.P.; , "Reduction of Test Power and Test Data Volume by Power Aware Compression Scheme," *Advances in Computing and Communications (ICACC), 2012 International Conference on* , vol., no., pp.158-161, 9-11 Aug. 2012
- [6] Chandra, A.; Chakrabarty, K.; , "Frequency-directed run-length (FDR) codes with application to system-on-a-chip test data compression," *VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001* , vol., no., pp.42-47, 2001
- [7] Rao, W.; Bayraktaroglu, I.; Orailoglu, A.; , "Test application time and volume compression through seed overlapping," *Design Automation Conference, 2003. Proceedings* , vol., no., pp. 732- 737, 2-6 June 2003
- [8] Sivanantham, V.Sandeep, P.S.Mallick, J.Raja Paul Perinbam, "A novel approach for simultaneous reduction of shift and capture power for scan based testing", *In Proc. International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN)*, pp. 418-423, 2011.
- [9] Rajski, J.; Tyszer, J.; Kassab, M.; Mukherjee, N.; Thompson, R.; Kun-Han Tsai; Hertwig, A.; Tamarapalli, N.; Mrugalski, G.; Eide, G.; Jun Qian; , "Embedded deterministic test for low cost manufacturing test," *Test Conference, 2002. Proceedings. International* , vol., no., pp. 301- 310, 2002.
- [10] Stelling, P.F.; Martel, C.U.; Oklobdzija, V.G.; Ravi, R.; , "Optimal circuits for parallel multipliers," *Computers, IEEE Transactions on*, vol.47, no.3, pp.273-285, Mar 1998.
- [11] Koenemann, B.; Barnhart, C.; Keller, B.; Sneath, T.; Farnsworth, O.; Wheeler, D.; , "A SmartBIST variant with guaranteed encoding," *Test Symposium, 2001. Proceedings. 10th Asian* , vol., no., pp.325-330, 2001
- [12] Sivanantham, S., Sarathkumar, K., Manuel, J.P., Mallick, P.S. & Perinbam, J.R.P., " CSP-Filling: A New X-filling Technique to Reduce Capture and Shift Power in Test Applications", *2012 International Symposium on Electronic System Design, ISED-2012*, pp. 135-139.
- [13] Praveen Kumar M V, Sivanantham S, Balamurugan S & Mallick P.S, "Low power reconfigurable multiplier with reordering of partial products," *Proc International conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN)*, 2011, vol., no., pp.532-536, 21-22 July 2011.
- [14] D. Satya Valibaba, S. Sivanantham, P. S. Mallick, and J. R. P. Perinbam, "Reduction of testing power with pulsed scan flipflop for scan based testing," in *2011 - International Conference on Signal Processing, Communication, Computing and Networking Technologies, ICSCCN-2011*, 2011, pp. 526-531.
- [15] Sivanantham, S., Gopakumar, G., Pandey, A. & Paikada, M.J. 2013, "Adaptive test clock scheme for low transition LFSR and external scan based testing", *2013 International Conference on Computer Communication and Informatics, ICCCI 2013*.