Using MUXs Network to Hide Bunches of Scan Chains

Yinhe Han[1,2] Yu Hu[1] Huawei Li[1,2] Xiaowei Li[1,2]
1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080
2. Graduate School of Chinese Academy of Sciences, Beijing, 100039
E-mail : yinhes@ict.ac.cn

Abstract

This paper presents a decompression architecture using a periodically alterable MUX network. Compared to static XORs network, the periodically alterable MUXs network has multiple configurations to decode the input information flexibly. Probability analysis can help us to select the proper parameter when considering the DFT schemes. With the dedicated efforts, the smaller test data volume and test application time can be achieved compared to previous techniques.

1. Introduction

The rapid development of IC process and wide reuse of pre-designed intellectual property (IP) cores is possible to put a large number of transistors on a single chip, which poses some serious test challenges at the same time. These challenges include test application time, test data volume, testing power, and expensive tester, etc.[1]. Among them, test time and test data volume are dominant in high-volume testing of ICs. For a full-scan circuit, both test data volume and test application time are proportional to the number of test patterns (P) and the length of longest scan chain(L). Thus, one can try to reduce P and L or both to reduce test data volume and test application time.

L-Reduction can be obtained through some new scan architecture. Using one scan-in pin to drive multiple scan chains can shorten the scan chain. Lee [2] proposes the methodology to use one scan-in signal to feed multiple circuits. The Illinois scan architectures in [3][4][26] show a broadcast architecture in which one scan-in signal is fed to multiple scan chains. [5] proposes a type of reconfigurable shared scan-in architecture. To achieve *P*-Reduction, some test compaction algorithms are presented. A test compaction algorithm in [17] is presented to identify the redundant patterns and then remove them. In [11][20][23], the unities of test compaction and compression are presented.

Test data compression is a conventional way to reduce the product of P and L. Golomb[6], VIHC[7], dictionary-based[21][27] and nine coded[22] codes have been presented to compress test data. [8] describes a method based on statistical encoding by converting some specified input values to unspecified logic values first. Reda [9] presents a mutation decoder to compress test patterns into bit stream that indicates which bits need to be flipped in current test slice to obtain the subsequent one.

"Scan Chains Hiding" or "Scan Chains Concealment" is another *L*-Reduction scheme. LFSRs are applied in [10] to hide scan chains. A large LFSR driven by few external pins is presented. It is composed of a large number of smaller LFSRs, each of which feeds a scan chain. The mutation version of LFSR: ring generator is presented in [12]. Few inputs information is expanded by the ring generator. Three distinct ATPG schemes are used to achieve a higher degree of compression ratio. Some other LFSRs-based techniques are presented in [24][25]. The XORs network is another class of "hiding" technique. They are shown in [9][11][14][20][23]. These methods use a network consisted by XOR gates to expand the input information. It is a very efficient method if combined with the custom test compaction algorithm

2. Review of Related Work

The MUXs network is also a good alternative "hiding" technique. Some literatures have published the work on it. In general, the structure of decompressor can be reconfigured in these scheme. They can be divided into some catalogs based on the reconfiguration styles:

(1) Configuration-Per-Pattern: [15] presents a low overhead reconfigurable shared scan-in architecture. In this architecture, every pattern has a configuration which indicates the connection relations between the external inputs and scan chains. [29] provides a Reconfigurable Interconnection Network(RIN) in BIST application which is also can be applied to full-scan environment. In the RIN, a pattern count and a configuration count are used to control and determine which configuration is selected.
(2) Configuration-Per-Cycle: [13] extends the work in [15] and presents additional ability to change values on the scan configuration signals(scan enable signals) during the scan operation. It gives the ability to select the different configurations per-shift.

In this paper, we will present a new scan architecture based on a periodically alterable MUXs Network. Different

* This paper is supported by the National Natural Science Foundation of China (NSFC) under grant No.90207002 and 60242001, key technique project of Beijing (No. H020120120130) and Basic Research Foundation of Institute of Computing Technology, Chinese Academy of Sciences(No. 20036160).

with the above two classes of MUXs networks, the proposed network can be reconfigured at a fix-length interval. This fix-length interval is the "period". Compared to previous XORs network and LFSRs, the following merits may be expected of the proposed MUXs network:

- Compared to XORs network, the connections between external pins and scan chains can be altered. This means that the solver would easily find a solution, which is specially important to the hard core when the test patterns provided by core vendor can't be tailored.
- Compared to LFSRs, the proposed MUXs network has low area overhead and provides a more flexible decoding scheme .

The rest of this paper is organized as: the proposed decompression architecture is introduced in Section 3. Section 4 introduces the probability analysis of decoding of MUXs network. The experimental results are analyzed in the Section 5.

3. Proposed MUXs Network Architecture

The proposed scan architecture with the MUXs network is shown in Figure 1. There are *M* external input pins which feed *N* scan chains through the MUXs network where *N* is far lager than *M*. Thus, the multi scan chains will be driven by one external input pin in the new scan architecture, which is similar to Illinois scan architecture [3]. However, the proposed hardware is superior to the latter since the connection relations between external pins and scan chains can be altered periodically. The periodic reconfiguration is implemented by changing the control signals of MUX gates periodically. These control signals are determined by the content of MUXS control registers, which is updated every *T* cycles. The update is triggered by the T-Counter. The data of MUXS control registers is from T-shifter when updating. In Figure 1, the T-bit shifter and MUXs control registers contain three registers (where *T* is equal to 3). The T-bit shifter is reseeded by a separate external input pin. After three cycles, the T-bit shifter should have a new content, and then the T-Counter sends a clock signal to MUXs control registers. The content in MUXs control registers would be uploaded from the T-bit shifter.

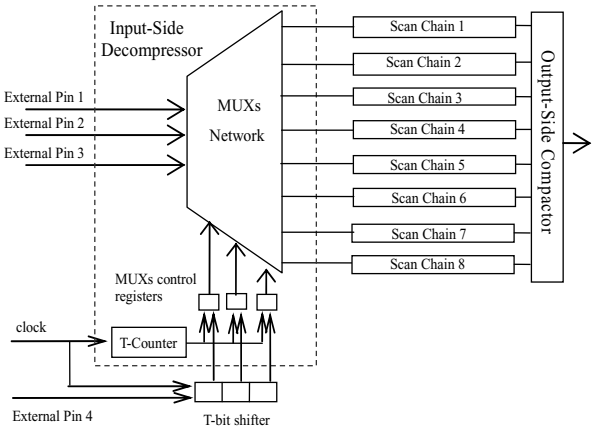

Figure 1. The Proposed Decompression Hardware

It is obvious that the reconfiguration period is equal to the length of T-bit shifter, which is also equal to the number of selecting lines in MUXs network. Thus, if the number of configurations of MUXs network is *C*, then the reconfiguration period of MUXs network *T* should be constrained as :

$$T \geq \lceil Log_2(C) \rceil$$

Like the XORs network, the MUXs network can be represented as a matrix in the formulation calculation. If the input data and output data of MUXs network are *X* and *Y*, and the MUXs matrix is *M*, then the relation between *X* and *Y* is:

$$[M] \times [X] = [Y]$$

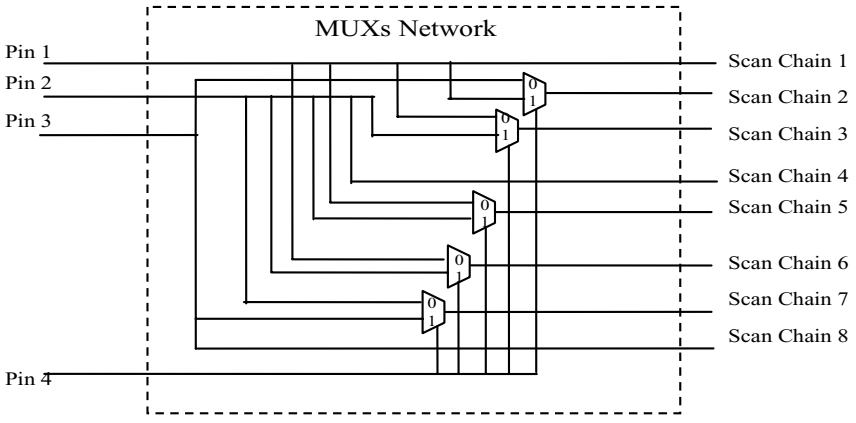

Figure 2. The internal structure of MUXs Network

Because the MUXs network can be altered, M may be variable in different periods. There is a set of MUXs matrixes: $\{M_1, M_2, \ldots, M_C\}$. In the i^{th} period, if M_i is selected, the decoding procedure will be further calculated as:

$$[M_i] \times [X_{(i-1)*T+1}, \ldots, X_{i*T}] = [Y_{(i-1)*T+1}, \ldots, Y_{i*T}]$$

Equation 1. The decoding procedure of MUXs Network.

where $M_i = [r_1 \quad r_2 \quad \ldots \quad r_N]^{TR}$, and the superscript TR means the transpose. Compared to the matrix of XORs network, each row vector r_i in M_i only contains single "1" since one scan chain is only connected to one targeted external input pin. An example is shown as:

[Example] In a circuit, 3 external input pins drive 8 scan chains. The MUXs network has two configurations. One of them is :{1-(1), 2-(3), 3-(2), 4-(2), 5-(1), 6-(1), 7-(2), 8-(3) }, where A-(B) means the A^{th} scan chain is driven by B^{th} external input pin. Under these connection relations, the MUXs matrix is:

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Another configuration is: {1-(1), 2-(1), 3-(1), 4-(2), 5-(2), 6-(2), 7-(3), 8-(3) }. Under these connection relations, the MUXs matrix is:

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

The MUXs network can be constructed by some MUXs gates. The implement of MUXs network of the above example is shown in Figure 2. It consists 5 two inputs MUXs gates. Because there are two configurations, only one control line of MUXs network is required. The Pin4 is used to change the configuration. If the control line is "0", the first configuration (M_1) is selected. Otherwise, the second configuration (M_2) is selected.

4. The Probability Analysis of MUXs Network

Let the test pattern set and the compressed pattern set for a core be Y and X, respectively. If the length of the longest scan chain is L and the number of test patterns are P, then there are a total of $L*P$ scan slices. The test pattern set can be partitioned into $B=\lceil L*P/T \rceil$ blocks: $Y = \{Y_1, Y_2, \ldots, Y_B\}$, where T is the reconfiguration period. The compressed pattern set can be partitioned into B blocks: $X = \{X_1, X_2, \ldots, X_B\}$, respectively. For N scan chains, the block Y_i can be represented as:

$Y_i = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{bmatrix}$ and the block X_i as $X_i = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_M \end{bmatrix}$. Each y_i and x_i is a T-bit vector. The ($M \times N$) MUXs decompressor is used to drive the N internal scan chains through M external inputs. In the i^{th} period, the matrix M_i of MUXs decompressor is selected:

$$M_i = \begin{bmatrix} r_1 \\ r_2 \\ \ldots \\ r_N \end{bmatrix},$$

then the decompression equation can be represented as:

$$\begin{bmatrix} r_1 \\ r_2 \\ \ldots \\ r_N \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_M \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{bmatrix}$$

Equation 2. Matrix representation of the decompression procedure for the MUXs Network.

Based on the Equation 2, in the following section, we will address the following question: Given a MUXs decompressor and the test patterns, what is the probability that the Equation 2 is solvable? The probability will enable us to predict the compression ratio. As described in the Section 3, each r_i only contains single "1", thus the set of y_i only consists at most M different elements which are $x_1, \ldots, x_M$. Hence, observing the following two propositions:

Prop. 1: The equation is solvable;

Prop. 2: For any vector y_j ($1 \le j \le N$), there is at least one element in X: x_i ($1 \le i \le M$) that is compatible with y_j. The definition of "compatible" of two vectors V_i, V_j is: $\forall q$, then ($V_i[q] = V_j[q]$) or ($V_i[q]$=X) or ($V_j[q]$=X).

Prop. 1 and *Prop. 2* are equivalent. Hence, we can compute the probability that the Equation 2 is solvable using the probability that there is at least of one x_i ($1 \le i \le M$) that is compatible with y_j ($1 \le j \le N$), which can be calculated easily.

Let p be the specified bit density of the test set. The probability that two bits are compatible can be calculated as: $(1-\frac{p^2}{2})$. Let P_1 represent the probability that two T-bits vectors are compatible, then P_1 can be calculated as the following equations:

$$P_1 = (1-\frac{p^2}{2})^T$$

Let P_2 be the probability that for a specified element y_j in Y, there is at least one x_i which is compatible with y_j. Then P_2 can be calculated as:

$$P_2 = 1-(1-P_1)^M$$
$$= 1-(1-(1-\frac{p^2}{2})^T)^M$$

Let $P_{(M,N,1)}$ be the probability that for any y_j ($1 \le j \le N$) in Y, the probability that there is at least one x_i ($1 \le i \le M$) that is compatible with y_j. Then $P_{(M,N,1)}$ can be calculated as:

$$P_{(M,N,1)} = (1-(1-(1-\frac{p^2}{2})^T)^M)^N$$

The $P_{(M,N,1)}$ is the probability which describes the *Prop. 2*. As said in the above paragraph, *Prop. 2* is equivalent with the *Prop. 1*. Thus, $P_{(M,N,1)}$ also gives us the probability that the Equation 2 is solvable. Let us consider a special example of this calculation. If M is equal to 1, this will mean that there is only one external pin to drive the massive scan chains. The external pin should be assigned as "1" or "0", then the probability that the Equation 2 is solvable is:

$$P_{(1,N,1)} = (1-(1-(1-\frac{p^2}{2})^T)^1)^N$$
$$= (1-\frac{p^2}{2})^{T\times N}$$

This result can verified from other way: If a pattern can be driven by one external input, any two row vectors in it should be compatible each other. To each two row vectors, the compatible probability is $(1-\frac{p^2}{2})^T$. Thus the probability that any two vectors are compatible is $(1-\frac{p^2}{2})^{T\times N}$. This is same as the calculation result of $P_{(1,N,1)}$.

From the $P_{(M,N,1)}$, we can see, in order to find a decompression MUXs network to decode a given deterministic pattern, $(1-\frac{p^2}{2})^T$ must be kept as large as possible. Thus, if the test patterns can't be tailored, we should select a small T; if the test patterns can be tailored, the test patterns must be generated with more X-bits as possible as can. This will help us to raise the compression ratio. However, in this case, more X-bits means that great number of test patterns. Thus, the tradeoff must be considered when we custom the ATPG algorithm.

5. Experimental Data

This section shows some experimental results to verify efficiency of the proposed MUXs decompression. We write a software simulator to simulate the behavior of the MUXs network. All of the simulator and equation solver is written by C language. The program is executed on a PC with a P4 1.6G processor and 768Mb RAM.

The first experiments, MUXs networks are designed to decompress the test patterns generated by *MINTEST[17]*. The results are shown in Table 1. In table 1, N is the internal scan chains and L is the length of longest scan chain. M is the external inputs of MUXs network. *ODV* and *CDV* is *Original Data Volume* and *Compressed Data Volume*. *CR* is calculated as:

$$CR = 1-(CDV/ODV)$$

X-ratio is the X-bits ratio in the total patterns. From the data in the table 1, to the given deterministic test pattern(*MINTEST*), if the MUXs propose network is inserted, the number of external inputs will be compacted dramatically. Due to this reduction, about 80% compression ratio can be obtained on average.

Table 1. The compressed data volumes of MUXs Network based on MINSTEST patterns

	N	L	MINTEST			Proposed MUXs network		
			Pat.	X-ratio	ODV	M	CDV	*CR*
S13207	100	7	233	93.2%	163,100	10	26,096	84.0%
S15850	100	7	94	83.7%	57,434	13	12,502	78.2%
S35932	100	18	12	38.6%	21,156	11	3,888	81.6%
S38417	100	17	68	68.1%	113,152	22	36,992	67.3%
S38584	100	15	110	82.3%	161,040	15	31,350	80.5%

Table 2. Comparison to XOR-based Techniques

Cores	XOR-based			Proposed
	XOR [11]	XOR +Overlap[20]	Mutation +XOR[9]	
S13207	25,344	14,145	15,783	26,096
S15850	22,784	13,919	10,798	12,502
S35932	7,128	4,492	3,972	*3,888*
S38417	89,856	52,793	42,264	*36,992*
S38584	38,796	26,644	22,636	31,350

Table 3. Comparison to some recent compression techniques

Cores	Golomb[6]	Nine-Coded [22]	Vail-Tail [28]	LPPW[26]	Broadcast Scan [3]	Proposed
S13207	41,658	28,852	21,081	99,252	85,546	26,096
S15850	92,054	19,309	20,969	--	76,030	12,502
S35932	59,537	--	11,632	17,368	9,136	3,888
S38417	92,054	65,512	56,495	820,715	127,932	36,992
S38584	104,111	55,510	69,964	303,072	129,580	31,350

In the second experiment, we compare our method to the previous scan chains hiding method: XOR-based techniques. Test data volumes of these techniques are listed in Table 2. Compared to the recent "XOR+Overlap" technique, three of five circuits can obtain the better results. Compared to all the XOR techniques, two best results can be obtained by the proposed scheme.

The MUXs decompressor can also be used as a stand-lone test data/test application time reduction scheme. The data listed in Table 3 compares the test data volume of proposed method to the previous compression methods. Golomb[6], Nine-Code[22] and Variable-Tail Code[28], are based on coding technique. LPPW[26] and Broadcast Scan[3] are based on new scan architectures. From the table, the better results can always be obtained by the proposed method compared to the coding techniques even the same deterministic test patterns(*MINTEST*) are used. Compared the Illinois scan architectures, our method also can achieve the further reduction of test data volume and test application time.

6. Conclusions

A periodically alterable MUXs decompressor based scan data compression/decompression technique was presented to shorten the scan chains, as can reduce the test data and test application time. Compared to previous "hiding" technique, more compression ration and test application reduction can be achieved. Furthermore, the proposed method can be applied to the embedded cores, and it also be used as a stand-alone compression technique for large commercial circuit or embedded core.

7. References

[1] ITRS. *2001 Edition. http://public.itrs.net.*

[2] K. J. Lee, J. J. Chen, and C. H. Huang, "Using a single input to support multiple scan chains", *Proc. of ICCAD*, pp.74-78, 1998.

[3] I. Hamzaoglu and J.H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores", *Proc. of DFT,* pp. 260-267, 1999.

[4] M. Sharma, J. H. Patel, J. Rearick, "Test Data Compression and Test Time Reduction of Longest-Path-Per-Gate Tests based on Illinois Scan Architecture", *Proc. of VTS*, pp.25-20, 2003.

[5] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, T. W. Williams, "A Reconfigurable Shared Scan-in Architecture", *Proc. of VTS, pp. 9-14, 2003.*

[6] A. Chandra, K. Chakrabarty, "System-on-a-Chip Test Data Compression and Decompression Architectures Based on Golomb Codes", *Trans. on CAD,* 20:113-120, Mar, 2001.

[7] P. T. Gonciari, B. Al-Hashimi, and N, Nicolici, "Integrated Test Data Decompression and Core Wrapper Design for Low-Cost System-on-a-Chip Testing", *Proc. of ITC,* pp. 64-73, 2002.

[8] S. Kajihara, Y. Doi, L. Li and K. Chakrabarty, "On Combining Pinpoint Test Set Relaxation and Run-

Length Codes for Reducing Test Data Volume". *Proc. of ICCD*, pp.387-396, 2003.

[9] S. Reda, A. Orailoglu, "Reducing Test Application Time Through Test Data Mutation Encoding", *Proc. of DATE,* pp. 387-393, 2002.

[10] A. Jas, B. Pouya, N.A. Touba, "Virtual Scan Chains: A Means for Reducing Scan Length in Cores", *Proc. of VTS*, pp.73-78, 2000.

[11] I. Bayraktaroglu and A. Orailoglu, "Test volume and application time reduction through scan chains concelment", *Proc. of DAC,* pp. 151-155, 2001.

[12] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, "Embedded Deterministic Test", *Trans. On CAD*, 23(5), 2004.

[13] N. Sitchinava, S. Samaranayake, R. Kapur, E. Gizdarski, F. Neuveux, T. W. Williams, "Changing the Scan Enable during Shift", *Proc. of VTS, 2004.*

[14] L. Wang, X. Wen, H. Furukawa, F. Hsu, S. Lin, S. Tsai, K. S. Abdel-Hafez, and S. Wu, "VirtualScan: A New Compressed Scan Technology for Test Cost Reduction", *Proc. of ITC,* 2004.

[15] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, T.W. Williams, "A Reconfigurable Shared Scan-In Architecture", *Proc. of VTS*, pp. 9-14, 2003.

[16] J. C. Culberson, "Iterated Greedy Graph Coloring and the Difficulty Landscape", Technical Report , TR-92-07, University of Alberta, Canada, 199, http://www.cs.ualberta.ca/~joe/Coloring/

[17] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits", *Proc. of ICCAD*, pp. 283-289, 1998

[18] H. K. Lee and D. S. Ha, "On the generation of test patterns for combinational circuits". Technical Report 12.93, Department of Electrical Eng., Virginia Polytechnic Institute and State University.

[19] H. K. Lee and D. S. Ha, "An efficient parallel fault simulator", *Proc. of DAC*, pp. 336-340, 1992.

[20] W. Rao, I. Bayraktaroglu and A. Orailoglu, "Test Application Time and Volume Compression Through Seed Overlapping", *Proc. of DA*C, pp. 732-737, 2003.

[21] X. Sun, L. Kinney and B. Vinnakota, "Combining Dictionary Coding And LFSR Reseeding For Test Data Compression", *Proc. of DAC*, 2004

[22] M. Tehranipour, M. Nourani, K. Chakrabarty, "Nine-Coded Compression Technique with Application to Reduced Pin-Count Testing and Flexible On-Chip Decompression", *Proc. of DATE*, 2004.

[23] I. Bayraktaroglu, A. Orailoglu, "Decompression Hardware Determination for Test Volume and Time Reduction through Unified Test Pattern Compaction and Compression", *Proc. of VTS*, 2003

[24] C.V. Krishna, A. Jas, and N. A. Touba, "Reducing test data volume using LFSR reseeding with seed compression", *Proc. of ITC*, 2002

[25] C.V Krishna, N. A. Touba, "3-Stage Variable Length Continuous-Flow Scan Vector", *Proc. of ICCAD*, 2003.

[26] M. Sharma, J. H. Patel, J. Rearick, "Test Data Compression and Test Time Reduction of Longest-Path-Per-Gate Test based on Illinois Scan Architecture", *Proc. of VTS*, 2003.

[27] L. Li, K. Chakrabarty, "Test Data Compression Using Dictionaries with Fixed-Length Indices", *Proc. of VTS*, 2003.

[28] Y. Han, X. Li, Y. Xu, H. Li, "Test Resource Partitioning Using Variable-Tail Code", Acta Electronica Sinca, 32(8), pp. 1346-1350,2004。

[29] L. Li, K. Chakrabarty, "Test Set Embedding for Deterministic BIST using a reconfigurable interconnection network", IEEE Trans. On CAD, 23(9), 2004.