On Test Data Compression and Decompression for Multiple Circuits

Irith Pomeranz[1] and Sudhakar M. Reddy[2]
School of Electrical & Computer Eng.
Purdue University
W. Lafayette, IN 47907

Electrical & Computer Eng. Dept.
University of Iowa
Iowa City, IA 52242

On-chip decompression of compressed test data is used in [1]-[6] to reduce the test data volume for scan circuits. In this work, we consider the design of an on-chip test data decompresser for *multiple* scan circuits implemented on the same chip.

The configuration considered in [1]-[6] is shown in Figure 1, while the configuration we consider here is shown in Figure 2. In Figure 2, the same decompresser D drives two circuits, C_1 and C_2. For simplicity, we assume that the two circuits have the same number of scan chains, N_S (if this assumption does not hold, it is possible to design D for the larger number of scan chains and connect only a subset of the outputs of D to the other circuit). Test data is loaded to the circuits through N_I inputs that drive D. The decompresser expands every N_I-bit wide vector into an N_S-bit wide vector, which is applied to the circuits C_1 and C_2 through their scan chains. If the length of the longest scan chain in C_i $(i = 1,2)$ is L, it takes L clock cycles to apply a test t to C_i. In every clock cycle, an N_I-bit wide vector r_k is applied to the decompresser, the decompresser produces an N_S-bit wide vector v_k, and the scan chains are shifted by one position with v_k as the new scan-in vector. After L clock cycles, the vectors $v_0, \cdots, v_{L-1}$ define an input combination of C_i. The vectors $r_0, \cdots, r_{L-1}$, and the corresponding vectors $v_0, \cdots, v_{L-1}$ must be selected such that the test t is obtained on the inputs of C_i.

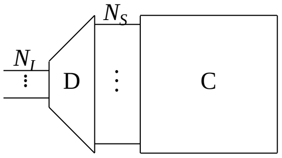

Figure 1: A basic decompression scheme

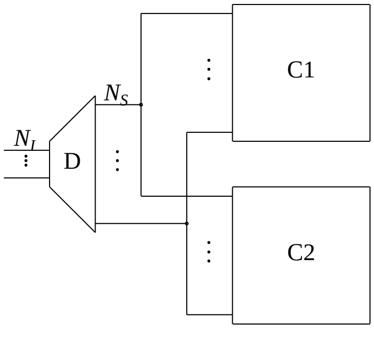

Figure 2: A decompression scheme for two circuits

The advantage of using a single decompresser D instead of two decompressers, D_1 for C_1 and D_2 for C_2, is that it is possible to share the logic required for D_1 and D_2 in a single decompresser D. Sharing is effective since test sets for two different circuits are likely to contain some of the same vectors. To maximize the amount of sharing, we propose a procedure for modifying the test sets applied to C_1 and C_2 so as to maximize the number of vectors they have in common. The applications of a single decompresser D connected to two circuits as in Figure 2 are as follows.

(1) Large designs, especially core-based designs, contain multiple circuits with well-defined inputs and outputs such as C_1 and C_2 in Figure 2. For such designs, the proposed procedure offers an alternative to the design of two separate decompressers.

(2) When a single circuit $\hat{C}$ has a large number of scan chains $\hat{N}_S$ and the logic required for implementing a decompresser $\hat{D}$ with a large number of outputs is large, it is possible to partition the scan chains of $\hat{C}$ into two (approximately) equal sets with $N_S \simeq \hat{N}_S / 2$ scan chains in each set. Instead of designing two decompressers D_1 and D_2 each with N_S outputs, the proposed procedure can be used to design a single decompresser D with N_S outputs that drive both sets of scan chains. Although in the worst case, D will be as large as D_1 and D_2 together, the sets of vectors for the two sets of scan chains can be made to overlap such that sharing of the logic between D_1 and D_2 will be possible.

We assume that a test set T_i is given for C_i, and that T_i needs to be applied to C_i through D for $i = 1,2$. Given a test set T_i for a circuit C_i, we first partition T_i into scan vectors. We pad the last vector with unspecified values if some of the scan chains are shorter than others. A test t_j is thus divided into vectors $v_{j0}, v_{j1}, \cdots, v_{j(L-1)}$. To apply t_j to the circuit, we apply $v_{j0}, v_{j1}, \cdots, v_{j(L-1)}$ in *reverse order* to store $v_{j0}, v_{j1}, \cdots, v_{j(L-1)}$ in the scan chains in the correct order.

The set of vectors obtained by partitioning all the tests in T_i is denoted by V_i. The set V_i is sufficient for creating every test in T_i by applying one or more vectors out of V_i consecutively. We also define $V = V_1 \cup V_2$. This is the set of vectors required for testing both C_1 and C_2. To allow translation of every vector $v \in V$ into a decompresser input vector r, the decompresser must have at least $\lceil \log_2 |V| \rceil$ inputs. Thus, we have $N_I \geq \lceil \log_2 |V| \rceil$ in Figure 2, and the size of V determines the number of decompresser inputs. Our goal is to minimize the size of V in order to, indirectly, reduce the size of the decompresser. This is similar to the approach taken in [6]. The decompresser size is reduced due to two effects. (1) If V is reduced such that N_I can be reduced, the decompresser has fewer inputs, which typically implies a smaller size. (2) Even for the same value of N_I, a smaller set V implies more don't-cares on the outputs of D corresponding to vectors that do not need to be produced by D. These don't-cares can be used to minimize the size of D.

The size of V is largely determined by the test sets T_1 and T_2. We minimize V by using the following techniques.

Some of the vectors in V have unspecified values. As a result, V may contain compatible vectors. We combine as many

---

[1] Research supported in part by NSF Grant No. CCR-0098091 and in part by SRC Grant No. 2001-TJ-950.

[2] Research supported in part by NSF Grant No. CCR-0097905 and in part by SRC Grant No. 2001-TJ-949.

compatible vectors as possible in order to reduce the size of V. Here, we may combine vectors from V_1 with vectors from V_2 in order to optimize the joint set V. This is not possible when separate decompressers are designed for C_1 and C_2.

Next, we attempt to replace scan vectors in V that appear only once in $T_1 \cup T_2$ by other vectors that appear in T_1 or T_2. Thus, if a test $t_j \in T_i$ consists of scan vectors $v_{j0}, v_{j1}, \cdots, v_{j(L-1)}$ and v_{jk} appears only once in $T_1 \cup T_2$ (i.e., v_{jk} appears only in t_j and only once in t_j), we attempt to replace v_{jk} in t_j by a different vector $\hat{v}$ that appears in $T_1 \cup T_2$ (in one or more tests of $T_1 \cup T_2$). Every time we replace a vector such as v_{jk} by a vector $\hat{v}$, we reduce the number of vectors in V by one.

To accept the replacement of v by $\hat{v}$ in a test $t_j \in T_1$ (or $t_j \in T_2$), we require that all the faults of C_1 (or C_2) that are detected by t_j would be detected either by the modified test t_j, or by one of the other tests in T_1 (or T_2).

As replacement vectors $\hat{v}$, we first consider vectors that appear more than once in $T_1 \cup T_2$. In this way, we do not increase the number of appearances of vectors that appear only once in $T_1 \cup T_2$, and we maximize our ability to remove these vectors later. Only after all the possible replacements using such vectors $\hat{v}$ have been made, we consider replacement vectors that appear only once in $T_1 \cup T_2$.

Once the replacement of single vectors is complete, we consider pairs of vectors $v_{j_1 k_1}$ and $v_{j_2 k_2}$, each appearing only once in T_1 or T_2. Let $v_{j_i k_i}$ appear in test t_{j_i}, for $i = 1,2$. We attempt to replace both $v_{j_1 k_1}$ and $v_{j_2 k_2}$ together by a new vector $\hat{v}$, which may or may not appear in V. Every time we replace a pair of vectors by a new vector, we remove two vectors from V and add at most one. Thus, the size of V is reduced.

We select a replacement vector $\hat{v}$ (not necessarily in V) based on $v_{j_1 k_1}$ and $v_{j_2 k_2}$ as follows. We define a vector v_c which is equal to $v_{j_1 k_1}$ and $v_{j_2 k_2}$ in every bit where they are equal, and is unspecified otherwise. We then fill the unspecified values in v_c exhaustively (if the number of unspecified values in v_c is small) or randomly (if the number of unspecified values in v_c is large). Every vector obtained from v_c by filling the unspecified values is used as a candidate for $\hat{v}$.

We replace $v_{j_1 k_1}$ and $v_{j_2 k_2}$ in t_{j_1} and t_{j_2}, respectively, by every candidate vector $\hat{v}$. If t_{j_1} and t_{j_2} continue to detect all the faults they detected before the replacement, the replacement is accepted and no additional candidates are considered for $\hat{v}$.

The following differences between the proposed approach and the one in [6] are important. First, the approach in [6] is designed for a single circuit while we consider circuit pairs. In addition, in [6], as many values in T_i as possible are unspecified before V_i is defined. This maximizes the number of compatible vector pairs in V_i, thus allowing its size to be reduced. However, this procedure unspecifies values in T_i without considering whether or not these values actually help create compatible vectors. The proposed procedure alleviates this problem by using the tests as they are, but allowing vectors to be replaced by other vectors when this reduces the number of vectors.

We applied the procedure described above to pairs of ISCAS-89 and ITC-99 benchmark circuits. We assume that all the circuit inputs are scanned, and we partition them into $N_S = N_{S,min}, \cdots, N_{S,max}$ scan chains. We use N_S which is always a power of two, with $N_{S,min} = 16$ and $N_{S,max} < \min\{N_{PI1}, N_{PI2}\}$ (N_{PIi} is the number of inputs of circuit i). As test sets for ISCAS-89 benchmark circuits we use compacted deterministic test sets. For ITC-99 benchmark circuits the test sets are selected out of 100,000 random patterns and compacted by reverse order fault simulation.

For comparison, we apply the proposed procedure to the sets V_i obtained for each circuit separately, as well as to the combined set $V = V_1 \cup V_2$.

In Table 1 we show information about the sets V_i constructed for each circuit separately, and the set V constructed jointly by the proposed procedure. After the circuit names, we show the number of scan chains. Under column Vi, for $i = 1,2$, we show the number of vectors in V_i, and the number of decompresser inputs $N_{Ii} = \lceil \log_2 |V_i| \rceil$. Under column V we show the number of vectors in V, and the number of decompresser inputs $N_I = \lceil \log_2 |V| \rceil$.

It can be seen that V is significantly smaller than V_1 and V_2 combined. This implies that a decompresser designed based on V would be smaller than two decompressers based on V_1 and V_2 separately.

Table 1: Results

			V1		V2		V	
circ1	circ2	NS	size	NI1	size	NI2	size	NI
s953	s1423	16	221	8	153	8	127	7
		32	151	8	78	7	140	8
s953	b04	16	221	8	351	9	134	8
		32	151	8	210	8	160	8
s953	b11	16	221	8	155	8	131	8
		32	151	8	89	7	141	8
s1423	b04	16	153	8	351	9	151	8
		32	78	7	210	8	135	8
		64	52	6	141	8	105	7
s1423	b11	16	153	8	155	8	142	8
		32	78	7	89	7	133	8
b04	b11	16	351	9	155	8	142	8
		32	210	8	89	7	167	8
s5378	s9234	16	1140	11	1200	11	545	10
		32	695	10	870	10	613	10
		64	399	9	444	9	601	10
		128	200	8	222	8	377	9
s5378	s35932	16	1140	11	30	5	250	8
		32	695	10	51	6	302	9
		64	399	9	69	7	324	9
		128	200	8	70	7	227	8
s9234	s35932	16	1200	11	30	5	383	9
		32	870	10	51	6	395	9
		64	444	9	69	7	345	9
		128	222	8	70	7	262	9
s15850	b14	16	1905	11	2940	12	1056	11
		32	1578	11	1627	11	668	10
		64	929	10	946	10	999	10
		128	485	9	576	10	526	10
		256	291	9	386	9	375	9

References

[1] A. Jas, J. Ghosh-Dastidar and N. A. Touba, "Scan Vector Compression/Decompression Using Statistical Coding", in Proc. VLSI Test Symp., 1999, pp. 114-120.

[2] A. Chandra and K. Chakrabarty, "Frequency-directed Run-length (FDR) Codes with Application to System-on-a-chip Test Data Compression", in Proc. VLSI Test Symp., April 2001, pp. 42-47.

[3] I. Bayraktaroglu and A. Orailoglu, "Test Volume and Application Time Reduction Through Scan Chain Concealment", in Proc. Design Autom. Conf., June 2001, pp. 151-155.

[4] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors", in Proc. Intl. Test Conf., Oct. 2001, pp. 748-757.

[5] G. Eide, "Embedded Deterministic Test - DFT Technology for Low-Cost IC Manufacturing Test", www.mentor.com/dft.

[6] S. M. Reddy, K. Miyase, S. Kajihara and I. Pomeranz, "On Test Data Volume Reduction for Multiple Scan Chain Designs", in Proc. VLSI Test Symp., April 2002.