

Output Compression for IC Fault Detection Using Compressive Sensing

Stephen J. Tarsa and H.T. Kung
Harvard University, Cambridge, MA

Abstract—The process of detecting logical faults in integrated circuits (ICs) due to manufacturing variations is bottlenecked by the I/O cost of scanning in test vectors and offloading test results. Traditionally, the output bottleneck is alleviated by reducing the number of bits in output responses using XOR networks, or computing signatures from the responses of multiple tests. However, these many-to-one computations reduce test time at the cost of higher detection failure rates, and lower test granularity. In this paper, we propose an output compression approach that uses compressive sensing to exploit the redundancy of correlated outputs from closely related tests, and of correlated faulty responses across many circuits. Compressive sensing’s simple encoding method makes our approach attractive because it can be implemented on-chip using only a small number of accumulators. Through simulation, we show that our method can reduce the output I/O bottleneck without increasing failure rates, and can reconstruct higher granularity results off-chip than current compaction approaches.

I. INTRODUCTION

Detecting faults in integrated circuits (ICs) that arise due to manufacturing variations is a notoriously difficult and costly problem. Colloquially, large chip design projects employ far more engineers working on IC verification than initial circuit design. In the case of military-grade ICs, reliability and security demands are a significant barrier for manufacturers [1]: in the past, the cost of testing MIL-SPEC ICs has led, in part, to some large companies leaving the marketplace and focusing solely on consumer electronics [2]. Going forward, battlefield dependence on devices driven by increasingly complex ICs means that fast accurate fault detection is becoming even more difficult and critical.

In practice, the logical-fault detection process proceeds by exercising the logical states of newly manufactured ICs using a sequence of test vectors, then verifying test responses against simulated fault-free responses. The steps of this process require first identifying a minimal set of test vectors whose application will reveal a near-total percentage of possible faults [3] [4] [5]. Second, the desired tests are applied by shifting test stimuli into the circuit using an industry standard (JTAG) interface [6]. And third, test responses must be shifted out of the circuit for examination. When faults are detected, the process may be repeated with a second set of more expressive test vectors to collect additional information on circuit behavior. Since the number of states taken on by VLSI circuits is huge, the volumes of input test data and output response data make testing I/O a bottleneck to the manufacturing process.

In addition to past efforts to reduce scan-in time for test vectors (e.g. [7], [8], and [9]), researchers have also devel-

oped methods for compacting test-responses on-chip to reduce output. Lossless compression circuitry is generally considered too costly, so lossy compaction methods are preferred. While current compaction methods improve test time, they lead to reduced test response granularity and increased false negative detection rates. Such detection failure is termed *aliasing*, since faulty test responses incorrectly appear fault-free due to the compaction process.

The most widely applied compaction method is space compaction, which uses XOR networks to reduce many bits from a single test response down to a few bits to be used for detection [10] [11] [12]. XOR networks are easy to implement, but can have high aliasing rates, since any even number of faulty response bits will lead to an erroneously fault-free detection bit when XORed together. Furthermore, space compaction detects faults to test vector granularity, and cannot distinguish between multiple faults expressed in the same test; this behavior can occur unexpectedly, or by design in order to reduce total test time.

The second compaction method used is time compaction, in which a signature is computed from multiple sequential test responses using a Multiple Input Signature Register (MISR). MISRs can yield larger reductions in output with fewer false negatives than XOR networks, but they also severely reduce detection granularity. With this method, faults are detected to one-in-many tests based on a single signature, and little diagnostic information is made available to designers. MISRs are also intolerant to test responses with unknown ‘X’ values, which arise from uninitialized flip-flop states. Additional circuitry must typically be included to mask off these ‘X’ values, increasing hardware cost [13].

We instead propose to apply the techniques of compressive sensing (CS) [14] to implement lossless output *compression* with only simple on-chip circuitry. With CS, we compress response data using a small number of accumulators that collect linear combinations over batches of test responses. The resulting reduced-dimension measurement vectors are then shifted out of the circuit, and results are reconstructed off-chip using a sparse signal recovery algorithm. This has several advantages over current methods:

- 1) Lossless compression of test responses is possible with simple on-chip circuitry
- 2) Successful signal reconstruction yields pin-level data off-chip, a result that is impossible with current lossy compaction methods

- 3) Aliasing rates are lower than both XOR networks, and MISRs that compact small numbers of test responses for highest granularity results

We first formulate the fault detection problem in the compressive sensing framework. Since CS requires that signals be expressible with a small number of non-zero components (i.e. sparse), we present a method for finding bases that sparsely express faults. Our technique works by ordering input test vectors to promote the locality of correlated test responses, and exploiting the resulting correlations between successive tests using the Karhunen-Loève transform (KLT). We then show how faulty behaviors decoded from previously examined circuits can also be encoded in this basis, to exploit redundancy among circuits. We compare our method to both time and space compaction to show gains in compression rate, output granularity, and false negative rate. Finally, we simulate the testing process to demonstrate performance.

II. COMPRESSIVE SENSING OUTPUT COMPRESSION FORMULATION

Compressive sensing states that sparse signals encoded, or measured, via linear combination can later be reconstructed using a small number of these measurements, provided the measurement coefficients satisfy a *Restricted Isometry Property (RIP)*. Reconstruction is possible using algorithms for ℓ_1 minimization by linear programming [15], or basis pursuit [16] [17]. Furthermore, any basis for sparse representation can be used for decoding, provided the number of measurements is approximately proportional to the number of non-zero components required to represent the signal in that basis. Sufficient measurement matrices satisfying RIP include those with normally distributed or Bernoulli random entries, and those whose rows are chosen from the Fourier basis functions [18].

Using these facts, we can express test response compression for fault detection in terms of CS's encoding and decoding steps. Defining X_t to be a batch of n test responses collected at time t , we write the encoding process as:

$$y_t = \Phi X_t \quad (1)$$

where Φ is an $m \times n$ ($m \ll n$) matrix with randomly distributed entries that have been predetermined prior to test time. m is chosen to be approximately proportional to k , the number of non-zero values expected in the decoded signal vector. When Φ contains Bernoulli $\{0,1\}$ entries, $m \approx 4k$. Recovering test outputs off chip requires solving the following minimization problem:

$$\min \|X_t^g - X_t\|_1 \text{ s.t. } y_t^g - y_t = \Phi \Psi_t (X_t^g - X_t) \quad (2)$$

where X_t^g is a vector of simulated gold circuit outputs, y_t^g the corresponding simulated measurements, and Ψ_t is a sparsifying basis for $(X_t^g - X_t)$. Since a single fault may cause the values of many test outputs to deviate from the gold circuit, we are interested in finding an appropriate Ψ_t so that faults are represented with only a few components.

III. SPARSIFYING BASES FOR TEST OUTPUTS

A. Exploiting Correlations Between Related Tests

Consider batches of test outputs collected at times $t = 1, 2, \dots, T$. The KLT basis is spanned by the eigenvectors of their autocorrelation matrix: $A = \sum_{t=1}^T (X_g - X_t)(X_g - X_t)'$ [19]. When circuit outputs in multiple batches are correlated, the KLT is a sparsifying basis that represents these correlated outputs with common components. For example, at the extreme when all X_t 's are linearly dependent, then they can be represented by a single component in the KLT basis.

Observe Figure 1(a), in which input test vectors for one of our benchmark circuits are placed in an *alignment matrix*, according to their Gray Code ordering. Based on the expectation that two tests will share many of the same sensitized paths when their test vectors only differ by a few bits, then the rows of the corresponding output response matrix will show correlation. This will hold when faults occur in gates along those shared paths, or when the paths are fault free, as Figure 1(b) shows for our test circuit.

We therefore take advantage of these correlations using the KLT by arranging test vectors in an n -row alignment matrix, and applying tests in batches according to the columns of the matrix. As batches of m measurements are offloaded and test responses are decoded, we use the trailing T column vectors to calculate the KLT basis. The result, Ψ_t , is used to sparsely represent the next batch, X_t at time t .

The alignment parameters T and n are circuit specific – if T is too large, then Ψ_t will likely do a poor job of sparsely representing X_t because it may include batches with uncorrelated test results. In general, Ψ_t is a sparsifying basis for X_t whenever X_t can be written as a small linear combination of the preceding batches' KLT components. In practice, simulations of the fault-free gold circuit outputs can be used to determine the appropriate values for T , n , and m .

B. Exploiting Correlations Among Circuits

Unavoidably, test outputs will arise that are not sparsely represented based on the behavior of prior tests. This can occur when circuit architecture makes finding an alignment matrix that is successful for all batches difficult, or when unexpected faulty responses occur that deviate substantially from prior test outputs. In this case, off-chip decoding will fail due to insufficient measurements. However, once these test responses are retrieved, they can be incorporated into future Ψ_t 's by adding the observed vector to the KLT: $A = \sum_{c \in C} \sum_{t=1}^T (X_g - X_t^c)(X_g - X_t^c)'$ where C denotes an index set of previously examined circuits. Therefore, when decoding fails, we offload the current vector at full rate, paying a one-time penalty in order to sparsely represent the faulty responses in future tests.

IV. IC TESTING PROCESS FLOW

In order to implement the sparsification procedures of Section III, Figure 2 shows an IC testing flow that includes two phases: training and circuit testing. The training stage is used to capture common faulty test responses in initial Ψ_t

I_1	I_2	...	I_T	I_{T+1}															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32				
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48				
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64				
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80				
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96				
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112				
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128				

(a)

X_1	X_2	...	X_T	X_{T+1}															
106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106				
106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106				
106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106				
106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106				
106	106	86	86	106	106	86	86	106	106	86	86	106	106	106	106				
86	86	86	86	86	86	86	86	86	86	86	86	86	106	106	106				
106	106	86	86	106	106	86	86	106	106	86	86	106	106	106	106				
86	86	86	86	86	86	86	86	86	86	86	86	86	106	106	106				

(b)

Fig. 1. (a) In order to promote locality of correlated test output values among adjacent columns, input vectors are ordered in increasing order by their integer representation across rows in a *alignment matrix*. They are then applied to the circuit in batches by column, denoted I_t . (b) For our benchmark circuit, rows of output responses do indeed show high correlation that can be exploited for compression.

bases so that they are compressible when the testing phase begins. In practice, designers model expected faulty behavior in simulation (e.g. to ensure that a high percentage of faults are detected by a set of test vectors), and we use these simulated faulty responses as training data. During the testing phase, test vector batches are applied as described in Section IIIA, and results are decoded off-chip based on m measurements. When decoding fails due to insufficient measurements, we offload the current vector at full rate, and add it to the autocorrelation matrix for future circuit tests.

V. PERFORMANCE ANALYSIS & COMPARISON

We compare our method to space and time compaction by analyzing output reduction rates, aliasing rates, off-chip response granularity, and hardware cost.

A. CS-Based Output Compression

For a circuit with L_{out} output pins, the output compression factor using our method is given by:

$$1 - \frac{m \log_2(\frac{1}{2}n2^{L_{out}})}{nL_{out}} \quad (3)$$

Raw output requires nL_{out} bits to offload n outputs. In contrast, our compressive sensing method requires only m measurements for a batch of n results. The maximum value in each accumulator is $\frac{1}{2}n2^{L_{out}}$ since half of the Bernoulli random coefficients will be 1's on average, and the maximum value of each test output is $2^{L_{out}}$.

Aliasing occurs when the measurement vector of a faulty batch of test responses matches that of the simulated fault-free responses. This probability is given by:

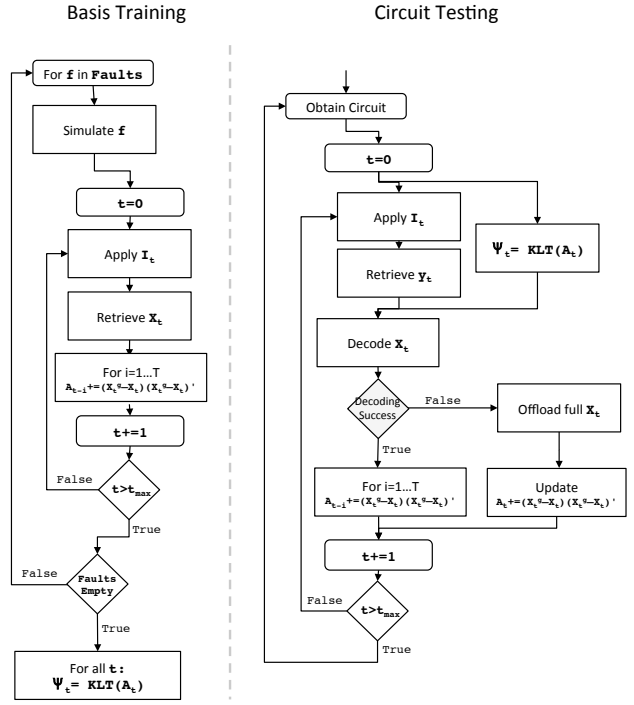


Fig. 2. Basis training is used to bootstrap output compression, and is conducted by simulating known fault behavior then calculating the KLT from batches of simulated test responses. Circuit testing proceeds by applying tests, offloading measurements, and decoding the results. Upon decoding failure, test responses are offloaded from the circuit at full rate in order to compress those responses in future circuits.

$$\Pr(y_t^g = y_t | X_t^g \neq X_t) = \frac{2^{(n-m)*L_{out}} - 1}{2^{n*L_{out}} - 1} \quad (4)$$

This expression is derived as follows: with L_{out} bits-per-response, all but one of the $2^{n*L_{out}}$ possible batch vectors are faulty. Since the system is under-determined by $n-m$ degrees of freedom, $2^{(n-m)*L_{out}}$ vectors satisfy the measurement vector y_t^g , and all but one of these is faulty.

The granularity of offloaded measurements is always to the pin-level. Either the vector is sparsely represented by m measurements, in which case it is successfully decoded, or it is offloaded at full rate upon decoding failure.

B. XOR Network Compaction

We next compare to space compaction using the X-Compact [12] method for XOR networks. Test responses with L_{out} bits are reduced to L_{sig} detection bits by XORing together L_{sig} distinct combinations of response bits. The output reduction rate is therefore $1 - (L_{sig}/L_{out})$.

Whenever an even number of faulty bits are XORed together, the result will be erroneously fault-free; aliasing occurs when this happens at all L_{sig} outputs. The overall aliasing rate is therefore driven directly by the output reduction rate, since a greater number of unique XOR combinations for a fixed L_{out} reduces the probability that all detection bits will be erroneous.

We quantify this aliasing rate by simulating the response of random X-Compact networks to faults in the benchmark

circuit shown in Figure 5. The results are shown in Figure 3, and we see that aliasing increases sharply when fewer distinct XOR combinations are used. For this circuit, the best achievable compaction with an aliasing rate below 1% is only a 35% reduction. While an XOR network can be engineered for lower aliasing in response to known faults, unanticipated faults will cause unpredictable aliasing, potentially increasing the overall rate.

At all output rates, detection occurs to the test-vector level, since the L_{sig} bits only provide redundancy against aliasing. This method cannot distinguish between multiple faults exercised by a single test.

C. MISR Compaction

We next compare to MISR time compaction, assuming no unknown ‘X’ values. Combining s test responses with L_{out} bits into a single signature of L_{sig} bits leads to an output reduction of:

$$1 - \frac{L_{sig}}{s * L_{out}} \quad (5)$$

Adopting the model in [11], the aliasing rate can be calculated by the probability of hash collision:

$$\text{Pr(Collision)} = \frac{2^{sL_{out}-L_{sig}} - 1}{2^{sL_{out}} - 1} \quad (6)$$

For a fixed L_{out} , MISRs achieve best output reduction and lowest aliasing rate when s is large. However, this reduces fault detection granularity to 1-in- s tests. In Figure 3, we show what happens when we set $s = 2$, in order to maximize the granularity of off-chip results. In this case, 60% compaction can be achieved before aliasing rises above 1%.

In contrast, our CS-based method similarly achieves best compression and aliasing performance when n is large, however it suffers no such loss in granularity. Figure 3 shows that the aliasing rate decreases significantly when we increase the batch size to 32 and offload the maximum number of measurements given a fixed output budget. Achieving these low aliasing rates without a granularity penalty is unique to our CS-based method.

D. Required Circuitry

We next examine the hardware cost of our method, using flip flop count as a comparison metric. Our on-chip architecture is illustrated in Figure 4(b). Encoding requires m accumulators with a maximum value of $\log_2(\frac{1}{2}n2^{L_{out}})$. We use m seeded random $\{0, 1\}$ number generators to store each row of measurement coefficients. Since $m < n$, $n - 1$ unique initialization states are required. Flip flop count is then:

$$m \left[\log_2(n - 1) + \log_2\left(\frac{1}{2}n2^{L_{out}}\right) \right] \quad (7)$$

Both X-Compact and MISR methods require only L_{sig} flip flops, meaning that our method has a higher hardware cost, especially when m is large. However, it is possible to reduce the number of accumulators in exchange for a higher decoding failure rate in order to meet cost or area budgets.

VI. EVALUATION

To evaluate our methodology, we simulate the fault detection process using a benchmark circuit from the ISCAS-85 suite [20], shown in Figure 5. We simulate both the training and testing phases described in Section IV, testing 10,000 ICs. IC’s have a 10% probability of a fault occurring, and for these circuits, the number of faults present is normally distributed over the range 1 to 6. During the training phase, we set aside 5% of faults as ‘unanticipated’ behaviors during testing.

For our CS method, we tune parameters $n = 512$ and $T = 16$. Given that at most 6 faults should occur in any circuit, we test with $m = 24$ and $m = 16$. This second parameter value has a lower hardware cost, but a higher probability of decoding failure when many faults occur. To compare with time compaction, we use two signature methods: MISR-2, which takes in 2 8-bit test responses and outputs a single 6 bit signature, and MISR-4, which takes 4 8 bit test responses and outputs a single 7 bit signature. The first is chosen for highest granularity results, and the second for better compaction and aliasing performance. For both methods, the characteristic polynomial is defined by a randomly generated bit stream, fixed for each circuit. To compare with space compaction, we randomly generate X-Compact networks that output 5 bits, the best reduction for sub 1% aliasing based on Section III B.

The results are shown in Table I. Our CS method delivers output reduction rates of 90.68% and 93.64%, respectively. Though taking only $m = 16$ measurements increases the number of decoding failures by 7x, the total output is reduced since fewer measurements are taken for compressible faults. The CS method achieves an 8% output reduction improvement over MISR-4, and a 56% improvement over the XOR network approach. Additionally, we observe no aliasing with our method, while all compaction methods have non-zero aliasing.

The gains of our CS implementation come at the cost of a 45x increase in the number of flip flops over MISR-4, the second best compaction method. This may be a prohibitive cost

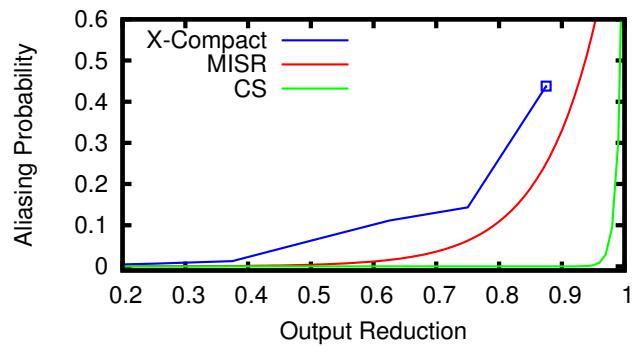


Fig. 3. Aliasing rate is plotted against output reduction rate for the three methods considered. For X-Compact, aliasing increases as the number of detection bits decreases. This leads to a maximum output reduction of only 35% for an aliasing rate below 1%. For an MISR taking two input signatures, aliasing performs better than X-Compact, and 60% compaction is achieved at 1% aliasing. With compressive sensing, we can reduce aliasing by using larger batches without reducing off-chip result granularity: when $n = 32$, 95% compaction can be achieved with 1% aliasing.

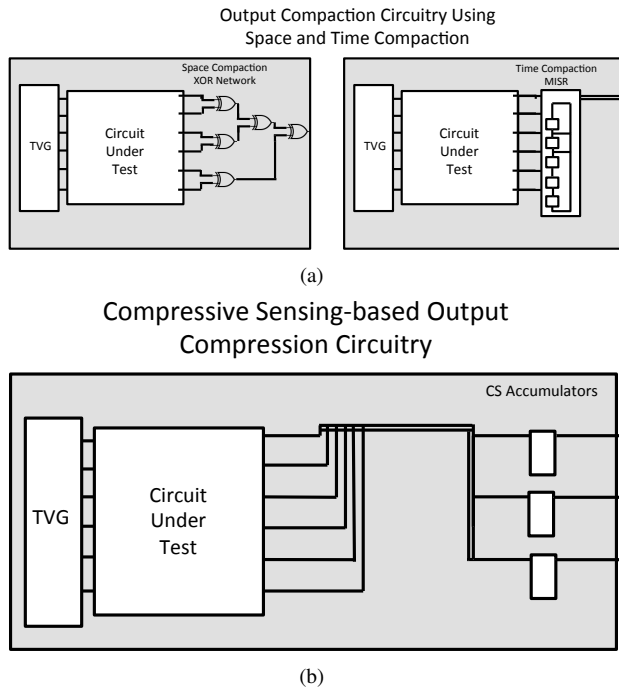


Fig. 4. (a) Circuitry for space and time compaction methods using XOR networks and MISR registers, respectively. (b) Our CS compression circuitry requires m accumulators to form compressive measurements from batches of test outputs. Our method requires more flip flops, but has significantly lower aliasing rates and higher off-chip granularity.

increase for testing small ICs, and we propose to investigate improvements based on analog domain encoding in the future work section.

VII. FUTURE WORK

A major avenue for future work will be extending our evaluation to additional circuits with varying size, gate count, path length, and fan-in properties. Beyond the examples in

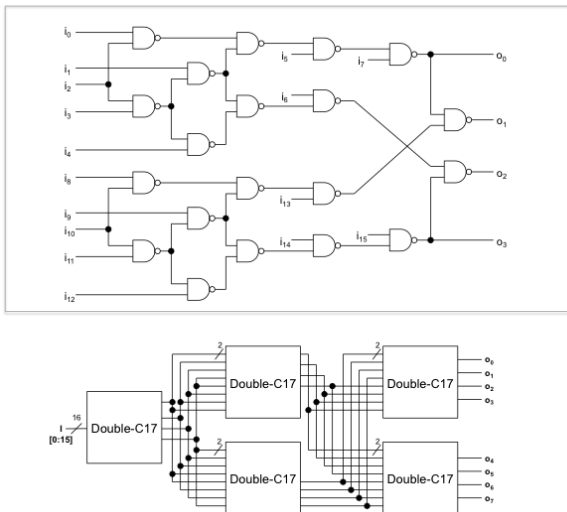


Fig. 5. Our benchmark circuit combines multiple c-17 circuits from the ICAS-85 suite, pictured above.

Method	Flip Flop Count	Output Reduction Rate	Aliasing Rate	Granularity	CS Decoding Failures
CS $m=16$	320	93.64%	0%	pin-level	77,166
CS $m=24$	483	90.68%	0%	pin-level	10,146
MISR-4	7	78.12%	5%	1-in-4 tests	-
MISR-2	6	62.50%	6%	1-in-2 tests	-
XOR	5	37.50%	1%	1-in-1 tests	-

TABLE I

the ISCAS-85 and ITC-99 benchmark suites, the work of [21] has developed methods for Monte Carlo simulation by analyzing prototype circuits for the above properties, and randomly generating similar circuits. Of particular interest is the analysis of our method under test sequences that exercise the internal states of flip flops in sequential circuits. With a larger number of tests to run, and sets of output results that toggle only internal memory states, we expect the amount of output correlation to increase, leading to better performance for our method.

A second major avenue of work is to investigate encoding pin outputs using analog circuitry. Analog circuitry exists for computing Fourier transforms, and as noted in Section II, it may be possible to piggyback off these components to encode signals at minor hardware cost. In addition, analog domain encoding would capture more information, including real-valued voltage readings that are useful for detailed circuit analysis.

VIII. CONCLUSION

In this paper, we presented a method for lossless output compression for the fault detection process that uses simple IC-independent test circuitry. We introduced new techniques for calculating bases that represent faults with a few non-zero coefficients. These techniques capture correlations between related fault tests and across circuits in the testing pipeline. Using the random linear encoding process of compressive sensing on-chip, and sparse signal recovery off-chip, we showed that we can reduce the size of test outputs without sacrificing granularity or increasing detection failure rates. This improves significantly upon current industry-standard methods of compaction in these respects. As a result, our CS-based compression method yields orders of magnitude more data to designers for fault isolation. Given the criticality of fault detection for MIL-SPEC ICs, where strict detection requirements are enforced, these gains have the potential to greatly improve the efficacy of testing and verification for designers and manufacturers.

ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement numbers FA8750-10-2-0115 and FA8750-10-2-0180. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government. The authors would like to thank the Office of the Secretary of

Defense (OSD/ASD(R&E)/RD/IS&CS) for their guidance and support of this research.

REFERENCES

- [1] U.S. Department of Defense Specification for Integrated Circuits (Microcircuits) Manufacturing, "Available: <http://www.landandmaritime.dla.mil/Downloads/MilSpec/Docs/MIL-PRF-38535/prf38535.pdf>," 2007, MIL-PRF-38535.
- [2] F. McCluskey, E. Hakim, J. Fink, A. Fowler, and M. Pecht, "Reliability assessment of electronic components exposed to long-term non-operating conditions," *Components, Packaging, and Manufacturing Technology, Part A, IEEE Transactions on*, vol. 21, no. 2, pp. 352–359, 1998.
- [3] S. Jain and V. Agrawal, "Test generation for MOS circuits using D-algorithm," in *Design Automation, 1983. 20th Conference on*. IEEE, 1983, pp. 64–70.
- [4] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," in *Fault-Tolerant Computing, 1995. Highlights from Twenty-Five Years'., Twenty-Fifth International Symposium on*. IEEE, 1995, p. 337.
- [5] T. Niermann and J. Patel, "HITEC: A test generation package for sequential circuits," in *Design Automation. EDAC., Proceedings of the European Conference on*. IEEE, 1991, pp. 214–218.
- [6] C. Maunder and R. Tulloss, *The test access port and boundary scan architecture*. IEEE Computer Society Press, 1990.
- [7] P. Hortensius, R. Mcleod, W. Pries, D. Miller, and H. Card, "Cellular automata-based pseudorandom number generators for built-in self-test," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 8, no. 8, pp. 842–859, 1989.
- [8] N. Touba, "Survey of test vector compression techniques," *Design & Test of Computers, IEEE*, vol. 23, no. 4, pp. 294–303, 2006.
- [9] D. Baik and K. Saluja, "Progressive random access scan: a simultaneous solution to test power, test data volume and test time," in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*. IEEE, 2005, pp. 10–pp.
- [10] J. Rajski, J. Tyszer, G. Mrugalski, W. Cheng, N. Mukherjee, and M. Kassab, "X-press compactor for 1000x reduction of test data," in *Test Conference, 2006. ITC'06. IEEE International*. IEEE, 2006, pp. 1–10.
- [11] L. Wang, C. Wu, and X. Wen, *VLSI test principles and architectures: design for testability*. Morgan Kaufmann, 2006.
- [12] S. Mitra and K. Kim, "X-Compact: an efficient response compaction technique," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 421–432, 2004.
- [13] E. McCluskey, D. Burek, B. Koenemann, S. Mitra, J. Patel, J. Rajski, and J. Waicukauski, "Test data compression," *Design & Test of Computers, IEEE*, vol. 20, no. 2, pp. 76–87, 2003.
- [14] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, vol. 3. Citeseer, 2006, pp. 1433–1452.
- [15] E. Candès and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [16] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [17] D. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- [18] E. J. Candès, J. K., Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [19] K. Fukunaga, "Introduction to statistical pattern recognition." Academic Press, 1990.
- [20] M. Hansen, H. Yalcin, and J. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *Design & Test of Computers, IEEE*, vol. 16, no. 3, pp. 72–80, 1999.
- [21] M. Hutton, J. Grossman, J. Rose, and D. Corneil, "Characterization and parameterized random generation of digital circuits," in *Proceedings of the 33rd annual Design Automation Conference*. ACM, 1996, pp. 94–99.