

Hybrid Architecture for Embedded Test Compression to Process Rejected Test Patterns

Sebastian Huhn*[†]

Daniel Tille[§]

Rolf Drechsler*[†]

*University of Bremen, Germany
{huhn,drechsle}
@informatik.uni-bremen.de

[§]Infineon Technologies AG
85579 Neubiberg, Germany
Daniel.Tille@infineon.com

[†]Cyber-Physical Systems
DFKI GmbH
28359 Bremen, Germany

Abstract—This work presents a novel hybrid compression architecture that seamlessly combines the advantages of an embedded test compression technique with a lightweight codeword-based compression scheme. The proposed architecture tackles the shortcomings of state-of-the-art techniques, which are widely used to address the rising challenges of safety-critical applications enforcing a zero defect policy. Embedded test compression techniques had been introduced that allow the compression of a large share of the test patterns. However, depending on the test application (e.g. low pin count test) there is a certain number of test patterns, which are incompressible due to the architecture and will be rejected. This leads to a test coverage decrease which, in turn, jeopardizes the zero defect policy. Therefore, the rejected test patterns are typically transferred in an uncompressed way bypassing the embedded compression, which is extremely costly. The proposed hybrid architecture mitigates the adverse impact of rejected test patterns on the compression ratio as well as on the test application time of state-of-the-art techniques. The experimental evaluation of industrial-sized designs clearly shows that a significant compression ratio up to 67.4% and a test application time reduction up to 65.7% can be achieved.

I. INTRODUCTION

The latest accomplishments in the field of design and manufacturing of *Integrated Circuits* (ICs) enables completely new fields of application. For instance, the newest generation of *Electronic Control Units* (ECUs) integrates a huge number of sophisticated on-board ICs to implement advanced driver-assistance systems. Potential defects occurring during the manufacturing process have to be reliably detected to be compliant with these challenging requirements [1]. Thus, a very high test coverage is required leading to large sets of test patterns.

Complex designs utilize powerful embedded test compression techniques like [2] to cope with this high test data volume. These techniques aim at reducing the overall testing time and, thus, saving costs. Typically, dedicated hardware is embedded on-chip that allows an on-the-fly decompression of the (compressed) test data during the transfer between the automatic test equipment and the circuit-under-test. This works fine for many of the test patterns, however, the remainder of the test patterns cannot be compressed at all; they are rejected due to the introduced compression architecture. These *Rejected Test Patterns* (RTPs) are then transferred to the chip in a sequential and completely uncompressed fashion, i.e., bypassing the compression architecture. This yields to an adverse impact on the overall compression ratio as well as to a significant test cost increase.

This work proposes a novel hybrid architecture, which seamlessly combines a state-of-the-art embedded test compression technique with a codeword-based compression scheme replacing the costly bypass structure. The codeword-based approach is

meant to be applied for the RTPs to tackle the shortcomings of existing techniques. More precisely, the adverse impact of RTPs on the overall compression ratio is significantly reduced. Furthermore, the introduced hardware overhead is negligible and a powerful retargeting approach has been implemented on the basis of state-of-the-art formal techniques as proposed in work [3]. First experiments have already shown that a compression ratio up to 67.4% for the RTPs can be achieved and the test application time can be reduced by up to 65.7%.

II. HYBRID COMPRESSION ARCHITECTURE

The scheme of the proposed hybrid compression architecture is shown in Figure 1 and focuses on the incoming data of RTPs, which have to be retargeted once prior to the test application. The retargeted patterns, i.e., a sequence of codewords, are transferred bit-wise to the circuit. When a codeword is completed, the decompressor expands it to the (original) dataword and temporarily stores it until enough data are available to feed every input of the scan chains simultaneously. This is a significant improvement over the regular bypass, since it can feed the test data into the scan chains in parallel and not serially as the regular bypass does. Three different modules are required to implement the proposed hybrid architecture, as described in the following paragraphs.

The **Codeword-based Decompressor** implements the codeword-based technique utilizing an embedded dictionary to decompress the RTPs on-chip without any loss of information. Reconsider that these RTPs have been compressed once by the retargeting procedure prior to the test, i.e., the compressed data consists of a sequence of codewords. Methods to determine efficient codewords for given test data can be found in [4]. One integral component is the dictionary, which holds the codewords c_i in conjunction with the associated (uncompressed) dataword u_i . Every entry consists of one binary-encoded, unique codeword c_i (with a length of 1 to 3 bit) and an associated (binary) dataword u_i . This dataword u_i tends to have a greater length $|u_i|$ since this enables the data volume reduction, i.e., the

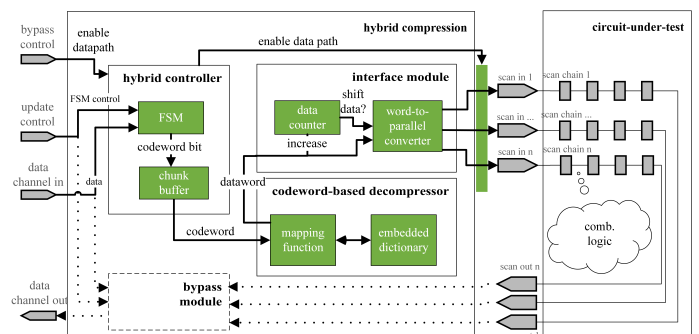


Figure 1: Hybrid Compression Architecture

TABLE I: Benchmark results

circuit name	#scan. element	#scan-chains	#pattern		#blocks		retargeting run-time [s]		pattern compression ratio [%]			test time red. [%]		
			min.	avg.	max.	min.	avg.	max.	min.	avg.	max.	max.		
ethernet	10,038	5	1,049	2	13.2	26.4	47.2	31.1	56.4	79.3	34.6	55.5	72.8	
vga_lcd	12,983	14	1,286	2	13.6	37.6	42.1	30.6	40.7	56.1	30.5	41.5	47.9	
netcard	96,569	97	9,939	12	104.4	126.0	230.4	28.5	38.1	67.4	29.6	45.7	65.7	

compression. Besides this embedded dictionary, this module also implements the mapping function $\psi(c_i) \rightarrow u_i$, which provides the functionality for the later decompression. By design, it is ensured that the datapath remains completely isolated until the bypass control signal is set. This principle allows avoiding any unintended interference when the regular compression infrastructure decompresses the unrejected test patterns.

The **Hybrid Controller** realizes the necessary control structures by introducing a *Finite-State-Maschine* (FSM), whose state transitions are controlled by the external compression control signal and synchronized with the test clock. The FSM design allows differentiating between data and instruction branch, which allows a clear distinction between data- and control-path. Besides this, the FSM implements four different instructions, e.g., the (de-)activation of the hybrid compression scheme. It also realizes the bitwise transfer of the compressed RTPs, whose chunks are stored in the internal *chunk buffer*.

The **Interface Module** realizes the junction between the introduced codeword-based decompressor and the N inputs of the scan-chains. This means that this module acts as the data source by decompressing newly received codewords to the associated datawords using Ψ and the inputs of the scan-chains act as the data sink. Finally, this module keeps track of the actual number of available data bits to decide whether enough data bits are available to execute a parallel shift to all inputs of the scan-chains. If enough data bits are available, the shift operation is performed or, otherwise, more data bits are received until the required amount of data is available.

III. EXPERIMENTAL EVALUATION

Different industrial-representative *OpenCores* as well as *Gaisler Research Benchmark* circuits from the *IWLS 2005* benchmark collection have been used for the experimental evaluation. The underlying scan capabilities, as well as the embedded test compression, has been inserted by a commercial tool. As indicated earlier, the number of RTPs might be significant. However, it depends on the test application and also may differ between different synthesis runs. In order to have reproducible results, we, therefore, consider in the following all test patterns for the experimental evaluation. The proposed codeword-based compression module is solely implemented in Verilog and the later application is validated by simulation. The test patterns, which are generated by a commercial tool, are then processed by the retargeting framework of work [3], [4] introducing a partitioning scheme, which leads to a block-wise processing. The retargeting is executed on an *Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz* with *32 GB* system memory within a *C++* compiler-environment (*gcc-Version 8.2.1*).

Table I presents the characteristics of the benchmark circuit as well as detailed results of the conducted experiments. More precisely, the benchmark circuit name, the overall number of scannable element, scan-chains and test patterns, the number of required data blocks per RTP, the minimal, average and maximal retargeting run-time per pattern in seconds is shown.

Furthermore, Table I also presents the minimal, average and maximal pattern compression ratio in percent and, finally, the further achieved test time reduction in percent - both compared against the regular bypass transfer without any compression.

In case of the *netcard* benchmark, the experiments clearly show that the run-time of the retargeting engine, while considering different design sizes, is stable **per** block. The achieved compression ratio is stable over the conducted experiments as well, which is indicated by a standard deviation of approx. 6.4% regarding the compression percentage (*netcard*). In contrast to this, the deviation between the minimum and the maximum of achieved test time reduction is greater, which is due to the fact that this depends on the distribution of codewords with a length of 1, 2 or 3, respectively. When processing the largest *netcard* circuit with the greatest number of patterns ($N = 9939$), the resulting test data volume (application time) can be significantly compressed by 38.1% (45.7%) on average.

IV. CONCLUSIONS

This paper presented a novel hybrid architecture for embedded test compression, which allows the processing of rejected test patterns. So far, these patterns were incompressible by state-of-the-art embedded test compression techniques and, thus, were rejected. This circumstance had an adverse impact on the overall compression effectiveness and, hence, untapped potential of reducing the test costs remained.

The proposed hybrid architecture seamlessly combines a state-of-the-art embedded compression technique with a lightweight codeword-based compression scheme. To the end, only a negligible hardware overhead in sense of register count is introduced to the design and the computational effort for the preprocessing step is manageable. A significant test data volume reduction of up to 67.4% was achieved for industrial-representative designs – with up to approx. 100k flip-flops – without any loss in test coverage at all. Furthermore, the test application time was reduced even stronger by up to 65.7%.

V. ACKNOWLEDGMENT

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 276397488 – SFB 1232 in subproject P01 ‘Predictive function’.

REFERENCES

- [1] M. Weber and J. Weisbrod, “Requirements engineering in automotive development-experiences and challenges,” in *IEEE Joint International Conference on Requirements Engineering*, 2002, pp. 331–340.
- [2] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, “Embedded deterministic test,” *IEEE Transaction on CAD of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 776–792, 2004.
- [3] S. Huhn, S. Eggersglüß, and R. Drechsler, “Reconfigurable TAP controllers with embedded compression for large test data volume,” in *IEEE International Symp. on Defect and Fault Tolerance in VLSI Systems*, 2017, pp. 1–6.
- [4] S. Huhn, S. Eggersglüß, K. Chakrabarty, and R. Drechsler, “Optimization of retargeting for IEEE 1149.1 TAP controllers with embedded compression,” in *Design, Automation and Test in Europe*, 2017, pp. 578–583.