J Electron Test (2016) 32:245–255
DOI 10.1007/s10836-016-5588-y

Optimization of Boundary Scan Tests Using FPGA-Based Efficient Scan Architectures

Igor Aleksejev[1] · **Sergei Devadze**[1] · **Artur Jutman**[2] · **Konstantin Shibin**[2]

Received: 31 July 2015 / Accepted: 21 April 2016 / Published online: 28 April 2016

Abstract This paper presents a method for optimization of board-level scan test with the help of reconfigurable scan-chains (RSCs) implemented in a programmable logic of FPGA. Despite that the RSC concept is a well-known solution for scan-based test time reduction, the usage of RSC may lead to un-acceptable hardware overhead. In our work, we are targeting a completely new approach of exploiting on-board FPGA resources that being unconfigured are typically available during the manufacturing test phase for carrying out tests using temporarily implemented virtual RSC structures. As the allocated FPGA logic is re-claimed for functional use after the test is finished, the presented method delivers all the advantages of RSCs at no extra hardware cost. Experimental results show that the proposed virtual RSCs can fit into all available commercial FPGAs providing a significant test time reduction in comparison with state-of-the-art Boundary Scan test tecnique.

Keywords Boundary Scan · Reconfigurable Scan-Chain · Embedded Instrumentation · FPGA

1 Introduction

During the last decade, systems under test (SUTs) have undergone significant changes. With a year-to-year shrinking of technology and growing of complexity, system units become smaller while the density of these units on a board is increasing. Contemporary high-end printed circuit board assemblies (PCBAs) are so densely populated that there is practically no free surface for test points. At the same time, as transistors become smaller, integrated circuits may now accommodate more functionality inside one chip. For example, contemporary single system-on-a-chip (SoC) integrated circuit (IC) holds the same complexity as a ten years old system, which consisted of several boards and multitude of different ICs. The usage of complex SoC devices on a board require a vast part of interconnections be moved into internal PCB layers thus cannot be accessed from the surface anymore. Nowadays, PCBAs usually have more than twenty layers instead of several like before. Even though in some cases there is a possibility to get an access to a signal, the physical connection to this signal may disrupt signal integrity, thus the signal will be measured incorrectly.

The lack of access to board interconnection lines and components was addressed by IEEE 1149.1 Boundary Scan (BS) standard [10] in 1991. Today, major electronic companies approved it as a de-facto standard for structural faults testing and also successfully use it for testing interconnection lines to devices without BS structures inside (including RAM test, FLASH test and programming, etc) [16]. A test is performed by driving and measuring deterministic values through the Boundary-scan shift data register (BSR). Despite of wide adoption of BS among test engineers, 86 % of them frequently encounter problems when testing SRAM/DRAM or Flash memories [7]. The main reason for this is an essentially low-speed nature of the standardized Boundary Scan architecture, which leads to long test application times - the Achilles heel of all scan-based test techniques. The timing issues become especially critical when testing high-end boards at the production phase. For example, In-System Programming (ISP) task where large amounts of data need to be transported over the

Responsible Editor: L. M. Bolzani Pöhls

✉ Igor Aleksejev
igor@pld.ttu.ee

[1] Tallinn Univ. of Technology, Tallinn, Estonia

[2] Testonica Lab OÜ, Tallinn, Estonia

low-throughput JTAG port to be programmed into e.g. a Flash memory can be very time consuming. With modern industrial examples, this procedure typically takes tens of minutes, but it might require several hours in some specific cases (e.g. large flash memories with serial interface).

In this paper, we present a novel approach that can optimize the time required to execute existing Boundary Scan tests. The proposed architecture is based on the methodology of re-using the on-board FPGA to incorporate embedded instruments (EIs) dedicated for test [1]. The methodology implies that the FPGA is a centric device on a board or a system under test and has many connections to the peripheral components. In normal conditions, this FPGA is a part of functional design. It performs a control over the system and its operations. During the manufacturing test phase, when it is not yet configured, it can be specifically programmed with dedicated FPGA design (embedded instrument) that contains several instrument IPs capable to perform the series of tests. In this work, we propose the architecture of virtual reconfigurable scan-chains (VRSCs) implemented in FPGA programmable logic. The term *virtual* emphasizes that in our concept we do not modify the underlying hardware, thus does not require extra resources for test optimization. In consequence, the proposed instruments can be instantly used with all available commercial FPGAs.

This paper is organized as follows. The next section describes the technical details of JTAG standard needed to understand the root cause of the BS problems and details of proposed reconfigurable scan chains. After that, an analysis of related works directed to overcome BS problems is presented. In Section 3, the architecture of proposed reconfigurable scan chains is described. Section 4 presents the most promising configurations of RSCs and studies their properties. Section 5 evaluates the proposed configurations based on typical manufacturing tasks. The section ends with the experimental results of using selected RSC configuration for real life industrial applications. Section 6 draws the conclusions of the paper.

2 Background and Related Works

The IEEE 1149.1 standard defines test access port and hardware architecture of the BS test access and test application mechanisms. The standard assumes that there is an external computer (tester) that takes control over the system under test using on-board BS-enabled devices (see Fig. 1). In order to communicate with target device under test the external tester (JTAG controller) sequentially shifts bits (1 bit per test clock cycle) to BSR using test data input (TDI). The test data that should be sent to DUT will be propagated to the corresponded cells of BSR with each test clock cycle. After the propagation, the data exchange between DUT and BSR is triggered: the test data (stimuli) from BSR are applied to DUT and the responses of DUT are captured back into BSR. Next, the responses from DUT is shifted out from TDO in a similar manner: the bits in BSR are shifted along the scan-chain (1 bit per clock cycle) until all the captured values reach test data output (TDO).

The speed of BS test can be characterized by test access time - the time needed to transport test vector from a tester to the test data register in order to apply it to DUT. The test access time mainly depends on the test operating frequency (TCK) and the length of the test data register. In addition, for each test vector transfer we have to add a software delay that is produced by tester runtime and delays in a BS controller hardware operation during test execution. Consider a BS chain C that consists of BS devices $c_i \in C$, $i = 1,\ldots, n$. Each BS device c_i has an individual BS register with length of L_i. Total length of the chain C can be viewed as a sum:

$$C_{\text{total}} = \sum_{i=1}^{n} L_i \tag{1}$$

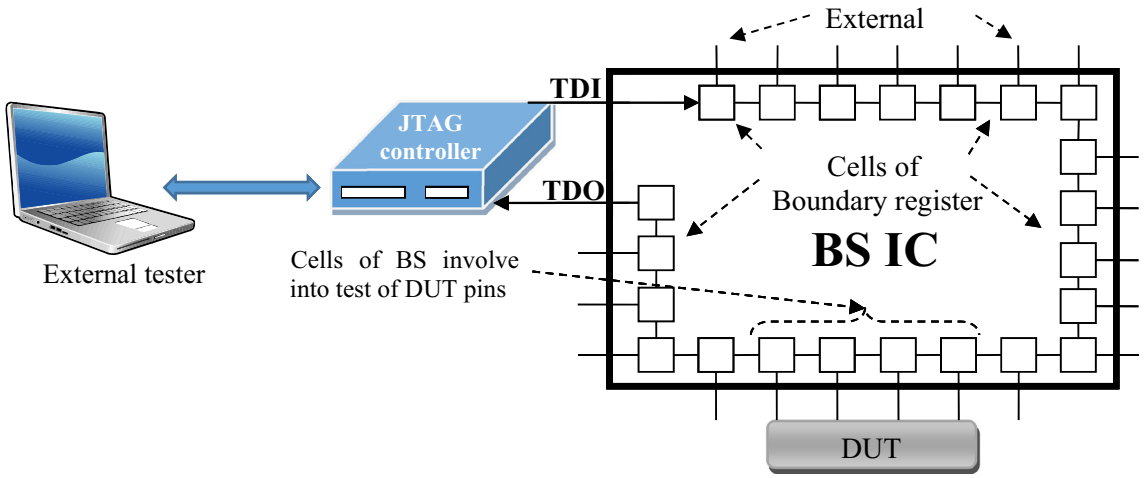

Fig. 1 Usage of boundary-scan register (BSR) for testing

Let F_{tck} be the test clock frequency, then the test access time T_{acc} needed to shift one test vector into BS chain is equals to:

$$T_{acc} = \frac{C_{\text{total}}}{F_{tck}} + \delta(s) + \delta(h) + 6 \cdot \frac{1}{F_{tck}} \quad (2)$$

where $\delta(s)$ represents the software delay and $\delta(h)$ - controller's hardware delay correspondingly. Also, we need extra 6 TCKs to make transition on test access port (TAP) state machine (i.e. from "Shift-DR" TAP state to "Update-DR" and then back to "Shift-DR).

The typical test program consists of applying N vectors to DUT, hence the test program length can be approximated by the number of *DRshifts* (transfer of test data in "Shift-DR" state followed by "Update-DR"). In order to apply one test vector to DUT the full shift of BS register should be performed. If we need to apply N test vectors to DUT, then we have to perform $N+1$ DRshifts (an extra DRshift is needed to readback the result of the application of the last test pattern). As a result, the overall time for a test program can be calculated as follows

$$T_{\text{program}} = \sum_{i=1}^{N+1} T_{acc} \quad (3)$$

Boundary Scan can also be used to test combinational and sequential logic clusters, SRAM/DRAM memory interconnect testing, Flash and I2C memories programming, testing LEDs, clocks, connectors, etc. We can estimate the overall time needed to test a board with BS as

$$T_{\text{board}} = \sum_{i=1}^{K} T_{\text{program}} \quad (4)$$

where K is a number of all BS tests that need to be carried for this board. T_{board} becomes important if at least one long $T_{program}$ exists or when K is big enough. Although maximally acceptable T_{board} and $T_{session}$ may vary, typically the reasonable test time should not exceed a couple of minutes. However, in practice the short test time is dominant only for small PCBAs with a few digital ICs. The time needed to test complex boards with many DDR and Flash memory chips usually exceeds the acceptable time frames.

The key shortcoming of BS architecture is the fixed structure of Boundary register. There is no means that could reduce the amount of clock cycles needed to perform test data exchange with a device under test; therefore a full shift of BSR is always required. Indeed, only a small portion of BSR cells (and corresponded pins of device) is typically connected to the DUT (see Fig. 1). The remaining cells/pins are either connected to other devices on-board or left unused. The above fact motivates the study of the prior work in the field of scan chain optimization - both in chip (IC) and board-level testing. There are two fundamental approaches that have been proposed by researchers. In [3, 17], authors partition one long chain into multiple small scan chains (SCs). Each scan chain surrounds particular part of the IC and one or several such SCs can be bypassed during the test if they are not used. A similar idea, but with a finer granularity was proposed in [14, 18, 23]. Here, authors surround each or particular scan cells with multiplexing blocks. Depending on the control value the cell is either included to the scan chain or bypassed. This architecture is known as a partial or dynamic scan. The application of the latter approach to Boundary Scan problem have been studied by several researchers [9, 22], where authors proposed to modify the BS architecture by adding multiplexing blocks to each cell. These solutions are very effective in terms of test time optimization, but involve a large hardware overhead.

The second group of works proposes methods that change IC internal structure, i.e. a change of IC Design-for-Testability (DfT). In [4], authors targeted a problem of Boundary Scan inability to reliably test interconnection lines between BS IC and high-speed devices, like DDR SDRAM or NAND Flashes. In their study, they proposed a standardized test access methodology for memory devices. The concept of the method consists of adding test logic to memory IC that takes a control of IC inputs/outputs. In test mode the memory cells are bypassed, while test logic is used to link input and output signals. ASIC developers, like Intel [15] or Cisco [20], propose to use BIST structures in order to test interconnection lines between two chips or boards. Both technologies, Intel's IBIST and Cisco's IOBIST, are targeted to test high-speed interconnection lines addressing both static and dynamic faults. However, this method is suitable only for ASICs where particular I/O pins have a predefined location. Moreover, both ICs have to have a corresponding BIST structure behind these pins to make the test possible. As a result, only big companies that develop both ICs and boards in-house may get a serious benefit from the usage of interconnect BIST.

The third group suggests re-using on-board resources for test purposes. This methodology requires only minor modifications within typical test setup. The benefit of these solutions is relatively low additional costs compared to the previously described proposals. The proposed works are reusing either FPGA [5, 6] or a microprocessor [21] located on a board. In [6], authors proposed to utilize FPGA reconfigured solely for the application of test vectors. An FPGA is providing a simple access to the internal signals or it can be used for applying burst of patterns to the DUT. The method implies that a PCBA is designed with this strategy in mind, which could be seen as a main drawback of the method. The development of a test code is supposed to be done manually by a group of test engineers. In [5], authors proposed an FPGA-based test system designed to improve BS test coverage. The test system represents a multiprocessor architecture that is divided into 5 layers in order to reduce the complexity of the FPGA design.

The architecture can be automatically generated depending on the particular task and DUT model. The implemented processor is controlled through the debug port by external PC. However, this solution requires to be generated and compiled for every new product or test case. In [2, 12], alternative variants of product-specific embedded instruments from the industrial companies are proposed. In [12], authors use embedded instruments for programming and test of in-system memory devices, while in [2] for high-speed memory testing. In both cases a specialized IPs are generated and compiled depending on the used FPGA and target devices under test located on a particular board. To sum up, all the previously proposed FPGA-based instruments are product-specific i.e. designed to support a particular test case and DUT to FPGA relative location.

3 Virtual Reconfigurable Scan-Chains

In this work, we propose a virtual reconfigurable scan-chains architecture that is capable for run-time reconfiguration of shift register for the sake of decreasing the length of active scan-path thus reducing the time needed for sending/receiving test data to/from DUT. The proposed architecture assumes its implementation in FPGA programmable logic thus does not require neither hardware change nor hardware overhead. The test clock speed (TCK) typically does not exceed 20–30 MHz, which is affordable for FPGA logic, so there is no problem regarding the performance. Unlike state-of-the-art embedded instruments designed for specific board – test pair, the key distinguishing feature is that the presented FPGA-based VRSC instrument is universal and independent of particular FPGA and device under test. The proposed VRSCs are designed in such a way that allow them to be automatically adapted inside FPGA to meet the requirements of any particular test case. On the output, the solution represents ready-to-use test instrument that can optimize any Boundary Scan test. The instrument is suitable for all test environments and does not require FPGA design recompilation for each new product or DUT change.

As it was previously stated, only a part of cells in the scan register is actually used to perform a test. *Dynamic* part gets new values with each of test patterns while other cells (*static* part, *static* values) are typically not changing during the whole test program. Obviously, it is possible to identify dynamic and static parts of the scan chain (BSR) before starting or even generating the actual test. The proposed VRSC approach uses this information in order to speed-up test access time. The presented architecture is based on the usage of special multiplexing blocks that can either bypass scan cells of the shift register or keep these cells serially included into the active scan-path (see Fig. 2). The bypassing scheme consists of a data cell, adjacent multiplexor and a mask cell. The data cell can functionally correspond to one of the IEEE 1149.1 BC_*X* [10] cell types in order to be compatible with the JTAG standard. The special mask signal is used to control the function (*bypass* or *include*) of multiplexing block. This signal is in its turn controlled by a memory element (mask cell). When the mask cell holds logic '1', then the data cell is included into the active scan-path. If the value is logic '0', then the cell is bypassed and the output of the previous cell becomes directly connected to the multiplexing block of the cell located just after the bypassed one. Several consecutive multiplexing blocks can be configured for *bypass* in order to skip (remove from the active scan-path) multiple adjacent data cells. It is worth to emphasize that even if a cell is bypassed it can still drive or measure the value on device pin. Hence, bypassed cell can be used as a *static* cell (for static part of test data), while a cell included into active scan-path (by configuration of mask register) is used as a *dynamic* cell (for dynamic part).

In order to support all possible arrangement of DUT-to-FPGA pin relative locations, there is a multiplexing block behind each I/O pin of the FPGA (see Fig. 3). As a result, by means of dynamic bypassing of unused scan cells, VRSC is capable to change the length of active scan-path depending on the number and location of cells/pins of FPGA device that are involved in test of DUT. The corresponded mask cells are connected into a separate shift register. This mask register allows run-time reconfiguration of the bypassing scheme of the VRSC without changing the underlying architecture. The mask register and data register are controlled by an external tester (instrument controller) using standard JTAG/IEEE1149.1 DfT structures available in FPGAs. This implies that the transportation of test data to/from embedded instrument is done by means of typical JTAG IR−/DR-shifts. This

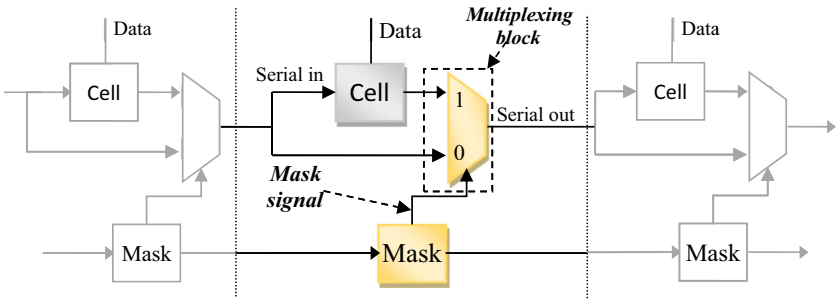

Fig. 2 Example of bypassing scheme in VLSR

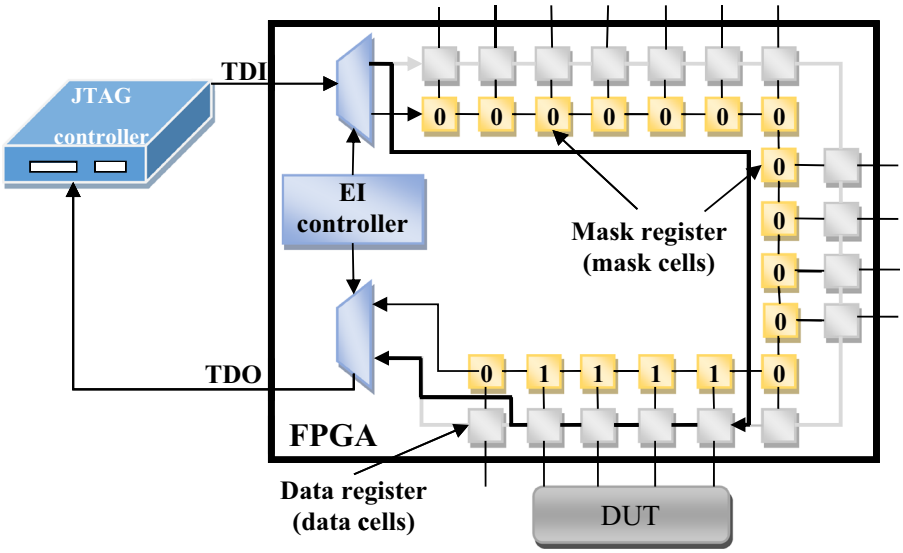

Fig. 3 Scan chain with bypassed cells

significantly reduces the costs of using Embedded Instrumentation approach since the existing BS hardware can be re-used. The Instrument controller inside FPGA is mainly used to manage currently selected shift register: depending on the loaded internal instruction the test data supplied to JTAG TDI pin is propagated into either mask register or VRSC active scan-path.

The opportunity to reconfigure VRSC instrument on-the-fly distinguishes our approach from the previously proposed methods. This feature makes VRSC being universal, i.e. independent of a particular DUT as well as of the location of DUT relatively to FPGA pins. The proposed method is able to reconfigure VRSC to form the *dynamic* (active) scan-path for DUT connected to any subset of I/O pins of FPGA device. As a result, the presented VRSC instrument can be instantly used for every SUT and does not need to be recompiled for a new product or after a product change.

To use the optimizing test architecture, an already developed Boundary Scan test program is modified only in the beginning where a *dynamic* part of VRSC is configured by modification of mask register. After that, the register with data and control cells is selected and the test is done according to the JTAG standard. The example of VRSC-based test program workflow is given below:

1. At first, one has to configure the *static* cells with the values that should be driven throughout the whole test program. In order to preload *static* values to VRSC all mask register values should hold logic '1'. In this case, all data cells are active in the chain and none is bypassed. Static data is then shifted into the data cells.
2. The second step is to form the *dynamic* part of VRSC by modifying the mask register. In Fig. 3, only four pins are connected to DUT, thus only these matching mask cells should be loaded with logic '1', while others should contain value '0'. Then, all the data cells will be bypassed except the four, which will be included into the active scan-path.
3. Perform a test with the shortened scan-chain. The test time will be drastically reduced.
4. For other DUT connected to other pins of FPGA, the procedure is repeated starting from the step 1.

The proposed VRSC instrument can speed-up all the Boundary Scan tests. The key benefit of VRSC is the reconfigurability: dynamic and static parts of its scan-chain can be altered in run-time without reprogramming of the FPGA. Thus, as an extra benefit, the relatively long FPGA configuration procedure is performed just once for all accelerated VRSC-based BS tests. Overall, the VRSC usage is justified when the following formula is true (K is the total number of BS tests):

$$\sum_{i=1}^{K} T_{BS\_test_i} > \sum_{i=1}^{K} T_{\mathrm{VRSC}\_test_i} + T_{\mathrm{FPGA\_config\_time}} \quad (5)$$

4 VRSC Configurations

The VRSC architecture can be implemented in multiple different configurations. Each such a configuration is suitable to particular test tasks. In this section, an analysis of most promising VLSR configurations as well as their cost are considered. After that, the proposed configurations are evaluated based on the test access time critical tasks.

Let us define the following five types of cells: the *output* cells we denote by *O*, the *control* cells by *C*, *the input* cells by *I*, the *bidirectional* cells by *I*/*O* and the *mask* cells by *M*. Based on these definitions several examples of VRSC configurations

are shown in Fig. 4. Before starting to compare them, it is worth to notice that in order to perform the test of DUT an emulation of functional protocol is used, which means that a certain sequence of signals should appear on operational inputs (e.g. Address, Data, Control) of the DUT imitating the protocol. To do this, the sequence of *DRshifts* is applied. According to their functionality, these *DRshifts* can be characterized as following operations:

- "Send test pattern out" (SPO) operation is used to send one test pattern from the VRSC to a DUT.
- "Change the direction of bidirectional Data lines" (CD) operation selects whether Data lines are driven by the VRSC or DUT.
- "Read value on Data lines" (RV) operation is used to capture the data sent by DUT on Data lines.
- "Expect value on Data lines" (EV) operation is similar to the RV operation. The difference is in the ability to continue a test when we only "Read value". In "Expect value" mode, we are unable to continue the test until the expected data appear on the Data lines.

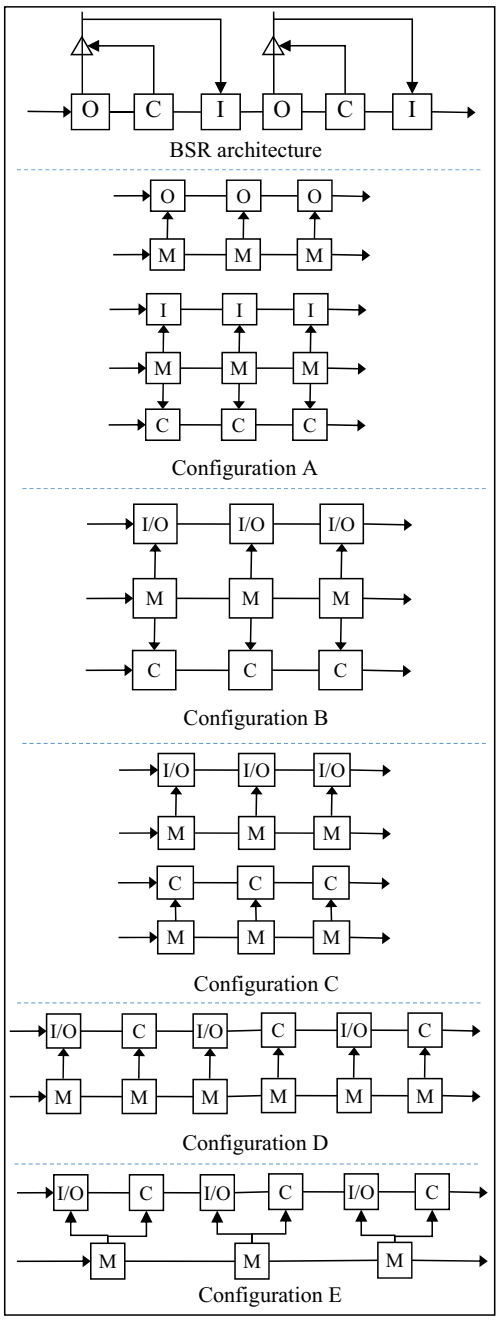

Fig. 4 VRSC proposed configurations

For each operation, different types of *I*, *O*, *C*, *I/O* cells are active. Also for each test task, the amount of applied operations as well as their diversity throughout the test are unique. Although typically SPO operations are dominating, CD, RV and EV operations may also comprise a significant part of test. The EV operation is usually executed separately; therefore it cannot be combined with other operations. SPO, CD and RV operations can be performed simultaneously (during one *DRshift*) if required cells are situated in the same scan chain. When dynamic cells are located in separate shift registers, a register switch operation to change the active scan register has to be performed.

Since VRSC architecture assumes implementation in FPGA programmable logic, it has one additional cost metric – the hardware cost of architecture. *Output*, *control*, *bidirectional* and *mask* types of cell consist of shift and update flip-flops (FFs). *Input* cell has only a shift flip-flop. A hardware cost of VRSC architecture can be approximated as a number of FFs used in the architecture. However, the hardware cost is not considered as an exceptional characteristic for evaluation. Since we are re-using the on-board FPGA only temporarily during manufacturing test phase, we are not constrained in FPGA resources. For us it is important whether a configuration will fit into FPGA devices or not. In case a configuration cannot be mapped to FPGA, a configuration with lower hardware cost has to be used.

As the reference, the representation of typical *BSR* architecture using the defined cell types is shown in the beginning of Fig. 4. Traditionally, the Boundary register is comprised of *input*, *output* and *control* cells per each pin of the device. It is worth to point out that even after FPGA configuration, a post-configuration BSDL re-defines only the input/output capabilities of cells of IEEE 1149.1 Boundary register. The length of the Boundary register, which is used for Boundary Scan test, is fixed and remained unchanged. Therefore, the modification of BSDL will not change the structure of Boundary Scan test. Let us denote the length of the register by *L*. As there is no possibility to bypass inactive cells in BS architecture, the cost of "SPO" and "EV" operations for Boundary Scan equals to *L* clocks.

In the first proposed configuration *A* we separate each type of cells to isolated chains. Since *control* and *input* cells are

often used for the same pins, it is rational to use shared mask register for these cell types. This configuration is mostly efficient when only single type of operation is vastly prevalent. Here we can include/bypass the active cells in the most efficient way, because we have a full control over each cell. When the operations are often changing, different types of cells are utilized to perform the test. In this case, the configuration becomes inefficient due to costly register switches. Let us denote the number of FPGA pins as X, then the hardware cost of the configuration A equals to $9*X$ FFs.

In the configuration B *input* and *output* cells are combined into *bidirectional* type of cells. The mask register is shared for both scan-chains. In this configuration a considerable amount of hardware resources is saved (the hardware cost is only $6*X$ FFs). On the other hand, an independent control over every cell is lost. Since the scan register with control cells is separated, this configuration is optimal when the direction of data is stable and changes rarely. In this case, the scan-chain with *control* cells should be loaded only once (or a few times) during the test. Otherwise, again costly register switches are needed when the direction of bidirectional lines changes frequently. The benefit of using *I*/*O*-cell instead of separate *I*-cell and *O*-cell is in the ability to capture the values to *input* cells simultaneously with shifting out new test data from *output* cells.

In the configuration C we separate mask registers for *I*/*O* cell types and *control* cells. By using this configuration one is able to select which *control* cells are needed for switching the direction of the corresponded Data cells and which *control* cells can be bypassed (typically only cells behind Data lines (C_D) change its direction). Due to the separated mask registers, the hardware cost increases and is equal to $8*X$ FFs. Comparing to the configuration B only C_D are used for the change of direction. Since the register with *control* cells is separated, two register switches are required to address these cells.

In the configuration D all *bidirectional* and corresponded *control* cells are combined and placed into the single scan-chain. The hardware cost remains the same as for configuration C and equals to 8*X FFs. The benefit over previous architectures is that we do not have to switch between shift registers during test operations: all the data is shifted to the cells within one DRshift. Consequently, this configuration is propitious when the SPO, CD and RV operations are frequently used. The shortcoming of the configuration is the redundant data that we have to transport to control cells even when the direction is not changing.

The configuration E represents the combination of architectures B and D. Since for some application many cells of I/O type (that are capable to change the direction) are often used, it is beneficial to share single mask-cell for each pair of *I*/*O* cell and *control* cell. The advantages and disadvantages of this configuration are also combined from the configuration B and D. We save on hardware cost ($6*X$ FFs), but we do not have a full control over each cell. Although only one shift is required to send new values to active cells, not changing *control* cell values include a large overhead to every DRshift.

5 VRSC Configurations Evaluation

In order to compare the proposed VRSC configurations we have to analyze the costs of real test tasks execution. In-System programming (ISP) of NOR, NAND, SPI Flash memories, DDR memory static interconnection test and video/audio interface tests are the tasks where the overall test time reduction becomes the main optimization criterion. From this list, ISP is the most liable case because the time needed to perform the programming sometimes exceeds hours in normal circumstances. Within this work, we select the following tasks as a basis for proposed architectures evaluation:

- NOR Flash ISP in “Unlock_bypass” mode
- NOR Flash ISP in “Buffered” mode
- NOR Flash data verify
- SPI Flash data programming
- SPI Flash data verify

These types of Flash programming are the most widely used during manufacturing test phase. Usually, one programming cycle is required to write 32 bytes and 256 bytes of data into the NOR and SPI Flash correspondingly. These cycles are repeated for hundreds of thousands times to program the required amount of data. The structures of one such cycle in terms of pre-described operations are shown in Table 1. As we can see, the SPO operation is dominant for Flash programming (especially for SPI programming). The amount of CD, RV and EV operations noticeably varies depending on a particular action.

Both in BS and VRSC architectures the data exchange is synchronized by test clock (*TCK*). To shift one bit of data to a BS or VRSC cell one test clock cycle is needed. As a result, the cost of one operation is equal to the number of TCKs required to shift in new test data into the dynamic cells. Let us denote the number of SPO operations during the test program as N_{SPO} and the cost of one SPO operation as C_{SPO}, then N_{CD} and C_{CD}, N_{RV} and C_{RV}, N_{EV} and C_{EV} are the numbers and costs of CD, RV and EV operations correspondingly. The total cost of performing a test task with BS or VRSC architecture can be expressed as:

$$C_{\mathrm{VRSC}} = N_{SPO}*C_{SPO} + N_{CD}*C_{CD} + N_{RV}*C_{RV} + N_{EV}*C_{EV} \quad (6)$$

The cost of one operation (C_{shift}) equals to a sum of a number of dynamic cells (X_{total}) inside the chain, software and hardware delays and TAP state transitions. Let us approximate SW and HW delays as 7 $\frac{1}{F_{tck}}$. This number reflects the

Table 1 Structure of selected tasks

DUT	Action	Size	Operation			
			N_{SPO}	N_{CD}	N_{RV}	N_{EV}
S29AL008D	"Unlock_bypass"	32 Bytes	112	32	0	16
NOR Flash I28F256P30	Programming "Buffered" Programming	32 Bytes	46	3	0	2
NOR Flash I28F256P30	Verify	32 Bytes	16	0	16	0
NOR Flash M25P32	Programming	256 Bytes	4219	0	0	1
SPI Flash	Verify	256 Bytes	8192	0	4096	0

delays for a high-performance test system. The reference to this test system was done intentionally to make the experimental results closer to real life and to minimize the influence of ineffective hardware. In this case, C_{shift} is:

$$C_{\text{shift}} = \frac{X_{\text{total}}}{F_{tck}} + 7 \cdot \frac{1}{F_{tck}} + 6 \cdot \frac{1}{F_{tck}} = (X_{\text{total}} + 13) TCK_S \quad (7)$$

The register switch required to address different scan chains costs 95 TCKs and consists of two IRshifts and one DRshift required to configure Instrument controller. It is also worth to point out that *Mask* register configuration to *bypass* static cells is used only once before the start of the test. Since during one test millions of DRshifts are executed, the time needed for the initial configuration is neglected.

The comparison of proposed configurations costs for performing selected tasks is presented in Table 2. For each selected task, we calculate the cost of task execution in TCKs. After that, the ratio between the cost of certain configuration and the cost of configuration that requires minimal number of TCKs is provided. The evaluation results show that all proposed configurations provide a significant speed-up over BS. One can notice that there is no single configuration, which is the best for all the selected tasks. However, the configurations A and E have significantly higher costs among others. The configuration A with separate inputs, outputs and controls is not efficient for real life examples. This solution is the best only for SPI Flash programming where the SPO operation is totally dominating. When different types of cells are in use, this solution is too costly due to the time needed for switches between shift registers. The drawback of configuration E is that in many test tasks *control* cells are rarely used. That is why *control* cells included into dynamic part add a significant overhead to the configuration.

Configurations B, C and D show the minimal time for several tasks. Moreover, the difference in costs of these configurations is not so considerable when compared to configurations A and E. Although, configuration B has 25 % less FFs than its alternatives, as it was stated before this is not a criteria for selection of the best configuration. It is also worth pointing out that configurations are different from software support point of view. Major changes are required if data and control cells are separated to different shift registers (Configurations A, B and C). On the contrary, if data and control cells are

Table 2 Comparison of VRSC configurations costs for selected tasks

DUT	Task		BS[1]	*A*	*B*	*C*	*D*	*E*
NOR Flash [11, 19]	32 bytes Unlock_bypass programming	C_{VRSC} (TCKs)	129,664	16,896	15,200	14,304	**9344**	14,592
		Ratio	13.88	1.81	1.63	1.53	**1.0**	1.56
	32 bytes Buffered programming	C_{VRSC} (TCKs)	48,624	3717	3477	**3393**	3504	5472
		Ratio	14.33	1.10	1.02	**1.0**	1.03	1.61
	32 bytes Verify	C_{VRSC} (TCKs)	16,208	4160	**912**	**912**	**912**	1824
		Ratio	17.77	4.56	**1.0**	**1.0**	**1.0**	2.0
SPI Flash [13]	256 bytes Programming	C_{VRSC} (TCKs)	4,273,847	**67,708**	71,740	71,740	71,740	143,480
		Ratio	63.12	**1.0**	1.06	1.06	1.06	2.12
	256 bytes Verify	C_{VRSC} (TCKs)	8,298,496	966,656	**139,264**	**139,264**	**139,264**	278,528
		Ratio	59.59	6.94	**1.0**	**1.0**	**1.0**	2.0
HW Cost			–	*9X*	*6X*	*8X*	*8X*	*6X*

[1] – the length of BSR is taken as 1000 cells

located in one scan register (Configurations D and E), the architecture can be described in terms of standardized IEEE 1149.1 BC_*X* cells combined into virtual "Boundary" register.

Considering all this information, we selected configuration *D* for the assessment whether it will fit into commercial FPGAs. This configuration showed the best result for NOR Flash "Buffered Programming" where SPO, CD, RV and EV operations are equally spread which is a common situation for JTAG-based tests. Besides that, this configuration can be represented as a short "Boundary" register, thus does not require serious modifications in a software. For this case, the test vector generation algorithm remains the same as the register fully complies with the standard. As a result, the proposed configuration can be easily incorporated to the already used BS test flow.

We have made a full compilation of configuration's physical implementation on all available Xilinx, Altera and Lattice FPGAs. In all cases the compilation was successfully completed which means that such a VRSC instrument can be utilized in any existing commercial FPGA. This assumes that we created and compiled an IP into a bitstream for every FPGA device but not for every package. The FPGA ICs physically are the same for all packages and typically the largest package have a superset of I/Os for a particular FPGA device. Consequently, we mainly need to support only the largest package with some exclusions for specific devices. This significantly reduces the number of bitstreams that need to be compiled. Currently, there are more than 500 different FPGA devices available from FPGA vendors. But, the size of complete library with compressed bitstreams is approximately 250 MB which is adequate to distribute among end users.

6 Experimental Results

Within this work, a VRSC-based embedded instrument with a corresponding software support was developed. The complete

Table 3 Experimental results for VRSC instrument

#	Task	Boundary Scan [8]	VRSC instrument	Ratio/Speed-up
1	FPGA: Xilinx Spartan3 xc3s1000, DUT: S29AL016 NOR Flash			
	Active cells	1187	82	14.5
	Program + Verify 1 MB	827 s	76.3 s	10.8×
2	FPGA: Xilinx Virtex4 xc4vfx12, DUT: TE28F320 NOR Flash			
	Active cells	983	80	12.3
	Program + Verify 1 MB	315.3 s	38.5 s	8.2×
3	FPGA: Xilinx Spartan3a xc3s200a, DUT: K9F2G08 NAND Flash			
	Active cells	637	53	12.0
	Program + Verify 32 MB	4705 s	627 s	7.5×
4	FPGA: Xilinx Spartan3a xc3s50a, DUT: VS1011 SPI Flash			
	Active cells	373	12	31.1
	Write + Read 16 KB	48.5 s	3.6 s	13.5×
5	FPGA: Xilinx Virtex5 xc5vlx110t, DUT: JS28F256P30 NOR Flash			
	Active cells	2517	98	25.7
	Program + Verify 1 MB	355.7 s	33.7 s	10.5×
6	FPGA: Xilinx Virtex5 xc5vlx110t, DUT: M25P32 SPI Flash			
	Active cells	2517	20	125.9
	Program 1 MB	2860 s	55.1 s	51.9×
7	FPGA: Altera Cyclone III ep3c10, DUT: M25P80 SPI Flash			
	Active cells	603	16	38.2
	Program + Verify 64 KB	168.2 s	11.2 s	15.0×
8	FPGA: Lattice XP2 lfxp2_5e, DUT: AT25DF021 SPI flash			
	Active cells	402	19	21.2
	Readback 60 KB	89.9 s	23.4 s[1]	3.8×
9	FPGA: Xilinx Kintex7 xc7k325t, DUT: PC28F00P30 NOR Flash			
	Active cells	1631	106	15.4
	Program + Verify 1 MB	312.4 s	37.0 s	8.4×
10	FPGA: Lattice ECP3 lfe3c35, DUT: AT25DF021 SPI Flash			
	Active cells	675	16	42.2
	Program + Verify 1 MB	95.9 s	12.0 s	8.0×

solution was patented and integrated into industrial test system from Goepel Electronic company. The real-life experimental results of Flash programming with Boundary Scan and VRSC instrument are shown in Table 3. In all experiments, the test clock (TCK) frequency was 20 MHz, which is typically a maximum for Boundary Scan tests. For each experiment part names of FPGA and DUT are given. After that, the numbers of active cells are shown for each technology. For Boundary Scan this number represents the length of BSR, for VRSC – the length of dynamic part. A theoretical ratio is shown in the 4th column. Measured time needed to perform the specified task is presented afterwards. For the VRSC-instrument-based test, the time needed for FPGA configuration is included in the result. This time varies depending on the size of particular FPGA device. Being very small (less than a second) for small FPGA families like Xilinx Spartan3 or Altera Cyclone III, the configuration time may exceed 10 s for large contemporary FPGA families like Xilinx Kintex7 or Altera Arria V. As a result, the time for configuration usually takes from 1 % to 10 % of complete test time considering the average test that lasts for several minutes.

An obtained speed-up is shown in the last column. The difference between theoretical and practical time improvements is induced mainly by the hardware delays and time needed to configure FPGA (especially larger ones). Although our experiments were done on a high-performance test system, its hardware was not optimized for short DRshifts (transfers of test data into the scan registers). The measured hardware delays between subsequent DRshifts were remarkable contributing to the smaller obtained speed-up than expected. For DRshifts with 60–80 active cells the hardware delays take 25–30 % of test time, while for DRshifts with 12–20 active cells the delays take half of test time. Nevertheless, although obtained time is not optimal and far from the theoretical ratio, the experimental results show that the VRSC instrument provides a significant speed-up over state-of-the-art Boundary Scan technology. In all cases, the time needed for NOR Flash programming and verification reduces noticeably. For example, in the experiment #6 with large FPGA (Virtex5) and SPI Flash the time was reduced from more than 45 min to less than a minute.

7 Conclusion

We have presented an approach that temporarily utilizes an existing on-board FPGA for test purposes. The virtual reconfigurable scan-chains are implemented in a programmable logic of FPGA for the sake of shortening the time needed to perform scan-based test. Within this study, five different configurations of VRSC architecture were proposed and one of which was implemented in FPGA design. The advantage of presented technique is twofold. First, we exploit the efficacy of RSCs at no hardware cost. The obtained experimental results showed that the test time could be reduced up to 52× times. Second, the VRSCs are designed without targeting of specific test case or a Board-under-Test (BUT). Once compiled, a VRSC may be used for accelerating any scan test on any BUT. This feature revises the usability of FPGA-based instruments as pre-compiled instruments can be shipped to end-users before even knowing the actual application. This option radically shortens embedded-instrument-based test development time avoiding relatively long FPGA design and compilation cycle.

Acknowledgments This work was supported by EU FP7-2013-ICT-11: 619871 project BASTION as well as through European Regional Development Fund.

References

1. Aleksejev I, Jutman A, Devadze S, Odintsov S, Wenzel T (2012) FPGA-based synthetic instrumentation for board test. In Proc. of International Test Conference, Austin, USA
2. Asset Intertech Inc., *How to test high-speed memory with non-intrusive embedded instruments*, Whitepaper, 2012.
3. Breuer MA, Narayanan S (1993) Reconfigurable Scan Chains, A Novel Approach to Reduce Test Application Time. In Proc. of International Conference on Computer-Aided Design, USA
4. Ehrenberg H, Russell B (2011) IEEE Std 1581 — a standardized test access methodology for memory devices. In Proc. of International Test Conference, Austin, USA
5. Escobar JHM, Sachße J, Ostendorff S, Wuttke H-D (2012) Automatic generation of an FPGA based embedded test system for printed circuit board testing. In Proc. of Latin American Test Workshop, Quito, Ecuador
6. Ferry J, Scesnak J, Shaikh S (2005) A strategy for board level in-system programmable built-in assisted test and built-in self test. In Proc. of Internation Test Conference, Austin, USA
7. Geiger PB, Butkovich S (2009) Boundary-scan adoption – an industry snapshot with emphasis on the semiconductor industry. In Proc. International Test Conference, USA
8. Goepel GmbH., Cascon Galaxy System.
9. Guo C, Zhang Y, Wen Z, Chen L, Li X, Liu Z, Wang M (2011) A novel configurable boundary-scan circuit design of SRAM-based FPGA. In Proc. of Computer Science and Automation Engineering, China
10. IEEE Standard test access port and boundary-scan architecture, 2001.
11. Intel Inc., *Intel StrataFlash® Embedded Memory*, Datasheet.
12. Intellitech Corp., Infrastructure IP for programming and test of in-system memory devices, Whitepaper, 2003.
13. Micron Technology Inc., *Micron M25P32 Serial Flash Embedded Memory*, Datasheet.
14. S. P. Morley et al. Selectable Length Partial Scan: a Method to Reduce Vector Length. in *Proc. of Internation Test Conference*, USA, 1991.
15. Nejedlo J, Khanna R (2009) Intel® IBIST, the full vision realized. In Proc. of International Test Conference, Santa Clara, USA
16. Parker KP (2003) The boundary-scan handbook. Kluwer Academic Publisher, Boston, MA, USA, p. 373

17. Quasem S, Gupta S (2004) Designing reconfigurable multiple scan chains for systems-on-Chip. In Proc. of VLSI Test Symposium, USA
18. Samaranayake S, Sitchinava N, Kapur R, Amin MB, Williams TW (2002) Dynamic scan: driving down the cost of test. In IEEE Computer, October
19. Spansion Inc., *S29AL008D 8MBit Flash Memory*, Datasheet.
20. Toai V, Zhiyuan W, Eaton T, Ghosh P, Huai L, Young L, Weili W, Rong F, Singletary D, Xinli G (2006) Design for Board and System Level Structural Test and diagnosis. In Proc. of International Test Conference, Santa Clara, USA
21. A. Tsertov, R. Ubar, A. Jutman, S. Devadze. SoC and Board Modeling for Processor-Centric Board Testing. In *Proc. of 14th Euromicro Conference on Digital System Design*, Oulu, Finland, 2011.
22. Wang S, Cao W, Wang L, Wang N, Tao P (2013) A novel structure of dynamic configurable scan chain bypassing unconcerned segments on the fly. In Proc. of International Conference on ASIC, China
23. Wohl P, Waicukauski JA, Colburn JE (2012) Enhancing testability by structured partial scan. In Proc. of VLSI Test Symposium, USA

Igor Aleksejev received the PhD in Computer Engineering from TU Tallinn, Estonia in 2013. Since 2008, he is a researcher at the Dept. of Computer Engineering of the TU Tallinn. His research activity focuses on FPGA-based embedded instrumentation, board-level and system test.

Sergei Devadze received the PhD degree in computer engineering from TU Tallinn, Estonia in 2009 and currently holds the position of researcher in this university. His main research interests include usage of chip-embedded instrumentation for system test, extended structural board test, fault tolerance, and fault management architectures.

Artur Jutman is the Managing Director of Testonica Lab. His research interests include DFT, embedded instrumentation, and system test. He has a PhD degree in computer engineering from TU Tallinn, Estonia. He is a former member of the IEEE, the council member of EAEEIE Association, and of the Nordic Test Forum society.

Konstantin Shibin is an R&D engineer at Testonica Lab. He has an MSc degree in computer engineering from Tallinn University of Technology, Estonia. Currently he continues his studies as a PhD student at TU Tallinn. His research interests include embedded instrumentation and fault management. He is a student member of IEEE.