Multi-channel Boundary Scan Controller

(published: Proceedings of SPIE, 2003, vol. 5125, session 3)

Wojciech Zabolotny*[a], Seweryn Jodlowski**[a], Michal Pietrusinski***[b],
Krzysztof Pozniak****[a], Maciej Kudla*****[b]
[a]Warsaw University of Technology faculty of Electronics and Information Technology;
[b]Warsaw University faculty of Physics

ABSTRACT

The paper presents, a specially designed chip, which ensures the carrying out of tests compliant with IEEE 1149.1-1990 Std. The above mentioned tests are very useful when checking for any errors and defects in electronics devices. The currently available solutions are not suitable for the requirements, which are quite specific and are the result of past experience. Therefore, a dedicated Field Programmable Gate Array (FPGA) core was created in VLSI Hardware Description Language (VHDL) implementing the desired functionality. The achieved result is a BS controller supporting many BS chains, working in parallel and using pipelining to boost performance.

Boundary Scan, JTAG, VHDL, IEEE 1149.1-1990, TAP, LHC, CMS

1. INTRODUCTION

One of important component in Compact Muon Solenoid (CMS) detector to be used in Large Hadron Collider (LHC) are Resistive Plate Chambers (RPC) ([1]). One of electronic systems supporting RPC's are Muon Trigger and Data Acquisition Boards (TriDAQ). These are complex boards containing many FPGA chips and many onboard connections. Due to its complexity it is necessary to provide means for effective hardware testing both during the assembly phase, and during the normal work of the system.

The malfunctioning of TriDAQ boards may result from many factors, such as: technical problems, crosstalks or mechanical defects. To prevent a situation where after a few months of running, some defects in trigger functioning are detectable, there is a necessity to have a dedicated module ensuring the prompt fault detection. It is also necessary that the trigger's electronics should be tested during the design phase.

Advances in technology have made traditional testing methods extremely difficult. Current in-circuit and functional testing techniques are becoming less effective in addressing industry-wide problems. The main causes of this situation are: complexity of components, surface-mount interconnection technology and shrinking trace-to-trace spacing. Because of these testing problems, in 1986, an international group of electronic manufactures, including Texas Instrument, was formed. This group was called the Joint Test Action Group (JTAG).

JTAG developed the boundary-scan proposal, which, in 1990, became IEEE Standard 1149.1-1990 Test Access Port and Boundary-Scan Architecture ([2]). IEEE 1149.1-1990 describes how to build circuitry into an integrated circuit to help test, maintain and support printed circuit boards ([3]).

Thanks to special functions ensured by IEEE 1149.1-1990, there is a possibility of testing devices produced by different companies and used in different branches of industry. Also a chip's designer does not need to worry about testing problem, and the only thing which he has to do is a circuitry built into a device compliant with Test Access Port and Boundary Scan Architecture. This architecture required only four pins for writing and reading data bits to or from a tested device. Direct driving of Test Access Port (TAP) pins by CPU would be very inefficient, and would consume a lot of CPU time, therefore it is necessary to provide a dedicated chip (module) able to efficiently communicate with CPU and on the other hand to communicate with TAP interface of tested board. This second feature could be useful in case of necessity to program FPGA by JTAG port.

2. OTHER CURRENTLY AVAILABLE SOLUTIONS

Unfortunately, on the market, there are not many similar solutions. One of them is National Semiconductor's chip PSC100, and the others are IP cores for BS controller available at http://www.opencores.org.

The PSC100 is a microchip, which is designed to interface a generic parallel processor bus to a serial scan test bus ([4]). It is useful in improving scan throughput when applying serial vectors to system test circuitry and reduces the software overhead that is associated with applying serial patterns with a parallel processor. The PSC100 operates by serializing data from the parallel bus for shifting through the chain of 1149.1 compliant components (i.e., scan chain). Scan data

returning from the scan chain is placed on the parallel port to be read by the host processor. Up to two scan chains can be directly controlled with the PSC100 via two independent TMS pins. Scan control is supplied with user specific patterns which makes the PSC100 protocol-independent. Overflow and underflow conditions are prevented by stopping the test clock. A 32-bit counter is used to program the number of TCK cycles required to complete a scan operation within the boundary scan chain or to complete a PSC100 Built-In Self Test (BIST) operation.
The features of the PSC100 are:

- compatible with IEEE Std. 1149.1 (JTAG) Test Access Port and Boundary Scan Architecture
- uses generic, asynchronous processor interface; compatible with a wide range of processors and PCLK frequencies
- directly supports up to two 1149.1 scan chains
- 16-bit Serial Signature Compaction (SSC) at the Test Data In (TDI) port
- automatically produces pseudo-random patterns at the Test Data Out (TDO) port
- fabricated in FACT™ 1.5 µm CMOS process
- supports 1149.1 test clock (TCK) frequencies up to 25 MHz

The PSC100 has not satisfied the requirements because of the following reasons:

- only two independent chains could be used at the same time
- it has a different way of programming which is not compatible with the previous, old version of the BS controller
- it generates interrupt after transfer of each word the BS chain - which results in too high interrupts rate

The IP core available at http://www.opencores.org is written in Verilog Hardware Description Language. This implementation of the TAP is fully IEEE 1149.1 compliant ([5]). It includes a TAP controller, a 4-bit instruction register and three test data registers: idcode register, bypass register and boundary scan register. Boundary scan register is connected to eight pins (2 inputs, 2 outputs, 2 tristatable outputs and 2 bidirectional pins). Besides the Verilog code, a BSDL file is also provided. The number of pins can be easily increased by following the instructions. The design had been tested with the JTAG Technologies testing equipment (The TAP controller was implemented in Xilinx 95144XL). The design will be expanded in the future to support additional instruction and debug capabilities. Development interface is used for development purposes (Boundary Scan testing and debugging). It is an interface between the RISC, peripheral cores and any commercial debugger/emulator or BS testing device. The external debugger or BS tester connects to the core via JTAG port that is fully IEEE 1149.1 compatible. The Development Port also contains a trace buffer with support for tracing the program flow, execution coverage and profiling the code. This second solution was not available when the work on the multi-channel boundary scan started and because of this it was not used.

3. MULTI-CHANNEL BOUNDARY SCAN CONTROLLER

To overcome limitations of the above mentioned solutions a flexible FPGA based controller was proposed. The first attempt to create it was a BS controller created in 1995 by Wojciech Zabolotny in AHDL language. However, because of limitations of AHDL, this solution has some shortcomings and the wish to eliminate them was the reason for starting the present project.
The main drawback of the old version was caused by AHDL features which make it very hard to design a project in such a way that its key parts would function in parallel, independently of each other. This fact was clearly visible during the implementation of the part of chip responsible for memory service. During read and write operations the rest of the chip had to wait idly for their completion.

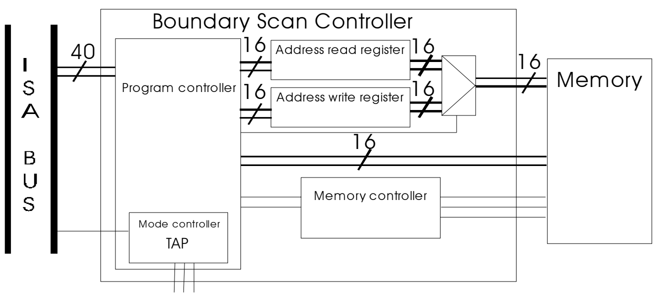

Figure 1: Schematic of previous version of Boundary Scan controller

For this reason a new version of BS controller has been entirely done in VHDL language, which allows for more natural description of parallel systems. The project can be divided into blocks called processes and the designer's task is only to synchronize functioning of the whole chip.

Another reason was the wish to increase complexity of tests done by BS. In response to demand, the new version allows to control up to nine, independent scan chains, able to work in parallel. It has larger address space, as well as more sophisticated handshake between the user and the chip.

In addition to this, support for additional controller states has been implemented. These controller states are shown on the controller diagram compliant with IEEE 1149.1-1990 Std. and they were not used in the previous version.

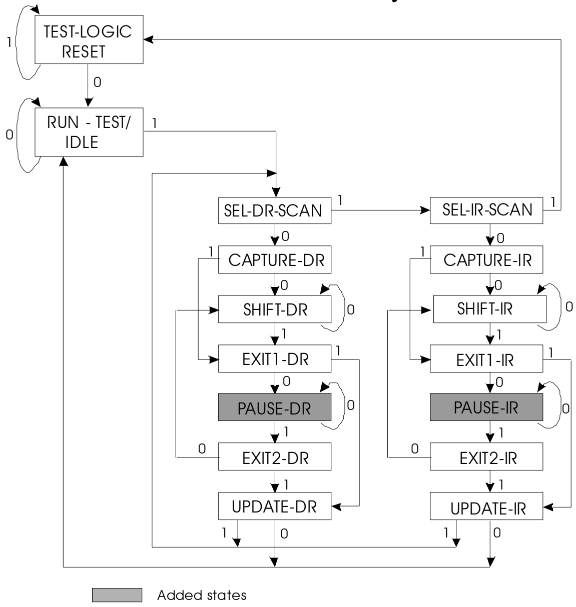

Figure 2: TAP Controller State Diagram

The designed chip is a synchronous automat, which could be treated as an extremely simplified microprocessor. The controller executes a program consisting of simple commands defined as 16-bit words. Syntax of commands was defined while creating the first version of BS in 1995.

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	Commands
0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	data
1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	end of program
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	return to Test-Logic-Reset state
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	return to Run-Test/Idle state
1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	inclusion of instruction register to BS chain (Shift-IR state)
1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	inclusion of data register to BS chain (Shift-DR state)
1	0	0	1	0	0	0	0	0	0	0	0	1	R	W	N	return to Run/Test Idle state from Shift(IR/DR) state with sending a N bit
1	0	0	1	0	0	0	0	0	0	0	1	0	R	W	N	transition between Pause(IR/DR) state and Shift (IR/DR) state with sending a N bit
1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	return to Shift(IR/DR) state from Pause(IR/DR) state
1	1	0	0	N	N	N	N	N	N	N	N	N	N	N	N	cyclic sending of bits in BS's loop without saving read bits (nnn – number of bits)
1	1	1	0	N	N	N	N	N	N	N	N	N	N	N	N	sending bits read from memory without saving received bits (nnn – number of bits)
1	1	1	1	N	N	N	N	N	N	N	N	N	N	N	N	sending bits read from memory with saving received bits (nnn – number of bits)
1	1	0	1	N	N	N	N	N	N	N	N	N	N	N	N	cyclic sending of bits in BS's loop with saving read bits (nnn – number of bits)

X – a value of bit (0 or 1) uses for setting the TDO outputs

R – read flag (indicate if N bit should be read from memory or not)

W – write flag (indicate if N bit should be write into memory or not)

Figure 3: Boundary Scan instruction set

The fifteenth bit was used to have a possibility for distinguishing between data and commands, 1 indicates instructions and 0 data. Furthermore, the microprocessor's instruction set list includes nine commands which are used for changing Test Access Port controller state (a special part of chip built into circuit, controlled by TMS and TCK inputs, which forces appropriate state of a tested device). The remaining commands ensured possibility of transmission of required numbers of bits through BS chains. The orders from this group provide following variation of bits' transmission:

- sending bits read from memory with or without saving received bits
- cyclic sending of bits in BS's loop with or without saving read bits

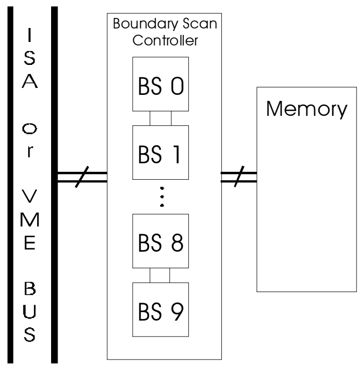

Figure 4: Schematic of new Boundary Scan controller

Each part of the chip responsible for one of the nine scan chains has its own memory address space in which a user can put appropriate sequence of instructions. Tests with BS controller are done according to the following algorithm: a user writes in asynchronous mode (in this mode a user is able to write/read memory into/from a chosen address space) the controller's program to the memory cells associated with the particular BS chain. Then he starts the test and after completing them particular blocks save their results into memory. In addition, when the BS controller finishes all operations an interrupt signal is generated.
This way of working requires a special type of synchronization, which ensures that no two blocks will try to use common resources at the same time (memory bus). This problem has been solved by implementing a priority chain using two signal lines, which connect all the blocks in serial. A block which requests access to common resources signals that fact using the request line and waits for an answer listening to the acknowledge line. When there is no objection from the other blocks (it means that none of the remaining blocks is using common resource at this time), it starts using common resources and signals this fact by the acknowledge line. After completing the read or write operation both line are set inactive. This type of synchronization was introduced in the INTEL 8080 microprocessor and ensured correct priority of interrupts. In presented priority chain preemption has not been implemented. A block which wants to use common resource has to wait until it is unused, even if this block has higher priority then block which is using common resource. Priority decided whose request will be serviced first in case of simultaneous requests from many blocks.

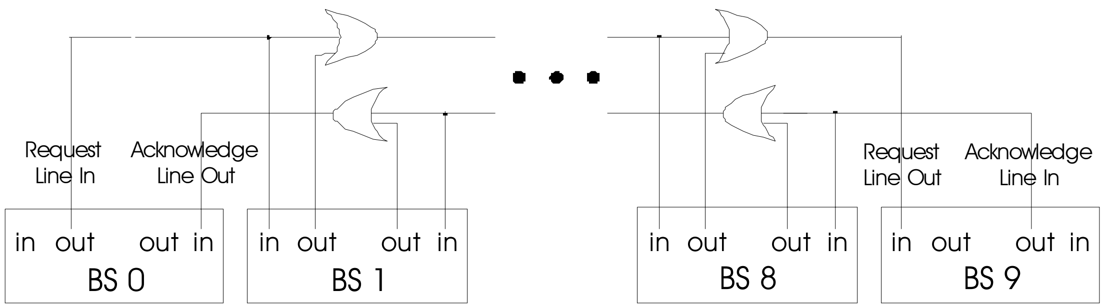

Figure 5: Diagram of priority chains

In order to enable data exchange and managing of BS controller's work into the discussed circuit special registers were built. These registers were mapped into PC's memory and allow for:

- setting up a start address for reading and writing
- choosing one of scan chains

- reading a current read and write address
- reading a status of executed operation
- writing data in asynchronous mode
- controlling of chip's work

The meaning of bits stored in the last register depends on entire device's mode. The BS controller could work in one of two modes. The first one is used for controlling selected scan chains, and the second one makes possible to write and read memory asynchronously. Choice between them is possible with a register which stores an identification number of working components. There are two groups of these components. The first one contains parts of chip responsible for one of scan chains, and the second – remaining part which allows for asynchronous operations.

The presented BS controller has the following features:

- compatible with IEEE Std. 1149.1 (JTAG) Test Access Port and Boundary Scan Architecture
- directly supports up to nine 1149.1 scan chains
- programmable number of TCK cycles required to complete a scan operation
- driving the values of TDO and TMS pins directly
- independent from hardware

4. TESTS

So far the tests of BS controller have been made by using simulation module included in the ALTERA MAX+plus program. Thanks to adding simulated memory, there was a possibility to test components responsible for memory service. For generating a basic test an appropriate list of commands was written in memory. This test showed errors in device's operation which resulted from incorrect connections and synchronization between components. Separately, all blocks were working properly but when joint together then behaved erroneously, in a very specific way. Only with preliminary simulation was this matter possible to be determined and solved. According to the results obtained from the MAX+plus program, the BS controller is able to work with frequencies up to 85 MHz.

5. SUMMARY

The Boundary Scan implementation presented in this article is a part of diploma work of one of the authors. Thanks to hierarchical body and using VHLD, future updates will be quite easy in implementation. Moreover, the suggested solution is not dependent from hardware. The main limitations are caused only by resources used for creating a Boundary Scan controller.

6. RERENCES

1. J. Krolikowski, G. Wrochna, M. Konecki, M. Kudla, A. Ranieri, E Pietarinen, K. Banzuzi, K. Pozniak and P. Zalewski, RPC trigger, in CMS, The Trigger and Data Acquisition project, C. E. Board, ed., I, pp. 419--480, CERN, 2000. Also available as http://cmsdoc.cern.ch/cms/TDR/TRIGGER-public/CMSTrigTDR.pdf
2.IEEE Standard Test Access Port and Boundary Scan Architecture, IEEE Std. 1149.1-1990
3.Colin M. Maunder, Rodham E. Tulloss, The Test Access Port and Boundary Scan Architecture, 1992
4.Scan PSC100 Datasheet , Embedded Boundary Scan Controller, 1998
5.Igor Mohor JTAG Test Access Port (TAP), http://www.opencores.org

*wzab@ise.pw.edu.pl; http://www.ise.pw.edu.pl/~wzab; **sjodlows@elka.pw.edu.pl; ***mpietrus@fuw.edu.pl; ****pozniak@ise.pw.edu.pl; *****kudla@fuw.edu.pl