AN EMBEDDED BOUNDARY SCAN TEST SYSTEM

Nuno Cardoso [(1)], Carlos Beltrán Almeida [(2)]

[(1)] INESC - R Alves Redol 9, 1000 Lisboa, Portugal
Phone (41) 227679479 Fax (41) 227678940 e-mail: Nuno.Vaz.Cardoso@cern.ch
[(2)] INESC - R Alves Redol 9, 1000 Lisboa, Portugal
Phone (351) 213100394 Fax (351) 213145843 e-mail: cfb@inesc.pt

Abstract – *This paper describes an application of boundary scan IEEE Std. 1149.1 at system level. It provides the description of the design and the implementation options of a VME boundary scan controller board prototype and the corresponding software.*
The prototype was designed to use the Module Test and Maintenance (MTM) bus, existing in the VME 64x backplane, to apply the IEEE 1149.1 test vectors to a single board with a specific test infrastructure also described in this contribution.
The software takes the boundary scan test/programming vectors generated by an automatic test pattern generator (ATPG) in serial vector format (SVF), and converts them into a format suitable to be used in the considered test infrastructure.

Keywords - Boundary scan test, system level test, VME instrumentation.

1. INTRODUCTION

This contribution describes a hierarchical boundary scan test system for VME applications [1]. The system has been designed to perform the test verification of the data acquisition boards (DAQ) and the data concentrator boards (DCC) used in the Electronic Calorimeter (ECAL) of the Compact Muon Solenoid (CMS) experiment [2, 3, 4] during either the production phase or the experiment lifetime.

One prototype of the boundary scan control board was produced which uses the MTM bus of the VME64 backplane to apply the IEEE 1149.1 [1, 5] test vectors to each one of the nineteen data acquisition boards existing in each subsystem. The complete experiment is composed of sixty-four subsystems. This prototype includes local memory, test control circuitry and an embedded boundary scan controller circuit responsible for driving the 4 test lines described in the IEEE 1149.5 standard [6] existing in the MTM bus [7].

Furthermore, a software package was developed to convert the test vectors in serial vector format (SVF), generated by an ATPG, into a format suitable to be used in this system architecture. A novel proprietary method named Scan Compress is used to reduce the amount of data occupied by the test vectors. Test result is a pass/fail assessment. Further diagnosis information requires the utilization of additional software test tools not considered here.

The contribution will focus three topics: the test system architecture, the implementation of the test controller board and the software developed to process the test vectors and to control the test operation of the board.

This work is the result of the collaboration between INESC and LIP in the Compact Muon Solenoid experiment being conducted at CERN under the contract PRAXIS XXI number CERN/P/FIS/15170/99.

2. ARCHITECTURE OF THE BOUNDARY SCAN TEST SYSTEM

The IEEE 1149.1 boundary scan test standard was originally developed for testing a single board using a serial linked TDI/TDO scan path. This solution can not be generalised to test several boards, sharing a common bus where the test signals run in parallel.

The given solution uses a boundary scan interface device (Scan Bridge) in each board under test [8] – Fig.1. This circuit provides a connection between the common test bus and the local boundary scan paths thus forming a hierarchical test system – Fig.2.

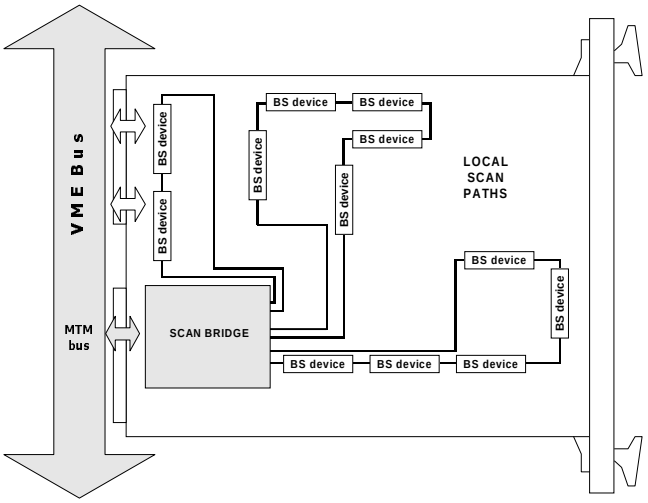

Fig.1 – Board under test: the Scan Bridge circuit is used as test interface between the MTM bus existing in the VME backplane and the local boundary scan paths.

Each board under test (designated DAQ and DCC boards in Fig. 2) interfaces the test lines of the MTM bus using a similar interface device. The test operations are applied by the Boundary Scan Controller (BSC) board, described in the next section, which communicates with the interface device using a 1149.1 compatible protocol .

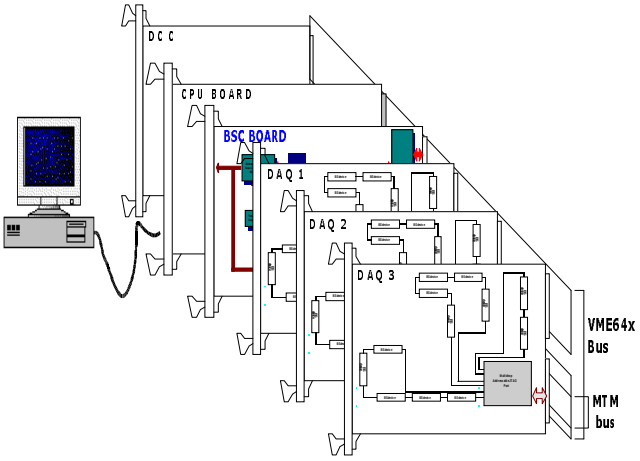

Fig.2 – Hierarchical boundary scan test system. The local boundary scan paths in each board under test (boards DCC and DAQ) are connected to a test bus through a interface device. The boundary scan controller board (BSC board) manages the application of the test.

This architecture extends the functionality of the IEEE 1149.1 Standard by offering an addressing scheme, which allows the test controller to communicate with a specific Scan Bridge within a network of these circuits. Each Scan Bridge in the system is configured statically with a unique address. After being addressed using IEEE 1149.1 compatible protocol, the Scan Bridge becomes the only device selected to receive additional IEEE 1149.1 test instructions.

The system provides boundary scan test for the data boards, including in-line configuration of programmable devices having 1149.1 infrastructure and interconnection test between different boards interfacing the VME bus through boundary scan transceivers [9].

3. THE BOUNDARY SCAN CONTROLLER BOARD

The Boundary Scan Controller (BSC) board works as a stand-alone tester applying the test vectors and reading back the test results. This board was implemented in a 6U sized VME board, with the configuration shown in Fig. 3.

A local embedded boundary scan device generates the IEEE 1149.1 test signals applied to the MTM bus. This circuit has a parallel interface section interfacing with the local test controller circuit (LTC) and a serial interface section interfacing with the MTM lines or with an external JTAG port. A 32-bit counter (not shown in figure 3) is used to count the number of TCK cycles to complete a scan operation or a Built-In-Self-Test operation.

A large local memory is used to store the test vectors, the control data and the test results. The test can be fully applied without the interaction with the test software if the memories are large enough to store all the test data (test vectors and test control data). In this case, the test application time depends only on the number of clocks and on the test clock frequency, which can be programmable between 400 kHz and 25 MHz.

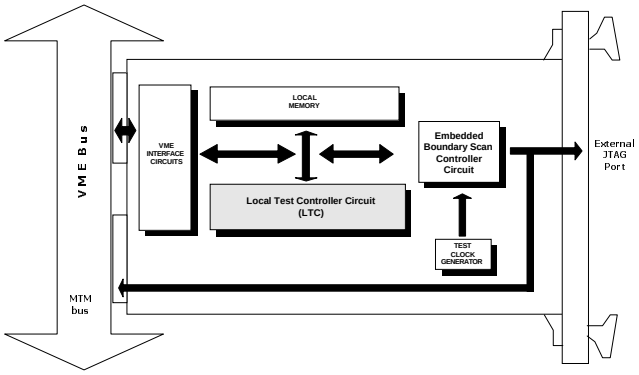

Fig.3 – Simplified architecture of the BSC board.

A VME interface block implements the data transactions required by the VME protocol when accessing the BSC board. It decodes the data transfer cycles generated by a master VME board (CPU board in figure 2) to access the local circuitry. The data transfer cycles are used to:

- verify the current state of the test board;
- load the test vectors and the test control data into the local memory;
- start, re-start and stop a test;
- read the test results from the local memory.

A local test controller, implemented in a re-configurable PLD, supervises the operation of the test board and its functionality.

4. SYSTEM SOFTWARE

The developed software uses the test or programming vectors generated by a commercial ATPG tool or PLD/FPGA development station. Most of these tools use proprietary formats to describe their test and programming vectors. In general, these formats are not compatible each other. However, the Serial Vector Format (SVF) developed by Texas Instruments and Teradyne in 1991, is accepted by the majority of these tools and therefore it was chosen as the input format to the test software.

SVF employs ASCII characters to specify the input and response vectors and the mask patterns to be applied in each set of vectors. It uses the concept of scan offset, which means that a test or programming file generated for a specific circuit can be re-used, after minor modifications, once that circuit is placed in a scan chain with other BS

circuits. In this case a simple introduction of a scan offset is sufficient to re-use the file.

The SVF format is used, not only for testing purposes, but also for in-circuit programming of re-configurable devices like Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs). In this case a SVF file is generated with the scan operations used to load the programming instructions into the device through their boundary scan interface port. This in-circuit configuration process uses the boundary scan test lines to load programming data and commands to the target device.

4.1 The SVF Converter

Our test software enhances the capabilities of the ATPG and programming tools by extending the test/programming application to a system composed of several boards with the configuration given before. The software architecture is shown in figure 4.

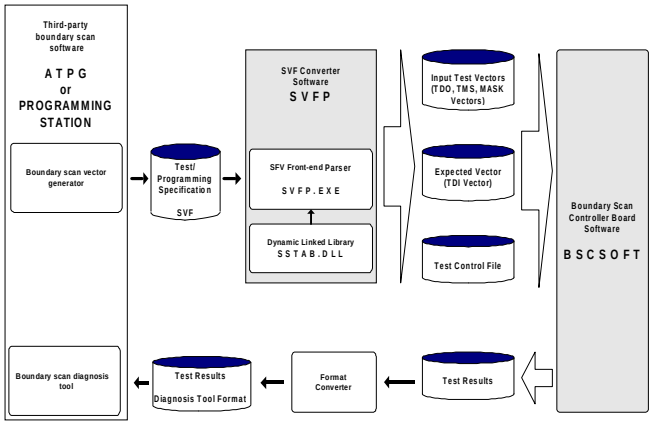

Fig. 4 – The system software is composed by two distinct tools. The SVF converter that translates the test/programming specification into SVF format and the BSC Software which controls the test operations of the BSC board.

The SVF Converter (SVFP) takes the test or programming specification in SVF format and generates the test files used in the BSC board software (BSCSOFT).

Depending on the tool used to generate the SVF file, two situations might occur:

1) The SVF specification includes only the test/programming data. In this case it is necessary to include additional BS offset instructions taking into account other circuits existing in the local scan chain and include the interface device (Scan Bridge) standing in each board under test.
2) The SVF specification includes the test/programming data and the BS instructions concerning other circuits in the chain and the interface device (Scan Bridge). In this situation the SVF file can be applied directly into the system after the conversion process.

For each situation the conversion software should take the appropriate actions in order to apply the test/programming file to the target system.

4.2 The BSC Board Software

The BSC board software (BSCSOFT) uses the files created by the preceding tool to manage the operation of the BSC board. This tool selects the type of the test data output and evaluates the pass/fail status.

The BSC board software is executed in the master CPU board placed in each VME crate. It uses the VME routines to access the local memory and the local registers of the BSC board.

These software modules, the SVFP tool and the BSCSOFT, run in different computer platforms.

4.3 Compression of Test/Programming Data

Difficulties arise due to the large amount of data and time required to apply the tests and to in-circuit program.

During boundary scan shifting of long scan chains, the TMS level is kept at logic low level for a long number of clock cycles. During the execution of device programming routines it is frequent do spend a long number of clock cycles during the Run-Test/Idle, Pause-DR and Pause-IR states. A similar situation occurs when executing the RUNBIST instruction in a device with BIST. The actual self-test runs when the TAP is placed in the Run-Test/Idle state. For a BIST operation taking many cycles to execute, the amount of TMS data increases proportionally.

A method, named Scan Compress is used to reduce the amount of data taken by the input and output test vectors (TMS, TDO and TDI vectors). Therefore, the test vectors are processed in order to detect logic level repetitions along several test clock cycles. A special type of codification was used to assign level repetition in the TMS line, avoiding the unnecessary waste of memory resources.

Significant improvements were achieved in the conversion of SVF files for in-circuit programming and for built-in self-testing. Table 1 illustrates some of the results achieved using the compression method.

Four applications, (1) to (4) are presented. The first two represent two test boards, named BSC and JTAG, which were tested using our system. The last two applications were used just for verifying our SVF converter.

For each application, different SVF files were used, having different sizes and exercising different actions – testing or programming. The same files were also converted using the compression of the TMS vector data. The number of control blocks shown in Table 1 gives an idea of the overhead achieved by the compression codification.

However, the compression method was found to be ineffective for tests with short shifting periods. In these cases, the amount of data is increased with the application of the compression algorithm. For large duration tests the compression can significantly reduce the amount of data. For BIST applications the TMS compression resulted in a considerable reduction of data, 33 percent. Further improvements are expected when applying the same process to the TDO and TDI vectors.

Table 1 – Results obtained from the conversion of several SVF files.

SVF test/programming file	# Clock cycles	TMS vector length (bits)	TDO vector length (bits)	TDI vector length (bits)	Control blocks (x48bits)	Memory occupied with compression (%)	Scan Compress used ?
(1) JTAG PF2150 Demonstration Board (7 circuits)							
tapit.svf	251	251	251	251	2	-	NO
tapit.svf	251	123	251	251	8	+19%	YES TMS vector
vit1.svf	2262	2262	2262	2262	2	-	NO
vit1.svf	2262	694	2262	2262	28	- 5%	YES TMS vector
(2) BSC Board Testing (7 circuits)							
ltc10.svf	48695	48695	48695	48695	2	-	NO
ltc10.svf	48695	2487	48695	48695	118	- 28%	YES TMS vector
(3) Built-In Self Test (BIST) Execution (1 circuit)							
bist.svf	120126	120126	120126	120126	2	-	NO
bist.svf	120126	126	120126	120126	6	- 33%	YES TMS vector
(4) – PLD Programming (1 circuit)							
altera1.svf	6 443 096	6 443 096	6 443 096	6 443 096	10	-	NO
altera1.svf	6 443 096	2 184 920	6 443 096	6 443 096	48768	- 10%	YES TMS vector

5. RESULTS AND FUTURE WORK

The software system was used in conjunction with commercial boundary scan software. The SVF files generated by this commercial tool were used for testing the boundary scan circuits and the interconnections of two digital boards having the test infrastructure described before. Until now the application of tests has been performed using an external connector because the VME boards available for the experiment do not use the MTM bus lines.

The SVFP tool was used to convert several SVF format files, with different sizes and for different applications, either for testing or for in-circuit programming of PLD configurable devices. The later one, due to the use of a specific SVF instruction (RUNTEST), need to be pre-processed manually to be accepted by the SVFP converter tool. The inclusion of this instruction in the pre-processing tool must be carried out in the future.

The Scan Compress method was found to be effective in SVF files with large shifting periods (normally used for in-circuit programming and for built-in self-test applications). Until now the compression method was used only in the TMS vector. However, further reduction in the test data size is expected in the future when the method will be applied to the TDO and TDI test vectors.

The operation of the complete test system in the hierarchical test infrastructure will be done in a near future, when the new generation of data acquisition boards with MTM lines will be delivered.

REFERENCES

[1] Harry Bleeker, Peter van den Eijnden, Frans de Jong, “Boundary-Scan Test - A Practical Approach”, Kluwer Academic Publishers, 1993.

[2] Carlos Beltrán Almeida, et al “Testability Issues in the CMS ECAL Upper-Level Readout and Trigger System”, Proc. of the 5th Workshop on Electronics for LHC Experiments, Aspen, 1999.

[3] CMS - The TriDAS Project, Technical Design Report, Vol1: The Trigger System, ISBN 92-9083-110-2, CERN, December 2000. WWW:http://cmsdoc.cern.ch/cms/TDR/TRIGGER-public/trigger.html

[4] The CMS Experiment: an Experiment for the Large Hadron Collider at CERN. WWW: http://cmsinfo.cern.ch/Welcome.html

[5] IEEE Standards Board , “IEEE Std 1149.1-1990 : IEEE Standard Test Access Port and Boundary-Scan Architecture”, New York, 1990.

[6] Dave Landis, Chuck Hudson, Pat McHugh, “Applications of the IEEE P1149.5 Module and Maintenance Bus”, Proceedings of the International Test Conference, 1992, pp. 984-992.

[7] Wade D. Peterson, “The VMEbus Handbook”, ISBN 1-885731-08-6, VMEbus International Trade Association, Scottsdale, Arizona, USA, 4th edition, 1997. WWW: http://nicewww.cern.ch/~lebwshop/LEB99_Book/DAQ.html

[8] Cary Champlin, “IRIDIUM Satellite: A Large System Application of Design for Testability”, Proceedings of the International Test Conference, October 1993, pp. 392-398.

[9] Wuudiann Ke, “Backplane Interconnect Test in A Boundary-Scan Environment”, Proceedings of the International Test Conference, October 1996, pp. 717-724.