

Boundary-Scan Test Control in the ATCA Standard

David Bäckström^{*}, Gunnar Carlsson⁺, Erik Larsson^{*}

Embedded Systems Laboratory^{*}
Linköpings Universitet
Sweden

Digital Processing Platform⁺
Ericsson AB
Sweden

Abstract -- The backplane in a multi-board system has a limited wiring capability, which makes additional backplane Boundary-Scan wiring to link the boards highly costly. The problem is to access the Boundary-Scan tested boards with the Boundary-Scan controller at the central board. In this paper we propose an approach suitable for the Advanced Telecom Computing Architecture standard where we make use of the existing I²C-bus and the Intelligent Platform Management Bus (IPMB) protocol for application of operational tests. We have defined a protocol with commands and responses as well as a test data format for storing test data on the boards to support the remote execution of Boundary-Scan tests. For validation of the proposed approach we have developed a demonstrator.

I. INTRODUCTION¹

A modern system often consists of several printed circuit boards (PCB) connected through a back-plane. In such a multi-board system, there is usually one central control board responsible for the control of the rest of the boards in the system. The boards in a system are usually equipped with Boundary-Scan to ease testing [5]. In operation and maintenance testing the boards must be accessed through the control board and a linkage is required. The backplane can for this purpose be extended with additional wires for Boundary-Scan.

There are several commercial solutions available to link a Boundary-Scan bus from the test controller on the central board to the local Boundary-Scan infrastructure through the backplane environment. However, these solutions require additional Boundary-Scan wiring in the backplane.

In addition to additional wiring in the already crowded backplane, there is also a need for a well defined way to control and manage downloading, storing and execution

of the on-board test sets. Today, almost every vendor of automatic test equipment (ATE) has its own specific API/command set and data format to transport and store onboard tests. The lack of a standardized command set is probably one of the most important reasons for the low deployment of embedded Boundary-Scan today. Also, this has led to unnecessary difficulties when designing system tests for multi-board system where boards or parts are often made by several different vendors.

In this paper we present an approach to limit the backplane wiring problem. Our approach is suitable to be included for systems based on The Advanced Telecom Computing Architecture (AdvancedTCA or ATCA) standard; an architectural multi-board platform for carrier-grade telecommunication applications [6]. We make use of the existing and well-known maintenance architecture in the backplane given in the ATCA; the I²C-bus and the Intelligent Platform Management Bus (IPMB) protocol. For the problem of lacking a standardized command set, we present a well defined command set and an embedded data format. This will provide both the ATE manufacturers with a uniform application interface and the system designers with an easy to implement data and control structure. The presented command set together with the data format is able to handle the following three levels of test scenarios:

- Embedded go/no-go test.
- Embedded diagnostic test.
- Remote diagnostics.

We have developed a demonstrator to validate the proposed scheme. And we assume that the reader of this paper has some basic knowledge about Boundary-Scan.

The rest of the paper is organized as follows. Section II gives an overview of related work and Section III introduces the preliminaries and the assumed environment. The problem definition is in Section IV and the proposed approach is described in Section V. Section VI contains a discussion, Section VII presents the demonstration and the paper is concluded with conclusions in Section VIII.

¹. The research is partially supported by the Strategic Integrated Electronic Systems Research (STRINGENT) programme.

II. RELATED WORK

This section gives an overview of previous work that has targeted embedded Boundary-Scan in a multi-board environment. There are several solutions available how to link Boundary-Scan control from a central test controller to locally distributed test controllers.

A. Boundary-Scan Architecture and Protocol

The backplane is extended with Boundary-Scan and the Boundary-Scan protocol is used. There are alternatives to connect the boards; the ring-architecture and the star-architecture. These are well-known but rarely used schemes in larger multi-board systems.

The ring-architecture creates a potentially long and cumbersome scan-chain to use. Such a solution in a multi-drop system also runs into problem when boards are removed or added, since it requires some sort of jumpers/bridges when a card is removed or the chain will be broken.

In the star-architecture every board in the system gets a dedicated TMS line and can then be controlled separately. However, such an approach requires a larger amount of connection lines in the backplane that might be available (*i.e.* one additional line for each card).

The advantage of the ring and star architectures is that they do not require any additional components or new protocols beside the required of the Boundary-Scan specification. This makes them simple to implement, but in a larger multi-board system often to cumbersome to use.

B. Boundary-Scan Architecture and Extended Protocol

The backplane is extended with Boundary-Scan and the Boundary-Scan protocol is extended. The approach is the one used mostly in today's systems.

Whetsel [4] presented a scheme where Addressable Shadow Ports (ASPs) are used to gain access to specific boards or scan-chains in a system. Texas Instrument supplies interface components that support this ASP scheme.

A variant of the approach was first proposed by Bhavsar [2] and later developed further by National Semiconductor into the SCAN Bridge scheme. In SCAN Bridge there is also an overhead protocol, but instead of using the Boundary-Scan bus in its idle states it shifts an address like an instruction using the shift-ir state on the TDI line.

Both the TI solution and the NSC solution are tightly linked to the 4-wire (optionally 5 wires) Boundary-Scan bus as a backplane interconnection (with their own specific modifications to the protocol). These additional wires, especially when an already implemented maintenance bus is available in ATCA, might be a deal-breaking

requirement.

In addition to the extra wiring in the backplane these solutions do not provide any detection of errors that may occur during backplane transmission. The result could expose the components-under-test to corrupted test and control data and in the worst case even damage the components. Ke *et al.* [3] proposed a novel scheme to include error detection into a standard Boundary-Scan backplane bus, such is used in the ASP and SCAN bridge designs. The IPMB protocol used in our solution does however already include error detection features.

C. Alternatives to Boundary-Scan

An alternative is to not make use of Boundary-Scan but instead make use of a totally different bus and protocol to transport test data in the backplane. The solution presented in this paper falls into this group, because we use the IPMB bus to transport and control the Boundary-Scan tests.

The problem with the IEEE standard 1149.5, the Standard Module Test and Maintenance (MTM), was that while it did specify the message transport interface on both sides, it did not specify an embedded test data format or a command set to run specific tests. This was left open to the users of the standard. It resulted in that the standard was never really adopted by the industry and is now abandoned.

Instead the Intelligent Platform Management Interface (IPMI) framework and the Intelligent Platform Management Bus (IPMB) bus was developed and is today increasingly gaining momentum through the ATCA system architecture. The IPMB bus is well suited for additional functions and only requires 2 wires in the backplane compared to MTM that requires 5 wires and ASP and Scan Bridge that require 4 each (standard Boundary-Scan Bus architecture).

Whetsel [4] also presented in his paper an expanded version of his proposed ASP scheme called Commandable ASP (CASP) where he suggests that CASP (or ASP) is possible to convey over a 2-wire serial bus instead of the 4-wire standard Boundary-Scan bus. The CASP scheme is however still bound to a particular bus protocol and architecture (2 or 4-wire) that is not available in today's system architectures like ATCA. Therefore it's essential to make the command set and the data format as platform independent as possible by separating it from the design of the bus architecture.

III. THE ATCA ENVIRONMENT

This section contains a brief introduction to the Advanced Telecom Computing Architecture standard (ATCA) [6]. The systems, which this paper is mainly

aimed at, are telecom and networking applications, like telephone and optical switches. These systems will be increasingly based on the ATCA-standard that specifies the mechanical building practice and the backplane interfaces [6].

The ATCA architecture is implemented using a *shelf* with several *slots* where control boards and application specific boards (*blades*) can be inserted. Often a master control board is used to control the operation of the application specific boards. The operational communication (control and user data planes) between the boards is mainly going through the *main backplane connection bus(es)*.

Specified in the ATCA-design is also a platform management system [7] (*i.e.* IPMI), for which the major role is power and thermal management or the “health” of the system by monitoring different type of sensors etc (see Figure 1). This usually includes management of functions like fan control, power control, temperature and voltage levels readings. IPMI uses a simple bus and protocol IPMB to communicate between the boards. In turn, IPMB is physically implemented using the I²C Bus (a serial 2-wire, widely used bus) as carrier.

In the IPMI concept, the platform management functions are controlled from a Shelf Master (SM, sometimes referred to as Shelf Management Controller - ShMC), typically placed on a system control board. At the application boards, Baseboard Management Controllers (BMC, sometimes referred to as IPM Controller - IPMC) are locally controlling the maintenance functions on the boards they sit on.

Note that IPMI is merely a framework for implementing new management functions. This paper suggests a way to implement fault detection into this structure.

IV. PROBLEM DEFINITION

In this section we will clarify the problem which the proposed solution in this paper is intended to solve. We

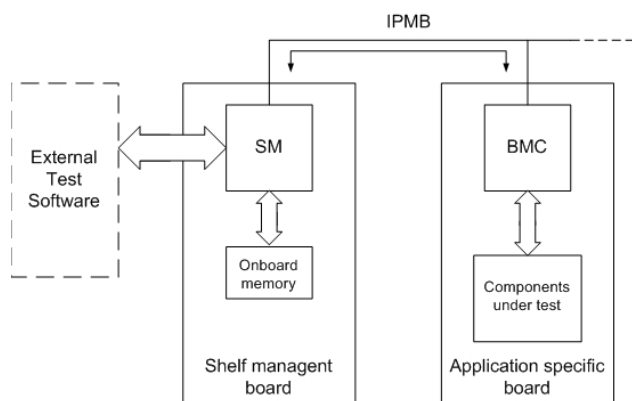


Figure 1: Basic IPMI structure.

will do this by describing three test scenarios and their related requirements on data and control transport. In the end of this section we will summarize the scenarios into a few goals and overall requirements.

A. Embedded go/no-go test

Typical usage is a test at cold (re)start of a board in the field due to a severe alarm, or as a regular test at non-busy hours. Short test time is often crucial. Control of the test may be through a system maintenance (operator) interface or through intelligent maintenance SW.

In this kind of test the test-set is usually resident locally on the target board/module and only called to run by a simple command from the maintenance controller on the main control board. The analysis and comparison of the test response data is also done locally on the board. When completed, the board test controller responds with a small pass/fail message back to the maintenance controller on the main control board.

B. Embedded diagnostic test

Typical use is an extended test to detect potential HW faults that are not detected by the go/no-go test. The reason may be frequent alarms and restarts with suspected HW problems. A slightly longer test time may be allowed, if it results in a higher resolution of the test. Control of the test may be through a system maintenance (operator) interface.

This kind of test is used to gain more information from a given set of stored test vectors/programs. As before, the comparison is still made locally on the target board, but instead of only sending a small test report back to the controlling operator, the operator can select to retrieve a test log or even the actual response data if needed.

C. Remote diagnostics

Typical use is diagnostic test in a reference system in the repair shop. The test will be more capable to pinpoint faults to components and interfaces. Control of the test may be through a system maintenance (operator) interface or from a connected external test system.

This is used when extensive testing and flexibility is needed. Beside the resident tests on the target board, new test sets can also be downloaded and run. The test responses is analyzed at the target or sent back to an external test system for comparison and further study. In this scenario longer test times are accepted due to the higher rate of availability and flexibility.

D. Goal and Overall Requirements

The requirements of these three scenarios together with

the requirement that the solution should fit in the ATCA context and not violate the IPMI management standard present the goals for the suggested solution.

V. PROPOSED SOLUTION

We will expand the usage of IPMI to include fault management, *i.e.* to let the IPMB carry Boundary-Scan test commands and data between the central SM unit and the locally distributed BMC units. This should be seen as a complement to the conventional way of having the main control processor controlling the testing of application boards, using the ordinary functional control path to the functional local board control processors. In this way, the IPMB could be regarded as a “system back door”, which allows testing even in case the ordinary control path is out of order. This way of using the IPMB is not violating the IPMI/IPMB standards.

The solution consists of three parts:

- A description of where and how the Boundary-Scan functionality is implemented in the ATCA/IPMI context.
- A specification of an API or set of commands used by a remote or an internal test program to manage the embedded tests and their execution.
- An embedded binary test vector format in which the locally onboard test sets will be stored in. This is the actual instructions that a low-level Boundary-Scan HW-driver will execute.

The test flow controller which receives and interprets the commands does not need any knowledge about the HW-implementation/driver and similarly the low-level HW-driver doesn't need any knowledge about the structure of the command set. This segmentation of our solution into several layers offers many advantages. First it reflects the typical different competence categories involved in the development of embedded tests. Secondly the separation between the test execution flow controls (*i.e.* the command set) and the low-level Boundary-Scan HW-driver enables reuse of the test controller in future applications of embedded Boundary-Scan test purposes. The second separation between the system communication links (and its management) and the test execution controls also eliminates the dependencies if commands are fetched from local onboard storage or sent from a remote site.

A. Added parts in IPMI

The IPMI is merely a framework for implementing management functions. It provides, among other things, interfaces and bus structures (*e.g.* IPMB) to support a distributed management system.

The main IPMI controller on the shelf management

board (*i.e.* the SM unit) will also be the main controller when performing Boundary-Scan applications. It receives commands from the system manager interface and according to those gives commands to the local satellite controllers (*i.e.* BMC's) through IPMI interface. Most of the low-level test data will be stored on the local boards in the system, but some additional test sets might also be stored on or sent to the shelf management board and under control of the SM unit. These additional test sets could be used when an extended embedded diagnostic is required on an application board (*i.e.* see the second test scenario above - Section IV.B). In this case the SM unit could even be performing some trivial comparison and analysis of the received test response. However, most of the time the SM unit will act like a bridge and command interpreter between the system management interface and IPMB link to the BMC's.

The local IPMI controllers on the boards are the units that perform the actual testing. They will receive commands and data from the SM unit through IPMB and act upon it. This is a short summary of the new functionality that has been added to the BMC:

- Write, read and manage onboard stored test sets.
- Run one or more onboard test sets.
- Perform comparison and straight forward analysis of the received test response.
- Logging of execution and results of test runs.
- Send test reports and logs to the SM unit when requested.

To ease of some of the burden of the BMC implementation, especially when handling the serial interface of Boundary-Scan bus, a special *embedded Boundary-Scan controller* could be used. This kind of controller is basically just an asynchronous parallel to serial (Boundary-Scan) interface and has been commercial available for some time.

B. Commands

This section contains a list and a description of each of the commands and the responses defined in the command set. The API enables the system management functions or a system operator an easy access to the onboard or remote embedded tests. The commands are divided into three groups:

- *embedded data management* commands,
- *test control* commands, and
- *support* commands.

1) Embedded Data Management Commands

The following *embedded data management* commands are available:

- LIST_TEST: Lists all test sets stored at BMC.

- LIST_LOG: Lists all test log entries stored at BMC.
- WRITE: Send/write and store new test sets at BMC.
- READ: Retrieve/read stored test sets or logs at BMC.
- DEL: Delete a specific test set or log entry at BMC.

2) Test Control Commands

The following *test control* commands are available:

- SET_TEST_RESPONSE: Control if each test should return test response always, only on failure or never.
- SET_TEST_MESSAGE: Control if each test should return its log entry always, only on failure or never.
- SET_LOG: Set if the result should be logged or not.
- SET_STOP_CONDITION: If tests should abort on first failure or continue whatever happens.
- RUN: Run test sets stored in the BMC.

3) Support Commands

The last group of commands is the support commands:

- HW_OPTION: HW specific commands to BMC.
- HELP: Displays a list of available commands.
- EXIT: Stops all execution of the Boundary-Scan.
- STATUS: Displays current system status.

C. The Embedded data format

All Boundary-Scan test vectors, expected test response and control data are stored in the system in a format called Binary Vector Format (BVF) (Figure 2). It is essentially a compact binary version of the Serial Vector Format (SVF) [8] and all posts are all well defined to make it as memory organization and processor independent as possible.

Every BVF-file is composed of a number of BVF partitions, and every BVF partition is composed of a number

of BVF records. Every BVF partition starts with a header record and ends with an end of partition record (*i.e.* EOP). The rest of the records are a mix of these;

- BVF_SDR, Scan Data Record.
- BVF_SIR, Scan Instruction Record.
- BVF_RUN, Run Test Record.
- BVF_TRST, Test Reset Record.

And these additional control records are also available:

- BVF_ENDDDR, Default end state for DR operations.
- BVF_ENDIR, Default end state for IR operations.
- BVF_STATE, Forces the Boundary Scan logic to a specific state.

The header record and the end of partition record together with the operating and control records above are all well defined and start (*i.e.* the first byte in the record.) with an op-code. This op-code determines how the rest of the record will be structured.

VI. DISCUSSION

The major drawback with the IPMB bus, and all serial busses, is the speed limit (*i.e.* the limits of the I²C Bus) of 100 kbit/s and together with a maximum packet size requirement of 25 data bytes (due to maximum overall message duration on the IPMB Bus of 20 ms) hampers the transport of larger tests sets significantly. The bit rate limitation of the bus mostly effects the operation in the the embedded diagnostic test and the remote diagnostics, the two last test scenarios presented above, since these two require larger amount of data to be transported. However, it is also those two scenarios that might allow a longer test time so this trade-off might be acceptable.

There are faster modes available in the I²C Bus standard (400 kbit/s in Fast-mode and up to 3,4 Mbit/s in High-speed mode.) However, these faster transfer rates aren't currently supported by the IPMI and IPMB specifications, which is one of the requirements in this project.

The first scenario presented is well suited for implementation in an ATCA/IPMI environment. The IPMB message interface was designed for conveying shorter status and control messages.

The command set presented in this paper is in reality only the core of the needed commands in a complete command set. We are aware that it might need modification and expansion to accommodate all the needs in embedded Boundary-Scan tests in a large multi-drop environment. However, one will come a long way by only using the WRITE, DELETE and RUN commands together with the defined embedded binary vector format.

In a more broad perspective where future efforts need to be made is in standardize a common and open command-driven interface which ATE and diagnostic tool vendors could use to develop embedded Boundary-Scan

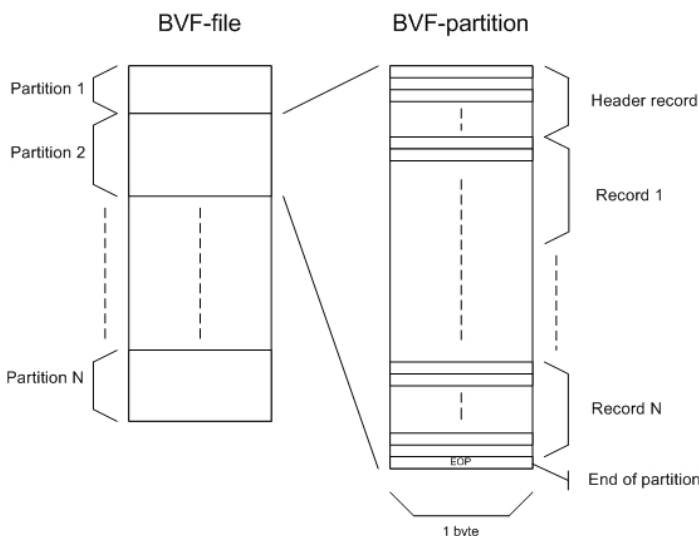


Figure 2: BVF-file format.

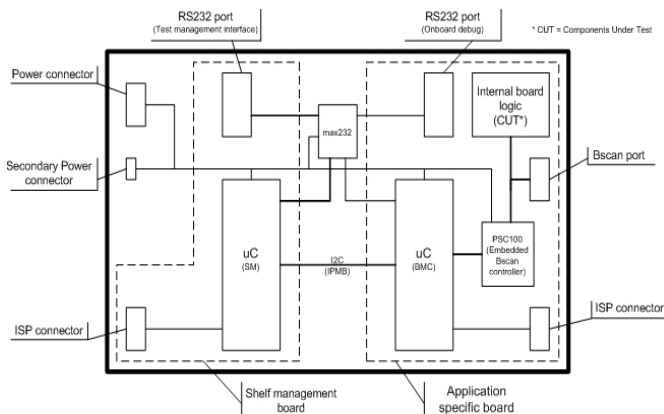


Figure 3: Schematic of the demonstration board.

tests more efficiently. Such an interface should have some of the characteristics of the API presented in this paper, especially the modular design and structured command and embedded vector format, to be successful. Today the tools are usually based on low-level, HW dependent, interfaces to access embedded Boundary-Scan paths in systems. One such common approach is a PCI-interface card with multiple TAP ports (a.k.a PODs) which directly link to the Boundary-Scan chains to the units under test. In our, and perhaps in a future solution a standardized interface could be used to access embedded Boundary-Scan tests through one single system test port.

VII. DEMONSTRATION

This section contains a description of the basic design of the demonstration board. The board consists of the fundamental parts of the ATCA/IPMI structure and has been implemented with selected a subset of the above presented new functionality. The objective is to display and validate that the solution presented in this paper.

The demonstration board is divided into two sides; the left side demonstrates the main shelf management board in an ATCA system and the right side demonstrates one of the application specific boards in the system (Figure 3). The two sides are using the IPMB Bus for communication as specified in the ATCA specification.

The shelf management side (left) has a micro controller acting as the Shelf Master (SM) and an external serial port for connection of a PC with management software. The application board side consists of a micro controller acting as Baseboard Management Controller (BMC), an Embedded Boundary-Scan controller and some internal components acting as targets for the Boundary-Scan tests (Components Under Test, CUT). This provides a complete scan-chain from the Boundary-Scan controller to the CUTs and back. There is also some support parts implemented on both sides, like RS-232 interface compo-

nents, a dc power connector, debug and in system programming (ISP) ports.

VIII. 8.CONCLUSIONS

A major problem in a multi-board system is the limited wiring capability in the backplane. Additional Boundary-Scan wiring to link the boards is therefore highly costly. However, the problem is to access the Boundary-Scan tested boards with the Boundary-Scan controller at the central board. In this paper we propose an approach suitable for the Advanced Telecom Computing Architecture standard where we make use of the existing I²C-bus and the Intelligent Platform Management Bus (IPMB) protocol for application of operational tests. We have defined a protocol with commands and responses as well as a test data format for storing test data on the boards to support the remote execution of Boundary-Scan tests. For validation of the proposed approach we have developed a demonstrator.

REFERENCES

- [1] K. P. Parker, "The Boundary-Scan Handbook", Kluwer Academic Publishers, 1992.
- [2] D. Bhavsar, "An Architecture for Extending the IEEE Standard 1149.1 Test Access Port to System Backplanes", *Proceedings of IEEE International Test Conference (ITC)*, Nashville, TN, USA, October 1991, pages 768-776.
- [3] W. Ke, D. Le and N. Jarwala "A secure data transmission scheme for 1149.1 backplane test bus", *Proceedings of IEEE International Test Conference (ITC)*, Washington, DC, USA, October 1995, pages 789-796.
- [4] L. Whetsel, "A Proposed Method of Accessing 1149.1 in a Backplane Environment", *Proceedings IEEE International Test Conference (ITC)*, Baltimore, Maryland, USA, September 1992, pages 206-216.
- [5] R. N. Joshi, K. L. Williams, and L. Whetsel, "Evolution of IEEE 1149.1 Addressable Shadow Protocol Devices", *Proceedings of IEEE International Test Conference (ITC)*, Charlotte, NC, USA, October 2003, pages 981-987.
- [6] PICMG 3.0: ATCA Core Short Form Specification, www.picmg.org/pdf/PICMG_3_0_Shortform.pdf
- [7] Intelligent Platform Management Interface, ftp://download.intel.com/design/servers/ipmi/IPMIv2_0rev1_0.pdf
- [8] Serial Vector Format specification, <http://www.asset-intertech.com/support/svf.pdf>