A Chip Solution to Hierarchical and Boundary-Scan Compatible Board Level BIST

Oliver F. Haberl, Thomas Kropf

University of Karlsruhe, Institute of Computer Design and Fault Tolerance (Prof. Dr. D. Schmid)
P.O. Box 6980, W-7500 Karlsruhe, Germany

To achieve the full benefit of self test approaches, current self test techniques aimed at chip level must be extended to whole boards and systems. There, the self test must be hierarchical and compatible to the standardized boundary-scan architecture. In this paper, a hierarchical boundary-scan architecture is presented together with the necessary controller chip and the synthesis software which make a hierarchical self test of arbitrary depth possible and provide sophisticated diagnosis features in case of failure detection.

1: Introduction

To overcome the limitations of traditional printed circuit board (PCB) testing techniques the IEEE 1149.1 boundary-scan architecture has been standardized [IEEE90]. It consists of an instruction based test access port controller (TAP) and a scan chain of all primary inputs and outputs for every chip (figure 1).

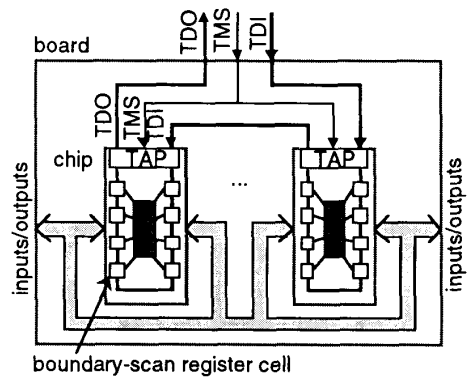

Figure 1: Board with standard boundary-scan

The boundary-scan architecture is aimed primarily at a test of the PCB interconnections and at a scan-based chip test with access limited to the primary inputs and outputs of every chip. Because of its scan nature such a test is of low speed and impractical for large systems. Hence a chip level self test is needed for standardized designs, using the RUNBIST instruction. Its implementation is not regulated by the standard and left to the designer. To achieve this goal, different approaches have been proposed ([Maie90], [HaKr92]). However, to get rid of expensive test equipment and to simplify in-field tests, whole PCBs and systems must be self testable, too. Hence self test concepts must be extended to board and system level. A board self test consists of the self test of every chip and a test of the interconnect wiring of the PCB. This holds for each level of hierarchy, e. g. a system test may consist of the self test of each board and the test of the backplane.

Assuming that all chips on a PCB have been designed in a boundary-scan compatible manner, different problems must be solved to achieve this goal:

- To be able to perform an autonomous self test of each chip on board by using the RUNBIST instruction, different self test parameters must be known: the encoding of the RUNBIST instruction, duration of the self test and the correct test response for each chip.
- The interconnect test depends on the wiring topology of the PCB, hence the interconnection structure in-between the chips must be known. Moreover, if busses have been used, the bus topology and the arrangement of the bus-enable cells in the boundary-scan registers of the chips must be taken into consideration for performing a short-free bus test. Additionally for wired-AND and wired-OR nets special test algorithms are necessary.
- To achieve a self test for whole systems in acceptable time, a hierarchical approach is needed, which consists of a component test *and* an interconnect test at all levels of hierarchy in parallel.
- To ease the replacement of faulty chips, mechanisms must be incorporated to identify these components.

With a pure boundary-scan topology as depicted in figure 1, the first three tasks may be solved in principle. However, expensive test equipment is still needed for an interconnect test, since all primary inputs and outputs of the PCB must be controlled and observed. Moreover, all relevant test information as described above must be known. Since external test equipment is needed, this technique may not be regarded as a self test.

0-8186-2610-0/92 $3.00 © 1992 IEEE

To provide PCBs with an interconnect test as well as with a test interface better suited for external test equipments, a test master chip has been proposed [JaYa91]. Nevertheless, it does not perform a self test of the chips on the PCB and fully hierarchical tests are not possible.

To achieve a hierarchical test, an architecture has been proposed in [LiBr89]. However, this approach needs different controllers each for a different self test task and hierarchy level.

In this paper an approach for a fully hierarchical self test of arbitrary depth is proposed. First, the hierarchical self test architecture is presented. It requires a boundary-scan controller chip (HBSC chip) realizing an IEEE 1149.1 compatible, instruction driven, self test controller at each level of hierarchy. The main tasks to be performed by the HBSC chips are described as well. The following section describes the functionality of the HBSC chip, i.e. the different self test instructions in more detail. Then, the implementation of the chip and its interaction with an EPROM containing necessary self test information is described. In addition, the software to automatically derive the contents of the EPROM from a given netlist description is briefly presented. The paper ends with a summary and conclusions.

2: The hierarchical boundary-scan architecture

To be able to perform a test at all hierarchy levels with a single type of test controller, it is necessary that every chip as well as every board is provided with a uniform test interface. For compatibility reasons the standardized test access port is used. This requires that the usual drivers at the PCB edge connectors have to be replaced by PCB boundary-scan register cells. The structure of a modified PCB with the HBSC chip and boundary-scan register cells is depicted in figure 2.

The HBSC chip is connected to the chip level TAP chain by the pins CTDO, CTMS and CTDI. Hence the HBSC chip can select all test functions of each chip on board by using their appropriate TAP instructions. The boundary-scan register cells at the edge connectors form the PCB boundary-scan (PCB bs). This scan path is connected to the HBSC chip by the pins BTDO and BTDI. The PCB boundary-scan is used to directly control and observe the signals at the PCB edge connectors.

The board itself is equipped with a standard test access port (PCB test access port). The handling of this TAP is identical to that of a chip as described in the boundary-scan standard. Hence a uniform test interface has been achieved and there is no difference in activating a self test of a chip designed in a boundary-scan compatible manner and a PCB designed in the described way.

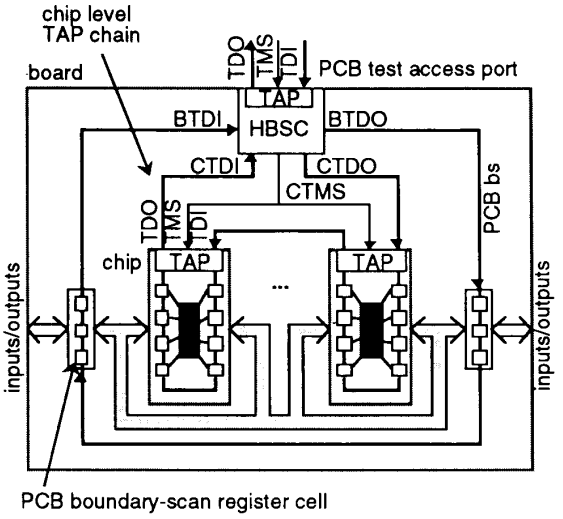

Figure 2: Board with HBSC chip

With this architecture all test features of the board are accessible in a standardized way via the PCB test access port, realized by a HBSC chip. This chip interprets each IEEE 1149.1 test instruction (EXTEST, BYPASS, INTEST, SAMPLE/PRELOAD, RUNBIST and IDCODE), activates the appropriate internal board level TAP signals, distributes test instructions to all chips on the board and collects test responses if necessary.

For example the execution of a whole self test of the board is activated by the RUNBIST instruction. This leads to a self test of all components on board – by simply providing each component with its RUNBIST instruction – as well as to a board interconnect test. The latter test is managed by the HBSC chip itself. For this purpose, it also controls the operation modes of the PCB boundary-scan register cells, which is necessary to separate the board during the interconnect test. After the self test is completed, a signature is provided in the HBSC chip in order to determine the correct functioning of the whole board. This self test procedure is described in detail in the next section.

Since the TAP interface is used uniformly for chips and PCBs, the test architecture may be extended to arbitrary hierarchical structures as shown in figure 3.

The activation of a top level self test propagates through the design hierarchy in such way, that each chip and all interconnects are tested at all levels of the hierarchy. Thereby the self tests of all chips and boards are performed in parallel. The test results are collected and transformed into an overall signature for the whole system. This signature is the result of the RUNBIST instruction at the top level HBSC chip.

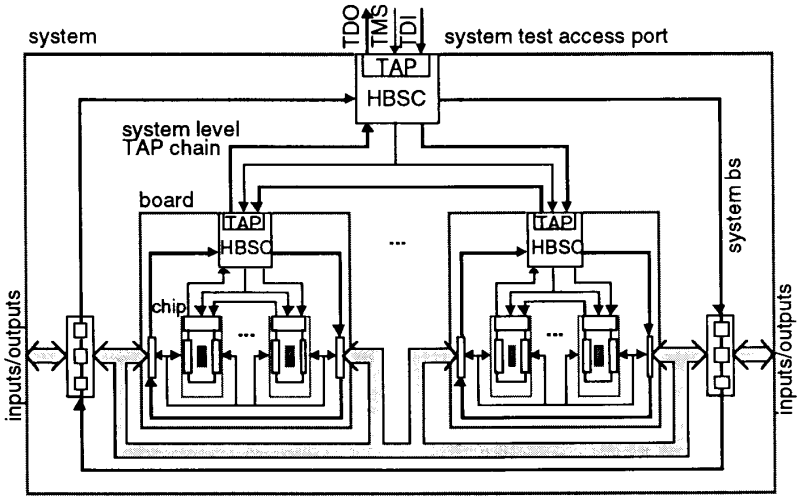

Figure 3: System with HBSC chips

3: The instructions within a hierarchical TAP architecture

Since every PCB now contains an IEEE 1149.1 interface, all standardized test instructions are possible at board level. Their execution is controlled by the HBSC chip. The instructions may be divided into three groups: instructions which are trivial or require only a simple access to the internal scan chain (EXTEST, INTEST, SAMPLE /PRELOAD, BYPASS and IDCODE), the self test instruction RUNBIST and diagnosis instructions (ERRORCODE and DEBUG). The instructions are described in the next sections assuming a PCB structure as given in figure 2. The effect of each instruction can be transformed in a straightforward manner to the architecture in figure 3 or more complex hierarchical structures.

3.1: Simple instructions

To perform a SAMPLE/PRELOAD, EXTEST or an INTEST instruction, the test controller activates the PCB boundary-scan register cells. Normally the INTEST instruction is used at chip level to test the chip functionally. This is performed by shifting test patterns into the boundary-scan cells of the chip, capturing the test responses of the chip and shifting them out for evaluation.

At PCB level a functional test is performed via the edge connectors. Using an HBSC chip together with the INTEST instruction the PCB boundary-scan is selected. This way, via the PCB test access port the edge connector pins are directly accessible using the PCB boundary-scan cells for testing the board functionally.

In the same manner the EXTEST instruction allows the test of the system interconnect (e. g. the backplane). The test patterns are shifted into the PCB boundary-scan and applied to the backplane. The test responses are captured, shifted out and evaluated. This procedure is similar to the test of the board interconnect described in the boundary-scan standard for a chip level EXTEST instruction.

The SAMPLE/PRELOAD instruction allows snapshots of the signals at the inputs and outputs of the system under consideration during normal operation. At board level the HBSC chip selects the PCB boundary-scan cells for a snapshot.

Additionally the BYPASS mode is also selectable at PCB level. In the system of figure 3 this can be used for deselecting a board from the system level test access.

The IEEE 1149.1 standard proposes an optional IDCODE instruction. Using this instruction the correct component mounting after fabrication or the actual configuration for the maintenance of an assembled system can be checked. To be compatible with the standard, the HBSC chip allows an identification code of the whole board to be read out.

3.2: The self test instruction

The self test of the whole PCB is activated by using the RUNBIST instruction in a identical manner to a chip self test. The necessary knowledge to perform the self test is stored in the PCB and leads to an autonomous test process. It is executed by the HBSC chip and proceeds as follows. First a PRELOAD instruction for each chip is utilized to

initialize the boundary-scan registers (i. e. the boundary-scan registers of all chips and the PCB boundary-scan register) with the first interconnect test pattern. Then, the EXTEST instruction is performed on each chip, which runs the interconnect test of the PCB. The test response of each input pad of the internal chips as well as of each PCB output are compacted to derive a signature. A detailed description of the interconnect test procedure is given in the next section. After the interconnect test the self test of each chip is activated by sending the RUNBIST instruction. The test responses of each chip are shifted out and are compressed to obtain a signature. The result of the RUNBIST instruction is the overall signature of the PCB self test. Additionally each chip signature as well as the interconnect signature are compared with the expected values which are stored on board. This allows to append an identification of the faulty component, if there is any, to the RUNBIST result. In contrast to the ERRORCODE instruction, to be described in the following, this identification is restricted to the actual hierarchy level.

3.3: Diagnosis instructions

To obtain diagnosis abilities the HBSC chip offers two additional instructions not described in the boundary-scan standard. The first instruction "DEBUG" allows a manual step by step test of the whole system for the maintenance engineer, the other instruction "ERRORCODE" automatically identifies a faulty component.

Activating the DEBUG instruction, the hierarchical structure is dissolved and each chip is accessed separately, e. g. the board level TAP signals are directly accessible using the PCB test access board. This allows to check each chip on board separately. Since the hierarchical concept may be used to combine PCBs to get larger self testable units, the maintenance engineer can apply this instruction from the highest to the lowest level of hierarchy always in the direction of the faulty component until the smallest replaceable unit is identified. The result is a scan chain from the highest TAP down to the defect component.

Besides the manual instruction DEBUG, the HBSC chip offers the possibility to read out an identification string of the faulty component. This could not be the IDCODE string because identical chips or boards can be used within the same system. So the application of an IDCODE instruction can result in the same data string for different components. Hence we use an identification which is computed by appending the positions within the test access port scan chain of the component and the HBSC chips in higher levels.

To describe the position of a component within an arbitrary hierarchical architecture, an identification string of arbitrary length is necessary. This string includes for each hierarchy level a code for the sub-module containing the defect component and for the lowest level the code of the defect component itself. While designing the HBSC chip it was not possible to determine the length of this string because it will change from usage to usage. So the ERRORCODE instruction does not fully conform to the boundary-scan standard which prescribes a fix length for the result of each instruction. However the ERRORCODE instruction provides a useful feature for in-field tests.

4: The interconnect test

The RUNBIST instruction requires a self test of each chip as well as an interconnect test. The interconnect test is executed by the HBSC chip. It is performed by a standard counting sequence approach [JaYa89]. The sequence vectors are applied to the output BTDO (see figure 2). Since in this mode the HBSC chip directly connects its BTDI input to its CTDO output the PCB boundary-scan cells as well as the boundary-scan cells of each chip on board are selected. Proceeding this way only "output" cells (all edge connector inputs and all chips' outputs) are filled with test patterns. The "input" cells (all edge connector outputs and all chips' inputs) are not involved in this test phase and set to a constant logical value.

If busses have been used on the PCB special care has to be taken in order to avoid bus collisions. It has to be guaranteed, that for each line of a bus only one output driver is active at the same time, i.e. the corresponding enable cell is set to "1". Hence to achieve a complete interconnect test the counting sequence has to be applied separately in different rounds for each set of incompatible output cells – cells which must not be activated at the same time – of each bus line. These sets are determined by usual test scheduling techniques [CrKS88].

The necessary topological information (ordering of input, output and enable cells in the scan chain) and the compatibility relationships between bus enable cells are stored in the PROM as described in the next section.

The pattern generation hardware for the interconnect test is shown in figure 4. The position counter contains the number of the actual cell in the scan chain (PCB boundary-scan and chip boundary-scan chain). By addressing the PROM the classification of the actual cell (input, output, enable) and the compatibility code in case of an enable cell is derived. For each output cell a vector bit out of the counting sequence generator is applied to BTDO. For enable cells the compatibility code is compared with the actual value of the round counter and decides whether to activate the enable cell or not.

The test response of the board interconnect is the contents of the input cells in the scan chain. These values are compressed in a signature analyzer. It is built using a standard linear feedback shift register with a primitive polynomial.

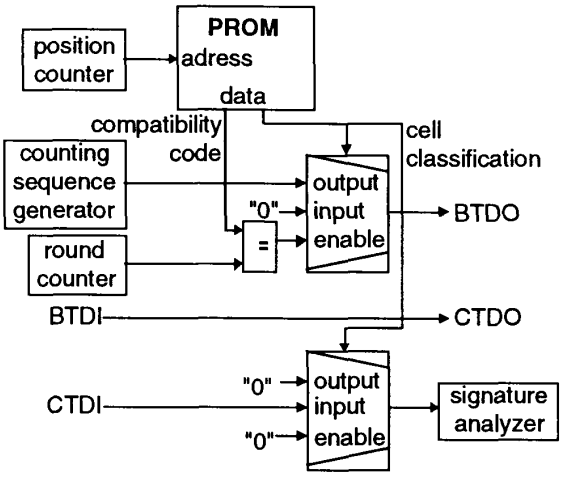

Figure 4: Pattern generation hardware

5: The PROM contents

The HBSC chip is personalized by a PROM to its test environment. The first part of the EPROM contains the identification code for the board. It is used for the IDCODE instruction. The chip relevant information in the PROM comprises:

- the encoding of the SAMPLE/PRELOAD instructions
- the encoding of the RUNBIST instructions
- the maximal number of clock cycles for the self tests
- the expected signatures of each chip

For the interconnect test the PROM includes :

- the classification code for each boundary-scan register cell (output, enable, input)
- the expected signature

To facilitate the EPROM programming, a program has been implemented, which compiles all the required information from a standard EDIF netlist of the PCB and a description of the test capabilities of each chip in BSDL [PaOr91]. The program extracts from the EDIF netlist the ordering of all chips within the chip level TAP chain. This information determines the order of the PROM contents.

To incorporate the PCB boundary-scan cells the drivers at the edge connectors of the PCB are manipulated within the EDIF file. Moreover, the program integrates the HBSC chip automatically into the EDIF netlist.

The BSDL files contain for each chip the SAMPLE/ PRELOAD and the RUNBIST instruction encodings. Furthermore with the RUNBIST instruction the self test length and the expected signature indicated in the BSDL files are directly stored in the PROM.

A BSDL file also describes the boundary-scan register of a chip divided in input, output and enable cells. This information is used to compute the cell classification data. A scheduling program uses the EDIF PCB interconnect description and this classification data to compute the compatibility codes.

Whereas the expected signatures of each chip self test are directly extracted from the corresponding BSDL file the resulting signature from the PCB interconnect test has to be computed by logic simulation. The simulator uses the EDIF file of the board as well as a functional model of the HBSC chip.

Finally, a description of the whole board with HBSC chip is created in the BSDL language.

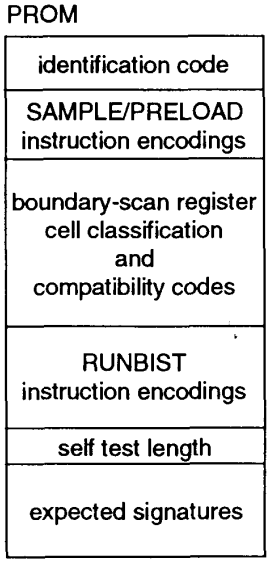

Figure 5: The PROM contents

6: Implementation of the HBSC chip

The HBSC chip architecture is given in figure 6. To provide a standard interface, the chip must contain a TAP interface, a TAP controller and the required registers. The on-chip controller manages the execution of the PCB self test. The necessary knowledge to perform the self test is given in an external EPROM.

To test the HBSC chip itself, self test hardware is integrated using the methods described in [HaKr92].

The HBSC controller has been designed as an ASIC in a 1.5µ CMOS technology with the design framework CADENCE. The prototype implementation requires an external EPROM to store the necessary self test information. Later designs will contain on-chip storage facilities.

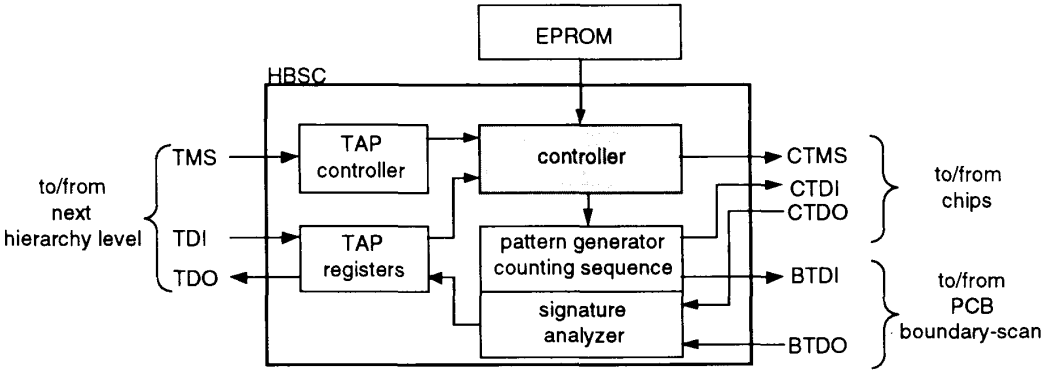

Figure 6: Architecture of the HBSC chip

7: Summary and conclusions

Provided that all chips on a PCB have been designed according to the IEEE 1149.1 standard, the proposed hierarchical boundary-scan architecture, using the HBSC chip

- leads to PCBs, containing a complete self test according to IEEE 1149.1,
- requires only a single test controller for each level of hierarchy,
- is automatically derived from a given PCB design, including the necessary EPROM contents for each PCB,
- leads to a true and autonomous self test since all necessary information for the PCB interconnect test as well as for the self test activation of all chips on the board are stored on the PCB itself,
- allows a fully hierarchical self test by adding one HBSC controller at each PCB hierarchy level and
- supports different diagnosis features to identify faulty components, which have to be replaced.

The hierarchical boundary-scan architecture together with our HBSC chip greatly enhances the ease of testing and maintaining even highly complex systems.

Acknowledgement

We like to thank R. Kumar for proofreading the paper, Peter Lang for his support in implementing the HBSC chip and the EUROCHIP project for providing fabrication facilities.

References

[CrKS88] Craig, G. L.; Kime, C. R.; Saluja, K. K.: Test Scheduling and Control for VLSI Built-In Self-Test. IEEE Transactions on Computers, Vol. C-37, No. 9, Sept. 1988, pp. 1099-1109

[HaKr92] Haberl, O. F.; Kropf, Th.: A Methodology for the Insertion of a Hierarchical and Boundary-Scan Compatible Self Test. Proc. 10th IEEE VLSI Test Symposium, April 1992

[IEEE90] IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std 1149.1-1990, May 21, 1990

[JaYa89] Jarwala, N.; Yau, C. W.: A New Framework for Analyzing Test Generation and Diagnosis Algorithms for Wiring Interconnects. Proc. IEEE International Test Conference, Aug. 1989, pp. 63-70

[JaYa91] Jarwala, N.; Yau, C. W.: The Boundary-Scan Master: Architecture and Implementation. 2nd European Test Conference, April 10-12, 1991, pp. 1-10

[LiBr89] Lien, J.-C.; Breuer, M. A.: A Universal Test and Maintenance Controller for Modules and Boards. IEEE Transactions on Industrial Electronics, Vol. 36, No. 2, May 1989, pp. 231-240

[Maie90] Maierhofer, J.: Hierarchical Self-Test Concept based on the JTAG Standard. Proc. IEEE International Test Conference, Sept. 1990, pp. 127-134

[PaOr91] Parker, K. P.; Oresjo, S.: A Language for Describing Boundary Scan Devices. Journal of Electronic Testing Theory and Application (JETTA), Vol. 2, No. 1, March 1991, pp. 43-75