ETP, the embedded core test processor

Kim Petersén

Malardalen University
dept. IEl Box 883
721 23 Vasteras Sweden
Email: kim.petersen@mdh.se

Bengt Magnhagen

School of Engineering Jonkoping University
Box 1026 551 11 Jonkoping Sweden
Email: bengt.magnhagen@ing.hj.se

key words: Intelligent Property, System On Chip, BIST, Embedded Test Processor.

ABSTRACT

This paper propose a hierarchical and systematic DFT architecture for complete self test of System On Chip (SOC) and MCM designs. The DFT architecture uses an Embedded Test Processor (ETP) to manage the tests, instead of an external tester. The basic idea is to utilize BIST internal core logic and Boundary Scan (BST) based test of the interconnections in between the cores. The architecture supports production test as well as maintenance test.

1. INTRODUCTION

To shorten the time to market (TTM), mega-gate ASIC designers have the opportunity to use predeveloped functional modules, today available on the market, called Intelligent Properties (IPs). This technique can be compared to the design of a MCM (Multi Chip Module) and a PBA (Printed Board Assembly), from now on called a board. Here the predeveloped functional modules are catalog ICs, in combination with PLD (programmable modules) and ASIC (customized modules).
This paper proposes a versatile DFT architecture [4] [11] to support complete self test of mega-gate ASICs and boards. The DFT architecture involves an automatic and independent two step implementation procedure:

- Implementation of the DFT hardware structure (as part of the functional development).
- Generation of the test vectors.

The DFT architecture proposed can be utilized as part of a system test architecture [3] [7].

2. TO HANDLE THE GROWING COMPLEXITY

Test of ASICs with full or partial Scan (virtual needles) have been used for a long time. With ASIC designs in the size of several mega-gates, will it still be possible to use Scan as the number one method for test ? The answer is no ! The use of Scan is based on development at the gate level. But the complex mega-gate ASICs designs must use a mixture of IPs and customized ASICs (to handle the complexity and the time to market demand).

3. IP WITH BIST

IPs are normally the property of the creator and the logic structure of IPs can be foreseen to be unknown by the user. One reason to use IPs is to save time. Maximum time saved is gained if IPs are designed for testability, preferably in the form of BIST. Then the user does not care about the physical realisation.
Mega-gate ASICs comprising IPs are becoming very similar to boards. So the question is if it is possible to use the same method for test of mega-gate ASICs as for boards ? The answer is yes, so we propose a DFT architecture that can be used for boards, mega-gate ASICs and IPs.

4. SCAN TECHNIQUE STILL TO BE USED

Why are Scan and BST so popular ? One reason is that the implementation can be done in two independent steps. **Step 1 (design independent):** The physical test structure can be inserted for any type of digital design, as long as a few basic design rules are followed. **Step 2 (design dependent):** Test vectors are created concurrently with the design work.
The proposed test structure support this two step implementation procedure. It supports total and hierarchical BIST, performed with the ETP. Complete BIST makes it possible to reuse the test at any time during the life cycle time. It is combination of already known test methods and technologies, it is how the parts are combined that opens the door for a test method as ”one type fits all levels”.

5. THE DFT ARCHITECTURE

It divides the design, from test point of view, into an area of interconnections and a number of functional blocks and it is composed of the following main parts as (also see fig. 1):

- An embedded tester which can be supplied in form of an IP module (with its own BIST).
- One standard 1149.1 [1] [9] Boundary Scan (BSN) chain around the boundary of the design. The BSN chain is connected to the embedded tester.
- One or more internal Boundary Scan (IBS) chain(s) connected to the ETP in the embedded tester.

All these parts are inserted as part of a synthesis operation. Both the BSN and IBS chains are composed of 1149.1 BST cells.

The embedded tester is composed of the following main parts as:

- One Boundary Scan Interface (BSI) including the 1149.1 TAP controller, the bypass register, the instruction register, the data registers and the multiplexer for the output TDO.
- One Embedded Test Processor (ETP). It carries out all test stimulus used for test of interconnections and performs all test analyses. It is a compact µPr [8] optimised for embedded usage.

First after the development of the design is ready, the program for the ETP is created. The program defines the test operations in the same way as the test vectors when an external tester is used. The program can be created in a similar way as ATPG.

The functional blocks can be a mixture of IPs and user specific. To keep the size of the test program in the ETP as small as possible, each block of logic shall have its own BIST implemented. The internal BSN chains can only be accessed through the ETP.

Figure 1 shows an example of the DFT architecture when four functional blocks and two IBS chains are used.

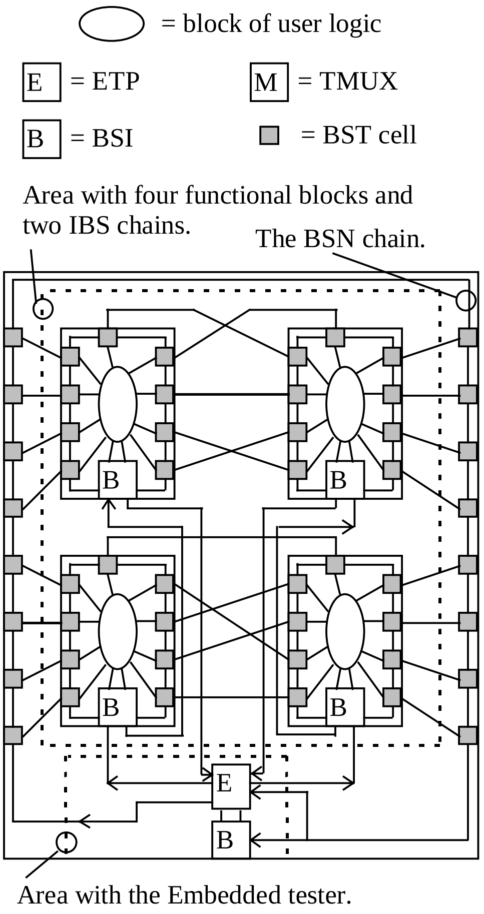

Fig. 1. The DFT architecture.

One or more of the four functional blocks can itself consist of a DFT architecture as shown in figure 1. This implies that the DFT architecture can be hierarchical with for example two or more ETPs.

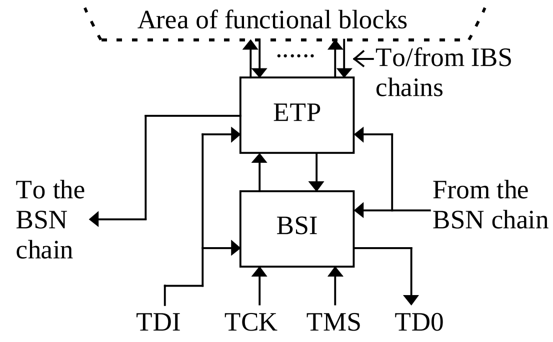

Fig. 2. The embedded tester.

6. The ETP

It is composed of the following main parts as (see fig. 3):

- One multiplexer (TMUX) which is incorporated into the 1149.1 BSN chain.
- A BSN interface for the BSN chain.
- IBS interfaces for all the IBS chains.
- A compact µPr [8].

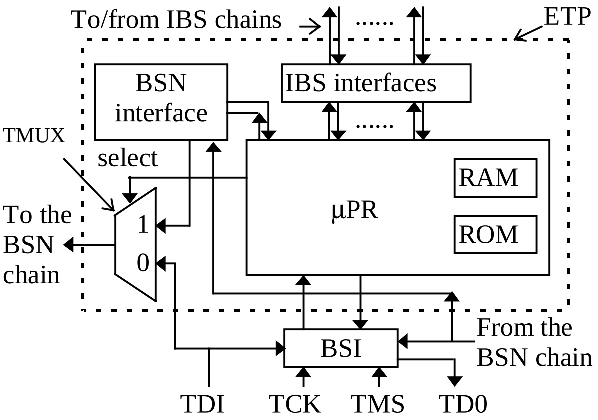

Fig. 3. The ETP.

The TMUX is a standard 2 to 1 multiplexer. The multiplexer is connected into the BSN chain as described in figure 3. This multiplexer is the only part of the DFT architecture that introduce a small modification, in terms of logic, of the standard 1149.1 BST architecture.

The TMUX makes it possible for the ETP to carry out test of all the interconnections between the BSN chain, surrounding the boundary of the total design, and the IBS chain(s). As long as the ETP is not carrying out any test, the 1149.1 input TDI is connected to the output of the TMUX (select=0). As long as the ETP is carrying out test of the interconnections (select=1), the ETP can force test patterns into and read the responses from the BSN chain.

The ETP has the capability to:

- Test the interconnections between blocks and between blocks and pads. The fault types tested for are opens, bridges and stuck-at.
- Activate BIST function(s) embedded in the functional blocks and read back the test result(s) or the final signature(s).
- For functional blocks without BIST. It is possible to use the ETP to perform a total test of such blocks, the number of such functional block should be limited.

We have chosen that the ETP uses pseudorandom test vectors [2] [6] [10] and some repetitive deterministic test vectors to test the interconnections.

It is important to keep the total size of the ETP small. This implies that the total size of the program for the µPr should be limited to a few kwords. For this reason, the BSN interface and the IBS interfaces are introduced to:

- Keep the total size of the µPr program within an acceptable number of kwords, this without losing the flexibility of the DFT architecture.
- Enable concurrent test activities (with just one µPr).

The IBS interfaces includes both setable LFSRs/MISRs and small controllers to handle the different TAP controllers in each BSI in the IBS chains. With the small controllers, the handing of the TAP controllers, placed in the IBS chains, are reduced into 8 instructions in the µPr. Several of the sequences carried out on demand by the 8 instructions are very similar to each other or simple. The eight different sequences, where the first three points cover six of the eight sequences, are:

- Change state from ”Run_test Idle” to ”Shift-DR(IR)”, perform the number of shifts defined and after all shifts carried out continues to state ”Pause-DR(IR)”.
- Change state from ”Pause-DR(IR)” to ”Shift-DR(IR)”, perform the number of shifts defined and after all shifts carried out continues to state ”Pause-DR(IR)”.
- Change state from ”Pause-DR(IR)” to ”Shift-DR(IR)”.
- Change state from ”Test-Logic Reset” to ”Run_test Idle”.
- Set the TAP controller in the home state ”Test-Logic Reset”.

The small controllers in the IBS interfaces must also have a setable count down counter, this counter is used to keep track of the number of bits to shift when in state ” Shift-DR(IR)”. The counter is loaded by the µPr as part of the setup before activating four of the eight commands.

A setable LFSR/MISR is also needed in each IBS interface. It is used, when in state ” Shift-DR(IR)”, to define the sequence of patterns shifted into a IBS chain. The function of the LFSR/MISR is defined by the µPr as part of the setup before activating four of the eight commands. When used as a pattern generator: The parameters set by the µPr are the start vector and the algorithm to use. The algorithm set can create a pseudo-random pattern or a deterministic pattern as for example ”000...”, ”111...”, ”1010...” or ”1100...” a.s.o. To improve the handling of random pattern resistant faults, bit flipping capability can be built in [5]. When used as a signature generator: The parameter set by the µPr is the start value. By using the IBS interface in the ETP: To define each operation to carry out at a IBS chain, it is only necessary to save a maximum of 4 words as part of the program for the µPr.

The BSN interface, it is designed and used in the same way as an IBS interface. It always consists of only one channel.

7. EXAMPLE OF A TEST EXECUTION

Lets use the design in figure 1 to describe one (of many) possible test procedure to program the ETP to carry out. A test procedure can consist of 5 points executed in the order as follows. For each test point executed, the ETP ends the test by saving the test result (as a signature) in a readable register in the IBS interface (part of the ETP).

1. Command from external: Through the BSI interface in the ETP, set up a 1149.1 RUNBIST command to the BIST logic in the ETP. The ETP tests itself, after ready it waits for next possible command.
2. Command from external: Through the BSI interface in the ETP, set up another 1149.1 RUNBIST command that activates the ETP to test the entire design. The ETP translates the received command for BIST into a set of software controlled test activities and test commands. The tests carried out are as follows.
3. The ETP tests that the BSN and IBS chains all are operational. if so continue to next point, else halt.
4. By using the BSN and IBS chains: The ETP tests all the interconnections between the IBS chains and between the IBS chains and the BSN chain. This test can use a mixture of pseudorandom and deterministic test patterns. If no faults are detected, continue to next point.
5. Time to test all the functional blocks. For the functional blocks with BIST: The program in the ETP defines when the different BISTs shall be carried out and how many BISTs that shall be activated to run in parallel. Each BIST is activated with a RUNBIST command through the IBS chains. After the BIST finished, the ETP reads back the result of each test, by using the IBS chains.

8. THE IMPLEMENTATION PROCEDURE

The procedure to implement the total DFT architecture can be divided into two main steps:

1. Step 1: Implementation of the hardware structure. It is design independent, ie. it is generic. The operation

can be compared with the insertion of scan chains when using full scan.
2. Step 2: Creation of the test vectors and scheduling of how the tests are carried out. It is design dependent. It can be compared to ATPG when using an external tester.

Step 1 is both performed before step 2 and independently of step 2.

Step 1: It is performed as part of the design work. For each user created functional block, surround it with a BST chain (to be part of a IBS chain) and a BSI. Normally should also an embedded BIST be created for the functional block. For IPs used, they should already have the BST architecture around the boundary and an embedded BIST. For certain types of regular functional blocks as memories and datapaths, the ETP can be used instead of an embedded BIST. Define the number of IBS chains to use, the number of functional blocks in each chain and the order the functional blocks are inserted in each IBS chain. The ETP is instansiated in the design as a firm IP.

Step 2: It is performed after the design work is ready. Now is the order of test operations scheduled. After some setup data has been filled in by the person running the tools used, the program for the μPr, in the ETP, is created automatically. In the case where the ETP shall carry out the BIST for one or more functional blocks. It may be necessary to write this part(s) of the test program by hand.

Since ”step 2” only affects the contents in a ROM, it is possible to wait after the layout is ready, before creating the final version of the part of the test program used to carry out the test of all the interconnections. The opportunity to take the final layout into consideration makes it possible to sometimes improve the level of fault coverage.

Compared to normal operation, execution of a test normally creates much higher activities (logical switching) in the design tested [2]. Scheduling of the Power dissipation must be a part of the ”step 2” creation of the test program to be used by the ETP, this to avoid hot spots and other possible damages and problems during the test execution.

9. FUTURE WORK

Test the DFT architecture on a design,
this to investigate how it performs. Implement setable pseudorandom generators and response analysers, this to investigate the area overhead needed to incorporate such functions into an ETP.

10. SUMMARY

The article has proposed a self test approach based on a systematic DFT architecture which uses an ETP as a master. The DFT architecture is created with the purpose to be useful to test mega gate ASIC designs and boards. The DFT architecture supports both production test and maintenance test. The proposed DFT architecture only introduce a small modification, in terms of logic, of the 1149.1 boundary scan standard. The hardware part of the DFT architecture can be implemented before the software part is defined. The hardware part can be implemented as part of the normal synthesis. The software part, which defines the behaviour of the ETP, it only affects the contents in a ROM. This implies that the number and order of test operations and also the pseudorandom stimuli generation algorithms, which all are software controlled, can be defined without affecting the hardware part of the DFT architecture.

11. ACKNOWLEDGEMENT

This work is supported by the Swedish National Board for Industrial And Technical Development (NUTEK).

12. REFERENCES

[1] IEEE Std 1149.1-1990; IEEE Standard Test Access Port and Boundary-Scan Architecture.

[2] Miron Abramovici, Melvin A Breuer, Arthur D. Friedman; Digital Systems TESTING and Testable DESIGN. Computer Science Press.

[3] Jan Håkegård and Zebo Peng: ”A Synthesis Approach for a Board-Level Test Controller”, International Test Synthesis Workshop 1996.

[4] Kim Petersén, Bengt Magnhagen and Henric Lindén: ”Embedded Test Processors (ETP)”. Baltic Electronic Conference 1996.

[5] Hans- Joachim Wunderlich and Gundolf Kiefer: ”Scan-based BIST with complete fault coverage and low hardware overhead”. European Test Workshop 1996.

[6] P. Nagvajara, M.G Karpovsky and L. B. Levitin: ”Pseudorandom Testing for Boundary-Scan Design with Built-In Self-Test”. IEEE Design & Test of Computers September 1991, pages 58-65.

[7] Rodham E. Tulloss: ”Functional Evaluation Model for the IEEE Std 1149.5 Module Test & Maintenance Bus with Observation on Introducation, Probability Cost and Non-Test Usage”. Distributed at IEEE BIST/DFT Workshop Mars 1995.

[8] Nordic VLSI AS, Norway: Are your future system needs a hard nut to crack ?

[9] Harry Bleeker, Peter van den Eijnden and Frans de Jong: Boundary-Scan Test A Practical Approach. Kluwer Academic Publisher 1993
ISBN 0-7923-9296-5.

[10] Paul H. Bardell, William H. McAnney and Jacob Savir: Built-In Test for VLSI Pseudorandom Techniques. John Wiley and Sons 1987 ISBN 0-471-62463-2.

[11] Kim Petersén and Bengt Magnhagen”A DFT architecture for total self test of mega-gate designs”. European Test Workshop 1997.