A DFT architecture for total BIST of mega-gate designs

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, BIST, Embedded Test Processor.

ABSTRACT

This document propose and motivate a systematic DFT architecture that can be useful in very large ASIC or MCM designs to handle total BIST. The architecture uses BIST for test of the internal logic, boundary scan based pseudorandom BIST for test of the interconnections and an Embedded Test Processor (ETP) to manage all the different test operations.

1. INTRODUCTION

At ITC 1996 the vision ”year 2001 will digital ASIC designs be of the size 26 million transistors” was presented. We propose a systematic DFT architecture useful for test such mega-gate design by a self test approach. The DFT architecture is composed of already known DFT methods and has an architecture similar to a boundary scan (BSN) based MCM. It uses an embedded test processor (ETP) [4] as a test master. The ETP is used to translate an external command for BIST into a set of software defined test commands and test operations. The test program is saved in an embedded ROM. To keep the physical size of the ETP small, the size of the ROM must be limited. To be able to keep the ROM small, a few basic design rules are introduced. We describe the design rules and also discuss and motivate why the design rules are introduced.
Finally we also show that the proposed DFT architecture also can be used to fully test MCMs.

2. MOTIVATION

The design work of an ASIC includes implementation, verification and test creation. Implementation signifies the work to create the logical functions as stated in a specification. Verification signifies the work to verify that the implemented logic fulfils the specification in terms of functionality and timing. Test creation signifies the work to insert the DFT architecture and to create the test vectors.

An ASIC circuit of the size 26 million transistor has a complexity similar to a MCM. To be able to fulfil the demand ”time to market”, such an ASIC design needs to contain a certain amount of already designed logical modules. This type of logical module already exists and are called intelligent property (IP). During 1995, about 11% of the bigger ASIC circuits comprise blocks of type IP. Dataquest has predicted that during 1997 this value will increase to about 21%. With this trend, the design of complex integrated circuits are becoming more and more similar to todays design of MCMs, where IP at the ASIC level can be compared to bare dies at the MCM level. The conclusion is that the use of IPs is increasing because it is a way to shorten the implementation part of the design work. *How will the use of IPs affect the verification and test creation part of the design work* ?

It is very time consuming to functionally correct a design developed by other persons. Since it can be expected that the producer and the user of an IP normally is not the same person, the logical function of an IP must be fully verified already when being implemented.

Timing verification is dependent on accurate timing information. The size of the silicon processes, used at production, are decreasing all the time and today a lot of production is performed in the process size 0,35 µm. Already at this process size, most of the timing is defined by the interconnections and not by the logic itself. To be able to extract the timing information needed to verify the timing for deep submicron designs, it is necessary to perform a layout. It takes a lot of computational power to perform a complete layout of a mega-gate design, i.e. it is a time consuming activity. A guard band must also be introduced if the extracted timing information, from the physical layout, has an uncertainty. The size of the guard band is dependent on the accuracy of the extracted timing information. To get maximum possible performance (i.e. maximum clock speed) , a guard band must be avoided or at least minimised. A way to avoid the time consuming layouts and minimise the size of a guard band, is to use IPs that has predefined layouts (hard macros). IPs with predefined layouts have the drawback that they must be created specifically for each process used and they will also give an increased design size. If three different shapes are available for IPs with predefined layout, the extra physical area introduced is reduced. The advantage with using IPs with predefined shapes are: 1) High level simulation models for the IPs, can also be used during timing verification of the design (much shorter simulation times). 2) The extraction of the timing information is only needed for the interconnections and the user created logical blocks. This will speed up the layout work, this since all the IPs with a predefined layout can be handled as empty black boxes. The advantages with a predefined layout seems to over ride the drawbacks.

Who can be responsible for test creation of IPs ? For an IP that has a predefined layout, the producer must be responsible for the test creation. No matter if the layout is predefined or not, a description of the implementation itself is probably not public information. If an IP without a predefined layout is used and the user has the time to do the test creation, there is always a risk that the desired fault coverage isn't reached. Only if the producer himself does the test creation, the level of fault coverage and performance can be optimised.

But what DFT method should be used for IPs ? BIST has the advantage that the external interaction needed, to execute the test, is very limited. The drawback is that the IP's own BIST only test the logic within the logical block but not the interconnections between blocks. SCAN has the advantage, if used for all the logical blocks, that it tests both the logical blocks and the interconnections between the blocks. SCAN can't be used for high density logical blocks [2] such as memories and also has the drawback that the external interaction needed, to execute the test, is high. A mega-gate design would probably include some high density logical blocks. For a mega-gate design: A DFT architecture that use BIST for test of logic and boundary scan based BIST for test of interconnections seems possible to use.

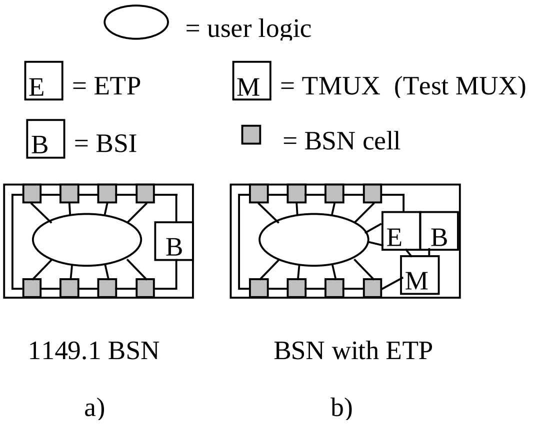
3. THE DFT ARCHITECTURE

The DFT architecture consist of a hardware part (as shown in figure 2 and a software part. The software part is the program that controls the ETP. The DFT architecture divides the design, from test point of view, into an area of interconnections and a number of blocks of logic. The DFT architecture is composed of the following main parts as:

- A standard 1149.1 [1] Boundary Scan (BSN) around the border of the design as indicated in fig 1a.
- One multiplexer (TMUX) which is incorporated into the 1149.1 BSN chain as indicated in fig 1b.
- One Embedded Test Processor (ETP) [4] which is included as indicated in figure 1b.
- One or more internal BSN chains connected to the ETP as indicated in figure 2.

The BSN interface (BSI) includes the TAP controller, the bypass register, the instruction register, data registers and the multiplexer for the output TDO.

The area "user logic" in fig. 1, consists of a number of blocks of logic and the interconnections between the blocks. The blocks of logic 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.



1149.1 BSN a)

BSN with ETP b)

Fig. 1. Standard 1149.1 BSN and BSN with ETP.

The ETP is connected between the internal BSN chains and the BSI. To activate BIST in the blocks of logic, the ETP uses the internal BSN chains. To test the interconnections, the ETP uses all the BSN chains. The ETP is activated, through the BSI, by sending a 1149.1 BSN compatible BIST command to the BSI. The ETP translates the external command for BIST into a set of software controlled test activities and test commands. Which test commands and activities the ETP shall carry out can be defined after the layout of the design is ready.

Figure 2 shows an example of a design using the DFT architecture and a "user logic" area consisting of four logical blocks and two internal BSN chains.

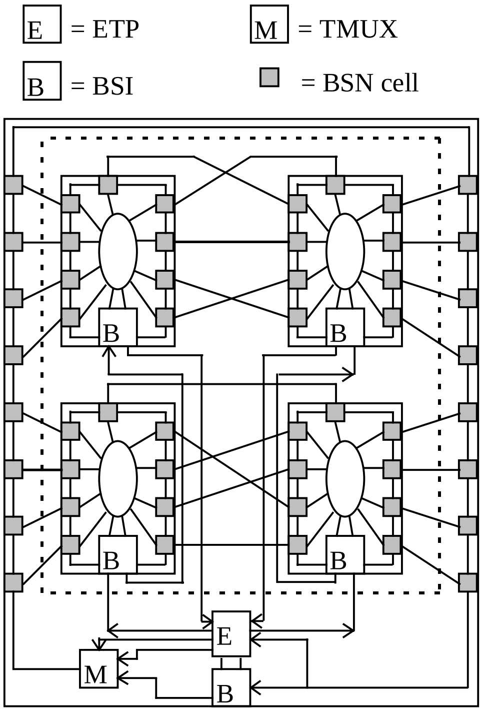


Fig. 2. The DFT architecture with BSN and ETP.

The ETP has the capability to:

- Test the interconnections between blocks and between blocks and pads. The fault types tested for are opens, shorts and stuck-at.
- Activate BIST function(s) embedded in the blocks and read back the test result(s) or the final signature(s).

We have chosen that the ETP uses pseudorandom test vectors to test the interconnections. One or more of the four blocks shown in figure 2 can itself consist of a DFT architecture as shown in figure 2. This implies that the DFT architecture can be hierarchical with for example two ETPs as shown in figure 3 where ETP E1 is a master over E2.

If the DFT architecture shall be used in an MCM: We propose that the ETP, BSI and TMUX are merged into one IC placed on the substrate. We also propose that the BSN cells, as small integrated circuits, are placed in conjunction with all the bonding pads around the border of the substrate. The BSN cell circuits should be placed on the substrate or, if possible, embedded into the substrate. The blocks of logic shall be designed in the same way as when used in a mega-gate ASIC design.

The DFT architecture can be connected to a hierarchical test architecture as [3] or [7].

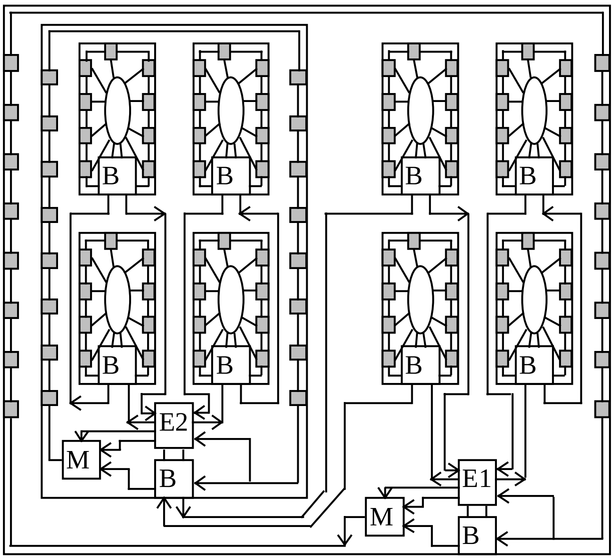


Fig. 3. The DFT architecture with two ETPs, E1 and E2.

3.1. The TMUX

It is a standard 2 to 1 multiplexer. The multiplexer is connected into the standard 1149.1 BSN chain and connected to the ETP as described in figure 4. This multiplexer is the only part of the DFT architecture that introduce a small modification, in terms of logic, of the standard 1149.1 BSN architecture.

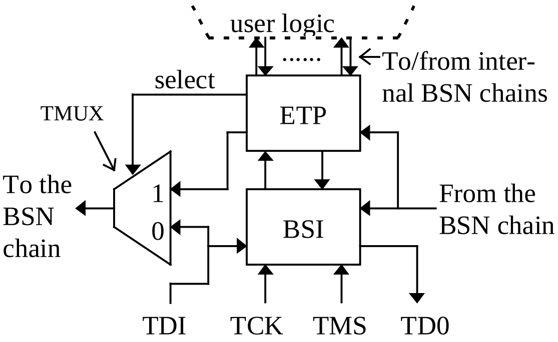


Fig. 4. The connection of the TMUX.

The TMUX is introduced to make it possible for the ETP to perform test of all interconnections between the 1149.1 BSN chain, surrounding the border of the design, and the internal BSN chains. As long as the ETP is not carrying out test of interconnections, the input TDI is connected to the output of the TMUX (select=0). As long as the ETP is carrying out test of interconnections (select=1), the ETP can force test patterns into the 1149.1 BSN chain.

3.2. The ETP

To minimise the area overhead for the ETP, it should be a compact µPr optimised for embedded use. The ETP should also be easy to test through the BSI. There exists already, on the market, µPrs optimised for embedded usage. An example of such a µPr is the µRISC [8] from Nordic VLSI AS. The µRISC is designed as 4 bit slices, where a slice can be used to create a 4, 8, 12 or 16 bit µPr. The µRISC has a Harvard architecture and has a total of 31 instructions where most of the instructions execute in one clock cycle. The different blocks in the µRISC has embedded BIST functions. The µRISC is designed to work with an operation frequency in the range 0 to 40 MHz. The total number of general purpose registers are 16 and it can use 64 I/O addresses. The µRISC with an 8 bit ALU core and in a 1 µm process has the physical size as defined in the table to follow. The width of the ROM is 16 bits and the width of the RAM is 8 bits.

ROM (words)	RAM (bytes)	Area (mm^2)
1024	64	3.3
2048	128	4.4
4096	192	6.2

Table. 1. Sizes of a µRISC with different ROM and RAM sizes

Compared to a standard µPr, the ETP should also have the following functions:

- TAP interface(s) for the internal BSN chain(s).
- Generate pseudorandom test patterns.
- Both compact the test responses into one or more signatures and analyse the generated signatures.

To minimise the time needed to execute the total BIST of the design, the ETP should have one TAP interface for each internal BSN chain and each TAP interface should include one pseudorandom generator. Each TAP interface can be setup with the parameters:

- The length of the test operation, this in number of clock cycles on its own TCK.
- The start pattern in the pseudorandom generator.
- The algorithm to use in the pseudorandom generator.

The ETP fetches all parameters required from the internal ROM. After all settings are ready: Without any interaction from the ETP, the TAP interface carries out the set operation.

At end of a reset, the ETP automatically tests its own functionality. The result from the test is a signature. This signature can be read from through the BSI.

3.3. The internal BSN chains

The internal BSN chains are composed of chains of 1149.1 BSN cells. They do not include pads for bonding. The standard 1149.1 is selected due to that it is an existing standard that use a serial interface (low area overhead) and it is optimised for test of interconnections.

Independently of the number of internal BSN chains and the number of ETPs, each BSN chains shall always be connected to one ETP.

3.4. The blocks of logic

The stack-at fault model is optimised for single errors. The presence of multiple faults in a test may reduce the total fault coverage of the test program [2]. Since each block of logic has its own BIST function, the probability for multiple faults are reduced. The BIST implemented in each blocks of logic, it can be a SCAN based BIST [5] [6] or any type of BIST. All blocks of logic shall be connected to one of the internal BSN chains, as shown in figure 2. Each block of logic shall have a BSI and be surrounded by a chain of BSN cells. During test, the chain of BSN cells are used to isolate the block of logic from the rest of the design. The BIST shall be possible to activate through the BSI and the test result, or created signature, shall be possible to read through the BSI. The test carried out by the implemented BIST, shall cover everything surrounded by its own chain of BSN cells. To be able to do this, during the execution the implemented BIST function must use its own BSN chain.

4. 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.

5. SUMMARY

The article has proposed a self test approach based on a systematic DFT architecture which uses an embedded test processor as a master. The DFT architecture is created with the purpose to be used to test mega gate ASIC designs and MCMs. 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 embedded test processor, 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.

6. ACKNOWLEDGEMENT

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

7. 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 ?