Efficient Methodology for Boundary Scan Insertion and Pattern Generation for MCM Based Designs

P. Soma Shekar, N.V.AnilKumar
Tessolve Services Pvt. Ltd
soma.shekar@tessolve.com, **anil.nimmagadda@tessolve.com**

Abstract

The standard IEEE 1149.1 poses a challenge when used for MultiChipModule designs. In this paper, a novel and efficient method to insert and test boundary scan circuitry in a MultiChipModule design has been proposed. Our strategy imposes minimal additional hardware during boundary scan insertion, and minimal manual intervention during the boundary scan pattern generation, while complying with the IEEE 1149.1 standard. The proposed method is implemented on a sample MultiChipModule design which consists of all kinds of pins and interconnects between the cores. Simulation results are presented to prove the presented scheme.

Keywords: MCM(MultiChipModule), IEEE, Boundary Scan, TAP(Test Access Port)

1. Introduction

The main goals of the emerging technologies in VLSI are high performance resulting from reduced signal delays, improved signal quality between chips, reduced overall size and reduced number of external components [5] [2]. This everlasting demand for high circuit speed and low die area had lead to the development of Multi-chip-Module (MCM). In high performance circuits, where we need VLSI technology as well as packaging technology has to be improved, MCMs plays a key role. As MCM packaging is a promising technology that provides high chip density and short line length, it reduces propagation delay between chips. As the die to die parasitic capacitance is low, it can significantly improve the performance of products. MCMs provide a compact and light weight alternative to conventional packaging [2].

But this dense packaging complicates manufacturing and testing process. The normal test strategies used for testing a single chip cannot be used for an MCM. We need much better test strategies for testing MCMs[5].

In this paper, we focused mainly on the implementation aspect of boundary scan architecture for a MCM design. The boundary scan architecture for any design has to comply with IEEE 1149.1 standard [1] and the proposed scheme complies it with minimal additional hardware and minimal manual intervention for pattern generation, which are used to verify the inserted boundary scan circuitry.

This paper is organized in the following manner. Section 2 contains the boundary scan implementation for an MCM. In section 3, we have presented a way on how to automate the pattern generation for the MCM boundary scan. Section 4 presents the experimental results to prove the presented scheme. Section 5 concludes the paper.

2. Boundary Scan Implementation for a MCM

Boundary scan architecture provides a means to test the interconnects and clusters of logic, memories etc [1] [6]. The boundary scan architecture in a design has to comply with IEEE 1149.1. In this section we are going to propose a new method on how the boundary scan architecture will be implemented on a MCM design, and how it will comply with IEEE 1149.1 standard, with minimal additional hardware when compared to the single chip boundary scan mode.

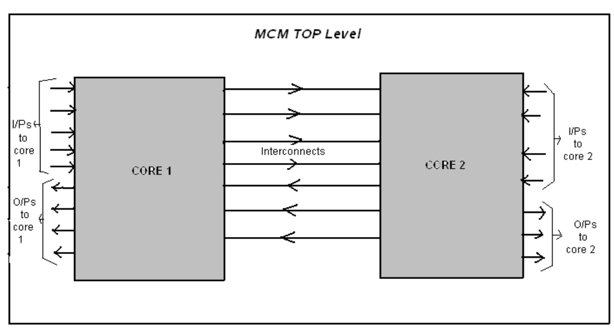

Figure 1. A MCM chip with 2 cores, with interconnects connecting between them

The proposed scheme will be explained by taking an MCM consisting of two cores for simplicity. Let CORE 1 and CORE 2 be the two cores which are having interconnects between them as shown in the figure 1.

Let us assume that the two cores have their individual boundary scan circuitry, designed before they are placed into a MCM environment.

The new scheme presented in this paper proposes how the two cores with existing boundary scan circuitry, will comply with IEEE 1149.1 standard [1] when connected in MCM mode, as shown in figure 2.

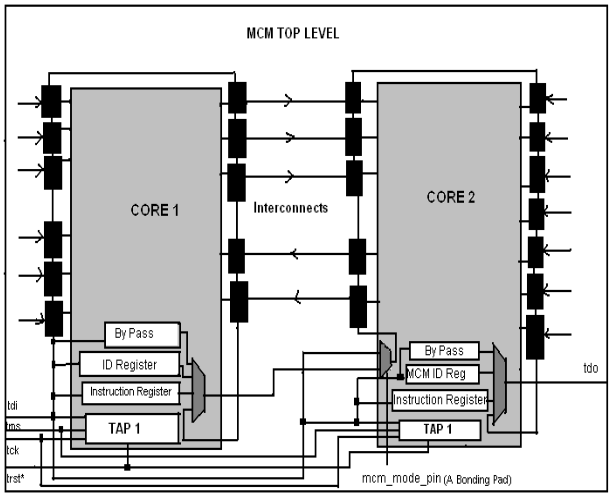

Figure 2. The proposed scheme for boundary scan implementation of MCM chip with 2 cores

The TDI pin of MCM will be connected to the TDI of the first core (i.e., to the inputs of all data and instruction registers), and to the instruction, bypass, and idcode registers of the second core(i.e., to the TAP controller of core 2). The TDO of core1 will be connected to the boundary scan register of core2. From the top level, both the TAPs are connected in parallel with each other, and the boundary scan registers of both the cores are connected in series.

It can be observed from the figure 2, that a mux is placed in the path of boundary scan register of core2. This mux provides an option of deciding the input to the boundary scan register of core2. If the chip is designed, such that it can be used either in MCM or in core2 only mode, then we can use this mux to provide either the TDO of core1 or the external TDI pin as the input to then boundary scan register of core 2, respectively. A bonding pad option can be provided for the select line of mux. This mux is optional, and not needed if it is decided that the chip is going to be used only in MCM

Let us see in detail, that how this method will comply with the standard IEEE 1149.1. An IEEE 1149.1 complaint device should obey the following [3][1] :

a) *Can have only one Test Access Port (TAP):* In this method the TAP which exists between TDI and TDO will be TAP2 of core2. Hence, from top level, only on TAP will be visible.

b) *Shall have a BYPASS register with a length of exactly one bit:* The BYPASS register which is connected between the TDI and TDO will be of core2, which is one bit in length.

c) *Can have an optional ID register with a length of exactly 32 bits:* The ID register which is connected between the TDI and TDO will be of core2, which is 32 bit in length.

d) *Can have an optional TRST* pin that, when activated, resets all the TAP controller logic to the defined 'test logic reset' state (TLR):* The TRST* will be the same for both the TAPs of core1 and core2, which when activated resets both the TAP circuitry.

Other than these rules, the device must have a single BSDL file describing the implementation details. The BSDL file generally will be generated by the boundary scan insertion tools and is not generated manually. This issue of generating a BSDL file will be discussed in the coming section.

3. Pattern Automation

The boundary scan patterns of an individual chip which is given to the tester tests the boundary scan circuitry inserted into the design [1]. But in MCM design along with the above task, the patterns should test the interconnects between the cores. The existing boundary scan tools cannot generate patterns for a MCM design. The problem with tools will be, in handling a design with multiple TAPs, which is one for each core. But generating the patterns manually will not be a good procedure, as it is a very tedious task to do. This section deals with this issue of pattern automation, and suggests a possible solution to minimize the manual intervention during the generation of patterns, required for MCMs boundary scan testing.

This issue will also be explained by taking the same example of two cores in an MCM. While using any of the existing boundary scan tools, if the MCM design be presented to the tool as a single core (ie., with a single TAP) instead of multiple cores, then the problem of multiple TAPs will be solved. In this proposed scheme, as both the TAPs are connected in parallel, we will reconfigure the design as in figure 3, just for the sake of pattern generation.

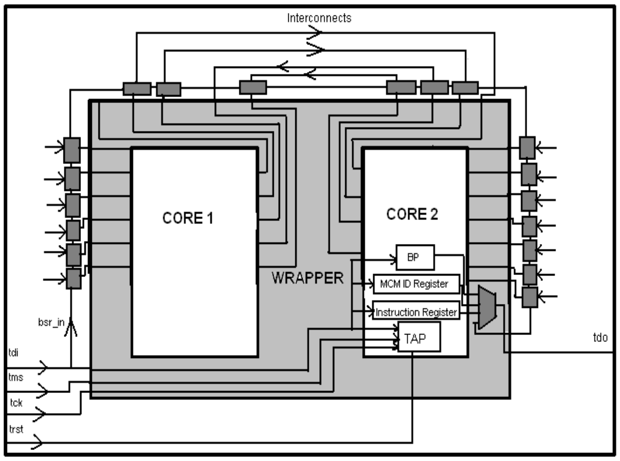

Figure 3. Wrapper built above all the cores of the design

A wrapper will be created on the top of all the cores as shown in the figure 3. The wrapper will be created such that it consists of all the core level ports as its ports, and all the core to core interconnections will be as shown in the figure. This wrapper which is a single core from the top, internally consists of all the MCM cores. All the boundary scan cells inside wrapper will be controlled by the TAP which is present between external TDI and TDO pins, i.e., the TAP of core2 in this case.

The patterns generated by the tool for this design will be able to verify the inserted boundary scan circuitry, but not the core to core interconnects. For the core to core interconnects the test case has to be written manually, which is less tedious task when compared to writing the whole test cases to test the boundary scan circuitry and interconnects, manually. The BSDL file generated by the tool for the wrapper will be sufficient for the MCM also.

4. Experimental Results

For simplicity, assume a MCM with two cores, interconnected to each other as shown in the figure 1. The design consists of all kinds of pins say input, output and bidi pins, and all kinds of interconnects between the two cores.

Boundary scan insertion for each core had been performed separately. Mentors BSDArchitect v8.2007_1.10 was used for this purpose. Both the cores were separately tested using Mentors Modelsim SE-64 at core level by using patterns generated by BSDA tool. After testing separately they were taken into a MCM environment. At MCM level, the changes that are discussed in the section 2 are made (a mux is included into the design of core2, for providing the input to the boundary scan register).

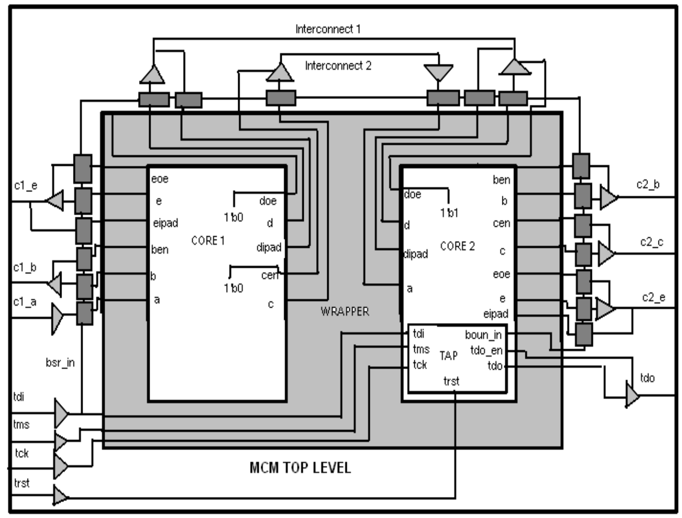

Figure 4. An Example MCM design shown with a wrapper inserted above the cores

For the sake of pattern generation, a wrapper was made around core1 and core2 as shown in the figure 4. This wrapper was given to the Mentors BSDArchitect v8.2007_1.10. The tool generates test bench consisting of patterns and BSDL file. The boundary scan register described in the BSDL file will be as shown in figure 5, which describes the type of the boundary scan cell inserted, its position number with respect to TDO, and the type of pad it is associated with.

```
attribute BOUNDARY_LENGTH of top : entity is 19;
attribute BOUNDARY_REGISTER of top : entity is
 --- num     cell      port    function   safe  [ccell  disval  rslt]
 "0   (    bc_1,     c2_e,    input,    X)," &
 "1   (    bc_1,     c2_e,    output3,   X,    2,     1,    Z)," &
 "2   (    bc_1,      *,     control,   1)," &
 "3   (    bc_1,     c2_c,    output3,   X,    4,     1,    Z)," &
 "4   (    bc_1,      *,     control,   1)," &
 "5   (    bc_1,     c2_b,    output3,   X,    6,     1,    Z)," &
 "6   (    bc_1,      *,     control,   1)," &
 "7   (    bc_1,      *,     internal,  X)," &
 "8   (    bc_1,      *,     internal,  X)," &
 "9   (    bc_1,      *,     internal,  X)," &
 "10  (    bc_1,      *,     internal,  X)," &
 "11  (    bc_1,      *,     internal,  X)," &
 "12  (    bc_1,      *,     internal,  X)," &
 "13  (    bc_1,      *,     control,   1)," &
 "14  (    bc_1,     c1_e,    output3,   X,    13,    1,    Z)," &
 "15  (    bc_1,     c1_e,    input,    X)," &
 "16  (    bc_1,      *,     control,   1)," &
 "17  (    bc_1,     c1_b,    output3,   X,    16,    1,    Z)," &
 "18  (    bc_1,     c1_a,    input,    X)";
```

Figure 5. BS Register description of the MCM design in BSDL file.

A few test cases were written, which tests the core to core interconnects. Then the MCM design is taken into a simulator along with the test bench. The simulator used was Mentors Modelsim SE-64. The results for the simulation were as shown in the figure 6, which proves that the patterns are passing.

```
11110011111
        169250 OK   *** TDO is set to 'Z' ***
        169650 OK   *** loading opcode: 1001***
        172250 OK   **** Checking bypass register output
        174650 OK   Data check OK *** Overshifted bypass register output
equals the inserted value 11110011111
        174650 OK   *** TDO is set to 'Z' ***
        175050 OK   *** loading opcode: 1100***
        177650 OK   **** Checking bypass register output
        180050 OK   Data check OK *** Overshifted bypass register output
equals the inserted value 11110011111
        180050 OK   *** TDO is set to 'Z' ***
Note: Total number of errors in simulation =        0.
```

Figure 6. Simulation results of the MCM with boundary scan circuit inserted into the design.

5. Conclusion

The proposed method is easy to implement and it greatly reduces the additional hardware during boundary scan insertion and the manual intervention during boundary scan pattern generation. Further more, with no modification in either the TAP controller or in the boundary scan cells, the scheme meets the IEEE 1149.1 standard. Thus, there will be no area overhead (except a mux inserted into the each core which is optional) as well as extra delay introduced when compared to the single core boundary scan. Further scope of work can be extended to even automate the generation of test cases required to test the interconnects of cores in a MCM. The proposed method can become a standard for implementing boundary scan and generating patterns in a MCM design.

6. References

[1] IEEE Computer Society. "IEEE Standard Test Access Port and Boundary-Scan architecture – IEEE Std. 1149.1-2001", IEEE, New York, 2001.

[2] Chih-Ang Chen and Sandeep K.Gupta, "BIST/DFT for Performance of Bare Dies and MCMs," *International Conference Proceedings,* combined volumes, pp.803-812, May 1994.

[3] Frans de Jong and Alex Biewenga, "SiP-TAP : JTAG implementation for SiP designs," *IEEE International Test Conference,* pp.1-10, Oct. 2006

[4] Jeffrey M. Aubert, "Boundary Scan Modification to Enhance Multichip Module Testing," *Proc. IEEE Aerospace and Electronics Conference,* vol. 3, pp.979-984, May 1992.

[5] Wang-Dauh Tseng and Kuochen Wang, "Testable design and testing of MCMs based on multifrequency scan," *Proc. Fifth Asian Test Symposium,* pp. 75-80, Nov. 1996

[6] Lubaszewski M, Marzouki M and Touati M.H, "A Pragmatic Test and Diagnosis Methodology for Partially Testable MCMs," *IEEE Multi-Chip Module Conference,* pp.108-113, Mar. 1994.

[7] Werkman H, Hofflinger B and Laquai B, "Boundary Scan Adaption for Active Substrate MCM-TEST," *Proc. Tenth Annual IEEE international ASIC Conference and Exhibit,* pp. 62-66, Sep. 1997

[8] Zorian Y, " A Universal Testability Strategy for MultiChipModules Based on BIST and Boundary Scan," *International Conference on Computer Design Cambridge,* pp. 59-66, Oct. 1992.

[9] Melton M and Brglez F, "Automatic Pattern Generation for Diagnosis of wiring Interconnect Faults," *International Test Conference, Baltimore,* pp. 389-398, Oct. 1992.

[10] Yervant Zorian, "A Structured Testability Approach for Multi-Chip Module Based on BIST and Boundary-Scan," *IEEE Trans. Components, Packaging, and Manufacturing Technology,* vol. 17, pp. 283-290, Aug. 1994.

[11] Kenneth E. Posse, "A Design-for-Testability Architecture for Multichip Modules," *Int. Test Conf., pp. 113-121, 1991.*