

효과적인 AMBA 기반 SoC 테스트를 위한 AHB/PCI 버스 브리지 재사용 기술

Design Reuse of AHB/PCI Bus Bridge
for Efficient Test Access to AMBA-based SoC

한주희, 송재훈, 조상욱, 박성주
한양대학교

{hanjh,jhsong,swcho,parksj}@mslab.hanyang.ac.kr

Abstract

This paper introduces an efficient test access mechanism for Advanced Microcontroller Bus Architecture (AMBA) based SoC to reduce the test application time while minimally adding a new test interface logic. Testable design technique is applied to an SoC with the Advanced High-performance Bus (AHB) and PCI bus bridge by maximally reusing the bridge functions. Testing time can be significantly reduced by increasing the test channels and by shortening the test control protocols. Experimental results show that area overhead and testing times in both functional and structural test modes are considerably reduced

I. Introduction

As deep submicron techniques are increasingly developed, it is possible to design and manufacture a System-on-a-Chip (SoC) comprised of various Intellectual Property (IP) cores meeting short time-to-market requirements. Although the design time can be reduced by utilizing reusable IPs, the testing time is significantly increased because of the high complexity of the SoC. Testing cost is mainly affected by the memory size and application time of the Automatic Test Equipment (ATE), the structure of the core test wrapper, Test Access Mechanism (TAM), and test methodology [1]. It becomes crucial to improve the test quality while maintaining the testing cost as low as possible to survive in the emerging silicon market.

Advanced Microcontroller Bus Architecture (AMBA) is an on-chip bus architecture developed by ARM Ltd. to strengthen the reusability of IP cores, and the AHB is a high performance system bus [2]. To test embedded cores in an AMBA-based SoC, Test Interface Controller (TIC) of ARM Ltd, External Bus Interface (EBI), and test wrappers are extensively adopted [2, 3, 4, 5]. The TIC is an interface controller for the testing of an AMBA-based system performing basic AMBA Read/Write transactions as an AMBA bus master. The EBI is used as an external test bus (TBUS) to transfer test data. The test wrapper allows to access the inputs and outputs of the core that are not directly connected to the on-chip bus.

However, the TIC [2] uses a single path between the TBUS and the AHB for transferring the address and the test data into embedded cores, and also transferring the test responses through the TBUS. Therefore, write and read data transactions must be performed exclusively with additional turnaround cycles to avoid the bus conflict on the TBUS. This leads to an increased test time for not only functional but also structural scan tests. In [3] test wrappers are designed for each core, but scan-in and scan-out must be performed exclusively requiring an extensive amount of test cycles. In [4] although scan chains are allowed to be concurrently scanned-in and out, the modification to the AHB-APB bridge logic impairs the compatibility with AMBA systems.

The Peripheral Component Interconnect (PCI) bus architecture has been widely implemented to connect chips and adaptors on a board [6]. For a PCI board embedding SoCs with AMBA internal bus,

the AHB–PCI bus bridge, often called as an on/off–chip bus bridge, is a necessary component. Such system chips with the AHB–PCI bus bridge have been widely adopted for various applications such as sound, graphic and network cards, etc.

In this paper, we propose an efficient test access mechanism utilizing the AHB–PCI bridge, EBI, and test wrappers. Test application time is considerably reduced by providing dedicated test paths and excluding the bus turnaround delays. Our technique reuses the on/off–chip bus bridge to transfer the external test data from an ATE into an SoC, thereby, the silicon overhead for the test interface logic is minimized.

This paper is organized as follows. Conventional test interface controllers for AMBA based SoCs are introduced in section II and new test control technique is described in section III. Design experiments show the superiority of our technique both in the area and test application time in section IV followed by the conclusions in section V.

II. Test Interfaces for AMBA–based SoC

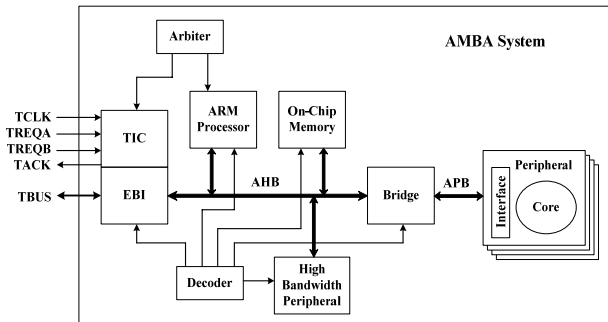


Figure 1. AMBA system with TIC

A conventional AMBA system is comprised of Advanced High–performance Bus (AHB) and Advanced Peripheral Bus (APB). as shown in Figure 1.

The TIC IP core of ARM Ltd. is a test interface controller for the AMBA system performing basic AMBA Read/Write transactions as an AMBA bus master. To access external memory modules in AMBA based SoCs, unidirectional 32 bit address and bidirectional 32 bit data pins of EBI are generally used [7, 8, 9]. By utilizing the EBI and AMBA as test buses, an additional Test Access Mechanism is not required. In test mode, for

isolation, controllability, and observability of the target core, the test wrapper, so called test harness, is defined [2]. The test wrapper can be configured according to the input/output and test strategy of each core, and even cores not complying with AMBA can be accessed through this wrapper [2].

For functional test, all of the TIC–based techniques require additional delays for the bus direction turnaround. For structural test, in [3], the scan test harness with a number of temporary registers was proposed to support cores with more scan chains and I/Os than the bus width. However, scan data cannot be shifted–in and out simultaneously with the current TIC leading to a lengthy test application time. In [4], concurrent scan shifting became possible by taking part of the EBI address bus as scan–out paths, but APB and EBI must be modified with additional bus signals to the AMBA bus, and the number of inputs and outputs of cores linked to APB are required to be less than or equal to 32 and 26, respectively. Although more than 32 scan chains are supported [3], since a number of write/read transactions are needed to scan–in and out with additional buffers [10], various techniques of Illinois scan broadcasting, daisy chains, TestRail, and test bus multiplexing have been suggested [11, 12, 13].

This paper introduces a new design technique for the AMBA–based SoC to reduce test application time significantly by reusing the design resource of the on/off–chip bus bridge.

III. New Test Access Mechanism for AMBA–based SoC

The main contribution of our technique is to *reuse the on/off–chip bus bridge as a test interface during the test mode. The AHB master component on the bridge is reused as an interface between the ATE and the chip under test, and then the ATE acts as a virtual bus master.* By providing a dedicated test path and eliminating the bus direction turnaround delays, test time can be considerably reduced for both functional and structural test. The compatibility with the AMBA protocol is sustained without requiring any internal modification of reusable cores.

The key components of the proposed architecture are the AHB-PCI bus bridge with test controllability, EBI and test wrappers (TW). The bridge is to be called as Test Ready bridge (TR-bridge).

1. Proposed test access architecture

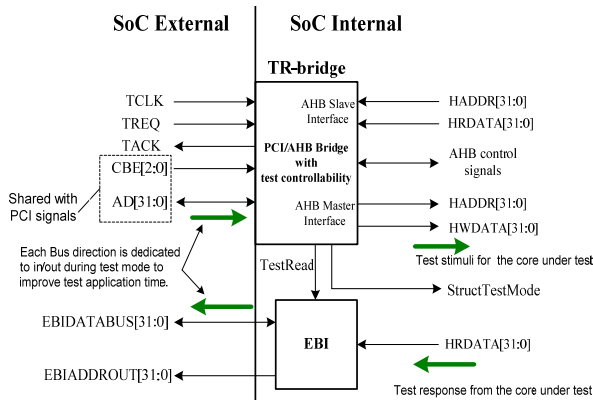


Figure 2. Proposed test access architecture

In the proposed test access architecture shown in Figure 2, the TR-bridge allows externally applied test vectors to be converted for internal bus transfers. The bridge provides both functional and structural test modes. Functional test reuses the test data used for the design verification. Test wrappers (harnesses) are used for the vector application and observation. The bypass multiplexer of the AHB-APB bridge is introduced to speed up the structural test.

The TR-bridge uses a minimal handshake mechanism by using TREQ, TACK and CBE pins to control the application of test vectors, and the CBE is the reused PCI interface [6] for test mode control. The AD bus of PCI interface and the EBI are reused for test data application to provide a high speed parallel vector interface. During the functional or structural test mode, the AD bus and EBIDATABUS are dedicated to apply the test vector and to take out the test response, respectively. The EBI is generally used to communicate with tightly coupled off-chip memories in system operation. The TestRead signal from TR-bridge to EBI controls the EBIDATABUS direction to take out the test response.

The StructTestMode signal from TR-bridge controls each test wrapper of embedded cores. Only during structural test mode, the signal is

asserted to enable test wrappers for structural test such as scan testing.

2. On/off-chip bus bridge with test controllability

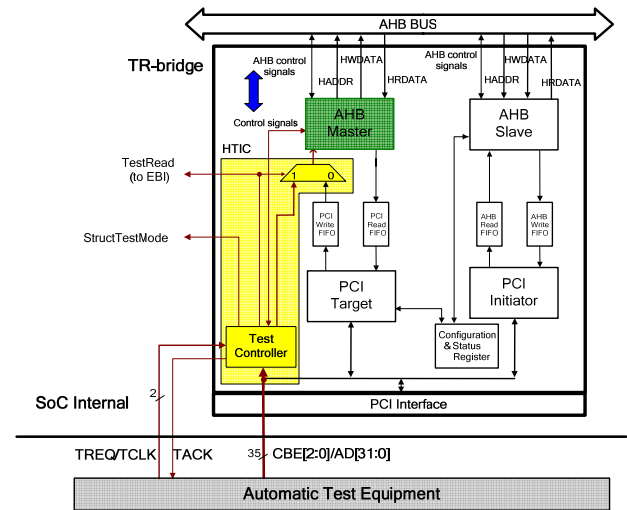


Figure 3. AHB-PCI bus bridge with test controllability during test mode

This section describes the proposed AHB-PCI bus bridge with test controllability, and Figure 4 shows the architecture of the bridge.

The bridge mainly consists of AHB Master, PCI Target, AHB Slave, PCI Initiator and Test Controller blocks.

During normal system operation, the AHB Master and PCI Target blocks act as an AHB bus master and a PCI bus slave, respectively, and these components operate when the SoC becomes a slave with regard to the PCI bus. The AHB Slave and PCI Initiator blocks act as a AHB bus slave and a PCI bus master component, respectively, and these components operate when the SoC is a bus master with regard to the PCI bus during the normal system operation. In Figure 3, the shadowed area named as Hybrid Test Interface Controller (HTIC) is the test control logic which is disabled during normal system operation by de-asserting test request signal, TREQ.

The TR-bridge acts as a test interface controller in which only the HTIC and AHB Master blocks are active during test mode. The HTIC, which consists of a multiplexer and a simple Test Controller block, is a key test control block interfacing an ATE to the AHB Master block. The

multiplexer controls the data path to AHB Master block, and then in test mode, AD bus of PCI interface is directly connected to the AHB Master block not via PCI Target and PCI Write FIFO blocks. Therefore *during the test mode, there is no need to be compatible with the complex PCI protocol, and this simplifies the test sequences. Moreover, this scheme makes it possible not to be affected by PCI speed limit which is relatively lower than the AHB.*

Table 1. Test control signals during normal operation

Input				Output	Description
TREQ	CBE[2]	CBE[1]	CBE[0]	TACK	
0	-	-	-	0	Normal operation
1	0	-	-	0	Request a func. test mode
1	0	-	-	1	Functional test mode entered.
1	1	-	-	0	Request a struct. test mode
1	1	-	-	1	Structural test mode entered

Table 2. Test control signals during either functional or structural test mode

Input				Output	Description
TREQ	CBE[2]	CBE[1]	CBE[0]	TACK	
-	-	-	-	0	Current access is incomplete
1	-	1	1	1	Address vector
1	-	1	0	1	Write vector
1	-	0	1	1	Read vector
1	-	0	0	1	Control vector
0	-	-	-	1	Exist test mode

The AHB bus master block, which is a system functional block, is reused during the test mode to allow externally applied test vectors on AD bus to be transferred into internal AHB bus.

The external test interface consists of a test clock (TCLK), two control signals dedicated to test mode (TREQ, TACK), and control signals shared by PCI interface (CBE[2:0]).

Table 1 and 2 describe the operation of the external test interface signals of HTIC. The signals have different functions depending on the current mode. A dedicated device pins TREQ and TACK indicate the test bus request and test bus acknowledge, respectively.

There are four different types of test vectors associated with the test interface, which are address, write, read, and control vectors. During the test mode, CBE[1:0] signals describe the type of test vector to be applied in the following cycle,

and CBE[2] indicates the test mode of the function and structure. An address vector is used to select a core to be tested. A write vector is for either functional or structural test stimuli and a read vector is for test responses. A control vector updates control values of AHB control signals such as HSIZE, HPROT, HTRANS and HLOCK.

To efficiently transfer test vectors to and from an SoC externally, AD[31:0] of PCI interface and EBIDATABUS[31:0] are reused as the external 32-bit test bus. The AD[31:0] bus, which transfers both address and data in normal mode, is adopted as a dedicated test input in test mode. The EBIDATABUS[31:0] bus, which interfaces external devices in normal system, is connected to an ATE as the test response output port. Such test buses dedicated to write and read explicitly can reduce the test application time significantly, by not asking a turnaround period to change a bus direction, and by allowing simultaneous scan in and out operations.

3. Operation of the test ready bridge

The state diagram in Figure 4 illustrates the operation of the TR-bridge. The TACK signal is used to control all transactions around the state machine, except for the transition from IDLE to START. With the assertion of the TREQ, the HTIC is transited into the test mode. The START state is used to ensure that the first vector applied is an address vector to prevent read and write vectors from entering before the address is initialized. The START state is only exited when CBE[1:0] indicate an address vector and the following state is ADDRVEC. Then, the state machine moves around ADDRVEC, WRITEVEC, READVEC, and CONTVEC states according to values on CBE[1:0].

Figure 5 shows the TIC state diagram [2]. In the TIC-based approach, a read vector or burst of read vectors is always followed by additional two vectors of LASTREAD and TURNAROUND for the turnaround time of test bus direction change as shown in Figure 5. This is because TIC-based scheme uses the same external test bus for any type of transfers, and the read transfers after the other types of transfer require the test bus to be driven in the opposite direction. Therefore, additional cycles are necessary to prevent the bus

clash when the drivers of the test bus are changed [2]. This can significantly increase the functional test time, because in real system the read transactions occur more frequently than others. But, *the proposed technique does not require such turnaround vectors due to separate read and write test buses.* As shown in Figure 4, four states of *ADDRVEC*, *WRITEVEC*, *READVEC* and *CONTVEC* constitute a complete directed graph, thus any transition among these states needs only one clock cycle. For a control vector, TIC-based approaches always require at least one address vector prior to the control vector [2]. However, in our technique the control vector can be applied independently by introducing the *CONTVEC* state as shown in Figure 4 leading to further reduce the test application time.

To compare test clocks required by our method and TIC-based scheme, we will estimate the number of transitions in the state diagrams for a series of test sequence. If k , m , n , and p number of state transitions are needed in Figure 4, how many transitions are needed in Figure 5? Three key different cases are analyzed as follows:

Case 1)

The transition from *READVEC* to *WRITEVEC* or from *READVEC* to *ADDRVEC* requires three clocks respectively in Figure 5 instead of one clock in our method of Figure 4, thus the TIC-based scheme takes $3(k + m)$ clock cycles while ours just takes $(k+m)$ clock cycles.

Case 2)

The transition from *READVEC* to *CONTVEC* in Figure 4 requires at least four clocks in Figure 5 because the last vector of more than one successive address vectors is considered to be a control vector [2]. Therefore, the TIC-based scheme takes $4(n)$ while ours takes (n) clock cycles.

Case 3)

The transition from *WRITEVEC* to *CONTVEC* in Figure 4 requires at least two clocks in Figure 5 due to the same reason in case 2. Therefore, the TIC-based scheme takes $2(p)$ while ours takes (p) clock cycles

From the case 1, 2, and 3, it can be observed that the TIC-based scheme takes $(3(k + m) + 4n)$

+ $2p)$ while our approach takes $(k+m+n+p)$ clock cycles.

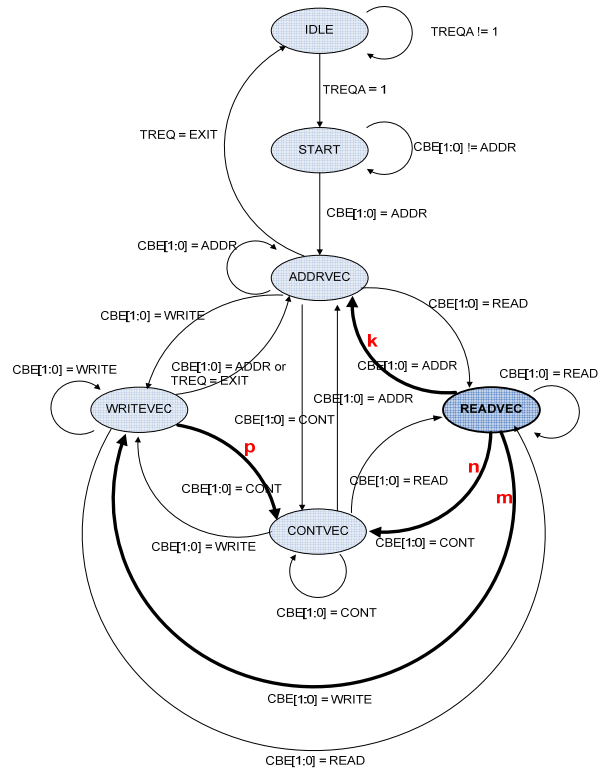


Figure 4. State diagram of the proposed Hybrid Test Interface Controller

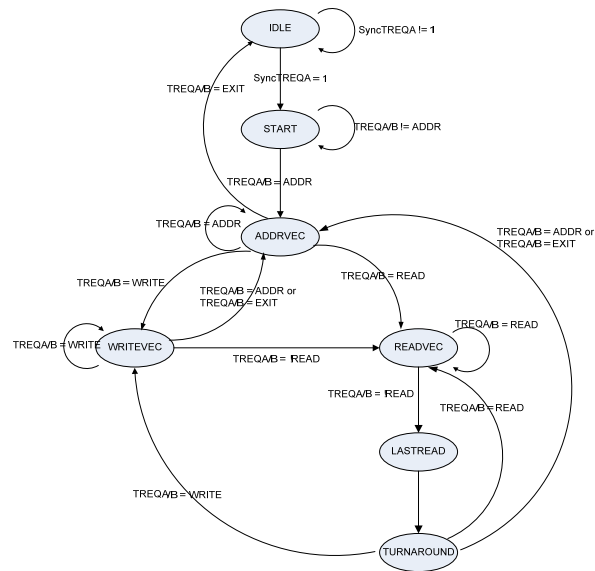


Figure 5. State diagram of the ARM Test Interface Controller

4. Efficient Structural test

For the structural test such as a scan test, we adopt the technique in [14] for the test wrappers

(harnesses) which are used for the vector application and observation. And we also adopt the AHB-APB bridge bypass multiplexer in [14] which is used to speed up the structural test, but the ad-hoc logic is excluded in our scheme.

The scan-in and out paths are used exclusively by the AD bus of the TR-bridge and EBI bus, thus the concurrent test application and observation are possible without requesting any read transactions to the AMBA bus during the scan shifting. Since our technique uses only the write transactions to the AMBA bus excluding the additional cycles for the read transactions, the scan test time can be significantly reduced.

IV. Design Experiments

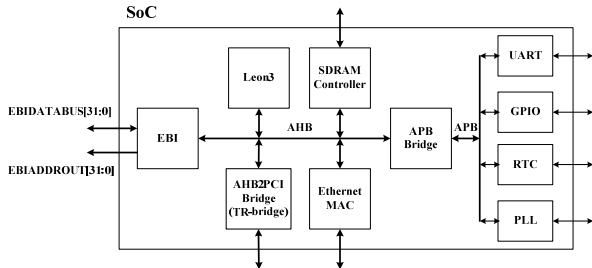


Figure 6. Example of AMBA-based system

The AMBA-based SoC of Figure 6 is adopted to evaluate the area overhead and test application time for each embedded core [15]. Excluding the PLL, 32 scan chains are inserted into each of the embedded cores to maximally utilize the scan-in and out paths to an ATE in an AMBA-based SoC. The test patterns are generated through Synopsys TetraMAX. The RTL codes are synthesized by the Synopsys Design Compiler with TSMC 0.25 μ m library, and the simulation is performed using the ModelSim.

The test harness of [3] includes the registers for both scan inputs and PIs, however test wrappers for our technique include only the registers for PIs. In addition, the proposed includes very small test control logic in the bridge, but [3] requires more silicon overhead due to the whole TIC. In Table 3, the gate counts for both [3] and the proposed are compared as the number of equivalent two-input NAND gates. As shown in the table, our scheme leads to about 13.83% area saving than [3].

Table 3. Comparison of area overhead

Cores		Area overheads (The number of NAND gates)		Area red. (%)
		[3]	Proposed	
AHB	Test interface controller	2983	709	76.23
	Leon3 processor	15782	15171	3.87
	SDRAM controller	3456	3224	6.71
	AHB2PCI bridge	2624	1853	29.38
	Ethernet MAC	4531	4301	5.08
APB	UART	1511	1271	15.88
	GPIO	2136	1944	8.99
	RTC	1168	988	15.41
Total		34191	29461	13.83

Table 4. Comparison of functional test time

	Test Time (The number of clocks)		Red. (%)
	TIC	Proposed	
READVEC → WRITEVEC	27720	9240	66.67
READVEC → ADDRVEC	23643	7881	66.67
READVEC → CONTVEC	860	215	75
WRITEVEC → CONTVEC	278	139	50
Others	45578	45567	0
Total	98079	63042	35.72

Table 5. Comparison of structural test time

Cores		Test Time (The number of clocks)		Test time red. (%)
		[3]	Proposed	
AHB	Leon3 processor	47478	31306	34.06
	SDRAM controller	2652	1846	30.39
	Ethernet MAC	55290	32540	41.15
APB	UART	19866	6026	69.67
	GPIO	468	201	57.26
	RTC	7800	2484	68.17
Total		133554	74403	44.29

Especially as shown in the second row if the area overheads for the TIC and HTIC are considered, about 76.23% reduction is achieved by reusing the bridge design resource.

Table 4 compares the functional test time for the key transactions of the first column. On average 35.72% test cycles are reduced by using our technique when functional verification patterns are applied for this experiment.

In Table 5, it is shown that the test application times for structural test of AHB and APB cores are considerably reduced by on average 37.69% and

69.05% respectively, and globally 44.29%.

In the sequel, it is believed that the proposed technique contributes to reduce test application time significantly with minimal area overhead.

V. Conclusions

In this paper a design reuse technique of the AHB-PCI bus bridge for an efficient test access mechanism is proposed. Only simple logic is added to the on/off-chip bus bridge to utilize its functionality as a test interface controller. By discarding the bus turnaround time and by utilizing the Extended Bus Interface as the test output channel, the functional and structural test times are significantly reduced. Though the proposed technique is designed for the AHB-PCI bridge, our scheme can be extensively applied to other types of on/off-chip bus bridges to reduce test cost with minimal area overhead.

References

- [1] Y. Zorian, E. J. Marinissen and S. Dey, "Testing Embedded-core based System Chips," In Proceedings IEEE International Test Conference, pp. 130-143, Oct. 1998.
- [2] ARM IHI 0011A, "AMBA Specification (Rev 2.0)". May 1999.
- [3] C. Feige et al, "Integration of the Scan-Test Method into an Architecture Specific Core-Test Approach," Journal of Electronic Testing, Volume 14, pp. 125-131, July 1998.
- [4] C. Lin and H. Liang, "Bus-Oriented DFT Design for Embedded Cores," IEEE Asia-Pacific Conference, Volume 1, pp. 561-563, Dec. 2004.
- [5] Advanced RISC Machines, "AHB Example AMBA System Technical Reference Manual," ARM DDI 0170A, Aug. 1999.
- [6] PCI Special Interest Group, "PCI Local Bus Specification", revision 2.2, Dec. 1998.
- [7] Advanced RISC Machines, "ARM PrimeCell External Bus Interface (PL220)," ARM DDI 0249B, Dec. 2002.
- [8] ALTERA, "Excalibur Devices Hardware Reference Manual," Version 3.1, Nov. 2002.
- [9] Atmel Corporation, "AT91 ARM Thumb Microcontrollers," AT91R40807, Jan. 2002.
- [10] P. Harrod, "Testing Reusable IP - A Case Study," In Proceedings of IEEE International Test Conference, pp. 493-498, Sep 1999.
- [11] J. Aerts and E.J. Marinissen, "Scan Chain Design for Test Time Reduction in Core-Based ICs," Proc. Int'l Test Conf., pp. 448-457, 1998.
- [12] E. J. Marinissen et al. "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores," In Proceedings IEEE International Test Conference, Oct. 1998.
- [13] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Full-Scan Embedded Cores," Proc. 29th Int'l Symp. Fault-Tolerant Computing (FTCS 99), Digest of Papers, IEEE CS Press, Los Alamitos, Calif., pp. 260-267, 1999.
- [14] J. Song, P. Min, H. Yi and S. Park, "Design of Test Access Mechanism for AMBA Based System-on-a-Chip", IEEE VLSI Test Symposium, May 2007, to be published.
- [15] J. Gaisler and E. Catovic, "Gaisler Research IP Core's Manual," version 1.0.1, Jun. 2