The Design-for-Testability Features of A General Purpose Microprocessor

Da Wang\textsuperscript{1,2}, Xiaoxin Fan\textsuperscript{1,2}, Xiang Fu\textsuperscript{1,2}, Hui Liu\textsuperscript{1,2}, Ke Wen\textsuperscript{1,2}, Rui Li\textsuperscript{3}, Huawei Li\textsuperscript{1}, Yu Hu\textsuperscript{1}, and Xiaowei Li\textsuperscript{1}

1. Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
2. Graduate University of Chinese Academy of Sciences (GUCAS), Beijing, China
3. STMicroelectronics (Shanghai) Co., Ltd., Shanghai, China

E-mail: \{wangda, fanxiaoxin, fuxiang, wenke, liuhui\}@ict.ac.cn, peter.li@st.com, \{lihuawei, huyu, lxw\}@ict.ac.cn

Abstract

This paper describes the design-for-testability (DFT) features and test challenges in a general purpose microprocessor design. An optimized DFT architecture with its implementation strategies are presented in detail. Major DFT solutions are implemented which can meet high-volume manufacturing (HVM) and high quality test goals.

1. Introduction

Design-for-testability (DFT) techniques are critical to achieve economical integrated circuit (IC) testing with high quality and test coverage. This paper introduces the DFT architecture with its implementation strategies on a general purpose microprocessor designed in China. It is a nine-stage super-pipelining and four-issue general purpose RISC microprocessor based on 64-bit MIPS instruction set [1-2] (Figure 1). Its recent version has the following features. Pipeline stall is reduced by using a 64-entry TLB and larger entry number for reorder buffer and memory queue. Memory performance is enhanced with on-chip 144KB L1 cache, 576KB L2 cache and 333MHz DDR2 memory controller. The microprocessor includes nine logic blocks with 3 main clock domains, and JTAG with TCK clock domain, as shown in Figure 2. The IO block consists of L2 cache and Arbiter. The microprocessor contains embedded L1, L2 cache and custom register files. L1 cache is designed with separated data and instruction cache, each of which is 72KB and four-way set-associative. There are totally 0.73MB embedded memories in the chip. The chip is implemented with ST 90nm, 7 layer metal CMOS technology. It contains approximately 50 million transistors, and can work at 1GHz.

Since the microprocessor is 1GHz in 90nm silicon technology, it is difficult to test with former DFT solutions. The main fault models are changing from single stuck-at fault to delay fault models. Hence, it is confronting with many DFT challenges based on its own character compared to the previous designs [3-6]. First of all, there are more custom units in the critical paths of this microprocessor. To minimize DFT timing impact on the functional design is very important. These custom units are also obstacles for the improvement of test coverage. Secondly, considering the high performance of the design, it is necessary to test the whole chip at-speed and reach acceptable delay fault test coverage. Meanwhile, we should also endeavor to reduce both test power dissipation and test data volume for manufacturing test.

Techniques are used to meet manufacturing requirements of low test data volume and small test power dissipation.

This paper focuses on the DFT and automatic test pattern generation (ATPG) strategies for the microprocessor. The challenges of at-speed testing are also addressed. Techniques are used to meet manufacturing requirements of low test data volume and small test power dissipation.

Figure 1 Architecture of the Microprocessor

---

This paper is supported in part by National Natural Science Foundation of China (NSFC) under grant No. 60633060, 60606008, 90607010, 60576031, and in part by National Basic Research Program of China (973) under grant No. 2005CB321604 and 2005CB321605.
The paper is organized as follows. Section 2 describes the DFT features at register transfer level (RTL) and highlights the features for at-speed testing. A scan compression design is given in Section 3. Section 4 introduces the test pattern generation flow to reduce test data volume further more based on scan compression design. It also focuses on at-speed test pattern generation with considering test power dissipation. Future work on speed grading is described in Section 5. Section 6 gives the limitation of this work. Section 7 concludes this paper.

Figure 2   Blocks and Clock Domains of the Microprocessor

2. DFT Features at RTL

The microprocessor has several test modes for both manufacture test and system debug. The DFT logic of this microprocessor is designed to achieve high test coverage, ease at-speed testing, reduce test power dissipation, and shorten function and silicon debug cycle. In this section, a flexible design of using PLL for at-speed test is proposed. Techniques used to meet manufacturing requirement of test coverage and low power consumption are also discussed. The implementation detail of embedded memory test is given at the end of this section. The DFT sketch of this microprocessor is shown in Figure 3.

2.1  Clock and Test Control (CTC) Unit

CTC unit is the main clock and test control logic of the design, as shown in light blue at left downside in Figure 3. It not only guarantees proper clock for the design under different phases, but also controls the test signals. There are multiple clock domains in the design. The DRAM interface DDR2 block works in the ddr_clk clock domain, which is supplied by PLL_1. The Northbridge interface PCI block runs with the PCI clock pci_clk generated from a self-governed off-chip clock. Other seven blocks are running in the PLL_clk clock domain, which is the fastest clock supplied by PLL_0. In addition, JTAG runs in the TCK clock domain, as shown in Figure 2. PLL_0 and PLL_1 are the same type of PLL to provide on-chip clock. Each of these two PLLs is programmable by 5 configuration bits. The first three bits are used to multiply reference clock by 18 to 32 times, and the last two bits are used to divide the clock by one to sixteen. Thus, the output clock from PLL can be 9/8 to 32 times of the reference clock. To directly using test clock signal from ATE, bypass PLL logic is also designed. The output of PLL_0 is PLL_clk with a typical frequency of 1GHz. The output of PLL_1 is ddr_clk with a typical frequency of 333MHz. The pci_clk works at a typical frequency of 133MHz.

Figure 3   DFT Sketch of the Microprocessor
To ensure these various clocks run correctly during test, clock control logic is designed to give proper clock for test and functional modes, as shown in Figure 4. Three multiplexers are used to select suitable clock as an internal clock. Multiplexer ‘A’ uses ‘test_mode’ signal as the selection signal to choose needed test clock for test mode or \( PLL_{\text{clk}} \) for function mode. Multiplexer ‘B’ selects clock from ATE for low speed test or the output of multiplexer ‘C’ for at-speed test. Multiplexer ‘C’ is set by \( SE \) signal, and it can propagate the test sequences from PCC unit which is discussed in subsection 2.2. The ‘Internal\_clk’ signal can be used as \( core_{\text{clk}}, dir_{\text{clk}} \), and clock for Memory BIST (MBIST). The output of \( PLL_0, PLL_{\text{clk}} \), is divided into one-eighth and observed from ATE directly with considering the strobe ability of probes.

The CTC unit is also designed with the following purposes.

(a) Enhance test controllability and simplify the relationship between test control signals. By utilizing CTC, control signals for scan and AC/DC test are globally set and directly controlled from ATE. Test control signals for MBIST and other tests are individually set for blocks with combinational logic. Thus, it is more convenient for partition test to prevent power burst during whole chip test and facilitative test and debug process at block level. This method also reduces the overhead cost of DFT.

(b) Utilize IEEE Standard 1149.1 to access, test and debug the microprocessor. The JTAG controller, TAP, uses a 4-bit instruction register and more than 12 instructions to conduct test operations. It is controlled by 1149.1 interface pins, which consist of TCK, TDI, TMS, TDO, and TRST. Complying with the IEEE 1149.1 boundary scan specification, it supports standard commands such as BYPASS, HIGHZ, EXTEST and SAMPLE/PRELOAD for normal DC test and system test. IOmap design is also included in the JTAG module.

### 2.2 PLL Clock Control (PCC) Unit

Since the frequency of ATE is limited, using on-chip clock to apply at-speed test for a high performance design is an economic solution. Utilizing PLL clock for at-speed test has been studied in many cases [4, 7]. In this processor, at-speed test plays an important role in both defects detection and speed grading.

Researches in [8, 9] declare that typical transition fault (TF) test cannot attain high test coverage for a complex design with embedded memories, because the detection of some TFs must go through the memories, either for control or for propagation. They also point out that some paths in an IC design provided by static timing analysis (STA) as critical paths cannot really work at the highest speed of the IC, while some paths going through memory and stealing cycles are the real critical paths. Custom storage units and memory arrays in the microprocessor are bottleneck for at-speed test. The following two reasons make it difficult to test the paths that go through memories: (a) it is generally hard to model memories properly for ATPG tools, (b) the detection of faults which exist in the paths going through memories requires longer test sequences and more complex clock [7].

In the DFT design for this microprocessor, launch-off-capture approach is chosen as at-speed ATPG method. To solve the problem (b) mentioned above, a PLL Clock Control (PCC) unit is designed based on our previous work [10].

In Figure 5, the inputs of PCC unit are \( PLL_{\text{clk}} \) which is the clock signal from PLL, \( SE \) which is the scan enable signal from ATE, and \( PI \) which is the primary input of the clock_chain. In this unit, AND-gate network is the key logic to export test sequences. Each AND gate in the AND-gate network has three inputs. One of the inputs is connected with the Q port of a register, which comes from a special scan chain named clock_chain within test clock domain. The other two inputs are connected with the positive input value and the negative output value of the register, which comes from the upside_chain within \( PLL_{\text{clk}} \) clock domain. The clock_chain is used to set control signals for AND gates to generate different clock waves. Data in the clock_chain is generated by ATPG flow, which will be presented in section 4. The upside_chain is used to properly propagate SE signal without glitch. To make SE signal stable, a D flip-flop is used to lock its value with the active edge of test clock, which is inserted after the SE signal.

In the test mode, the negative value of SE is sampled and shifted into the upside_chain which is a 14-bit registers chain within \( PLL_{\text{clk}} \) clock domain. When SE is high (active), which means the circuit under test (CUT) is in the scan shift load/unload mode, values in the upside_chain are all ‘0’s. Once the load/unload process is finished, ATE will set SE to low (inactive) to let the CUT change from shift mode to launch-capture mode.

Before launch-capture, there should be enough setup time to ensure all flip-flops are in stable mode. To meet this requirement, five PLL clock cycles are inserted as hold time by using shift_reg[0-4] registers in the upside_chain. These registers work as a glacis of the switch from a slow clock to a fast clock to provide enough setup time during SE signal transfer and avoid high current appearance during clock transfer.
When SE has switched from ‘1’ to ‘0’ and transmitted to shift_reg[4], the positive input value of shift_reg[5]’s and the negative value of its output work together with the value coming from register clock_chain[0] and set the first AND gate to export the value of clock_chain[0]. At the next active edge of PLL_clk, shift_reg[5] will get the new SE value and disable the first AND gate, it insures the output of the first AND gate is an integrated PLL_clk cycle. The rest of the logic works in a similar way. Shift_reg[9-13] work as same as shift_reg[0-4] to provide enough hold time.

The function of PCC unit is to supply proper high-frequency pulses for at-speed test, as indicated in Figure 6. The clock signal ‘PCC_CLK’ is valid only when SE is low. It can have four pulses in the maximal case, and two pulses in the minimal case. If there are two ‘1’s in clock_chain[0] and clock_chain[1], two ‘PCC_CLK’ clock cycles are provided, as shown with black real lines in Figure 6. If there are three or four ‘1’s orderly in the registers of clock_chain, one or two more ‘PCC_CLK’ clock cycles are provided, as shown with green broken lines in Figure 6. It is simple for test patterns to choose ‘PCC_CLK’ cycles to go through all sorts of paths during at-speed test application.

2.3 Clock Gating

To reduce power consumption, devices used only for test (flip-flops, registers and latches) should be turned off in the functional mode. Due to the divisional setting of test control signals, clock gating logic is introduced for each block. Hence, unused elements in untested block can be turned off during testing to save power. In Figure 7, ‘test_mode’ and ‘BIST_mode’ signals in each block are given by its own test control signals. These two control signals are used as inputs of an OR gate. The output of the OR gate is the enable signal of the clock gating cell. ‘TE’ in the clock gating cell is connected to ‘SE’ signal, which can be directly controlled by ATE. In this microprocessor, by separating the functional and test logic of clock gating cell, it is very convenient to control clock of each block in either test mode or functional mode. Such clock gating design can avoid long wire routing, decrease delay brought by long wire, and facilitate controlling clocks from the top level. Clock for embedded memories is also controlled by setting ‘BIST_mode’ and ‘SE’ signals during testing. This logic can stop the memory clock when needed to minimize test power.

2.4 Memory BIST

To minimize the timing impact to the functional design, MBIST in this microprocessor only covers L1, L2 cache, and part of custom regfiles which are not in critical paths. For the custom regfiles in critical paths without MBIST,
the percentage of undetected faults only occupies a small proportion of the whole faults set, which is less than 0.5%. In order to increase test coverage, scanable flip-flops are inserted into each port of these unBISTed regfiles to test shadow logic. MBIST in this microprocessor is designed to provide at-speed testing for memory arrays. Back-to-back reads/writes operations are used to detect dynamic memory defects. The MBIST is also designed to simplify memory debug process and save MBIST area, the details will be discussed later in this section.

To meet silicon debug requirement, bitmap are usually fed into MBIST. A bitmap contains at least the information of one failure word, such as word address, data, status, etc.. A bitmap with one failure word for a $512 \times 32$ memory array will easily cost more than 50 registers to store the needed information. There are totally more than 120 memory arrays in L1 and L2 cache. If bitmap is inserted for each array, the area overhead cannot be tolerated. All the bitmaps will add more than ten thousand registers, which will increase test complexity and observation difficulties. Scan collar is designed to release from this inferior position. A scan collar is a specific scan chain with particular scan cells (P-cells) connecting to ports of the memory array in the MBIST logic to replace bitmaps for memory debug, as shown in Figure 8. It only indicates one BIST controller for one memory array. The using of scan collar has following advantages. (a) Various test data generated by different test algorithms can access caches to test and debug memories through the scan collar. (b) Instructions for functional test can be easily written into the caches beforehand through the scan collar. This makes it easy to do functional test with low cost ATE. (c) Shadow logic becomes much easier to test, since scan collar can also be used as a normal scan chain during scan test.

In the scan collar mode, data are shifted into the scan collar at first, and then written into the D port of the memory within one cycle. Then, data can be captured from Q port in next cycle. The scan collar can shift and capture data at a low speed, or shift at a low speed and capture with \( PCC_{\text{clk}} \), which depends on the motivation of memory written. During the shift process, ‘SE’ signal is high and acts as a memory array selection enable (CSN) signal. During the capture process, CSN is determined by the value of the register connected to CSN. In the scan collar, ‘SE’ is also used to control the write-enable port of the cache to protect values in the cache, as shown in Figure 10. Data in the flip-flops can be written into the cache only when ‘SE’ is low.

Considering the memory locations and prevention of wires congestion, a MBIST controller is arranged for every four memory arrays. Figure 11 presents an integrated scan chain contains several MBIST scan collars. One MBIST scan collar can be a part of a scan chain. Or several scan collars can be connected together as a long scan chain. These structures make it easy to shift in/out data from caches for test and debug.
In this design, all L1 and L2 memory arrays are provided with test bypass function. In MBIST, to avoid combinational feedback loop, which happens easily when the memory is used as a buffer, some of the memory bypass path is reconstructed including a synchronous component, as shown in Figure 12.

Figure 12 Sequential Test Bypass

To save test resource, MBIST results of each cut, such as bist_fail, bist_bad, bist_end, are exported to the primary output (PO) pads through an OR gate or an AND gate. For example, each group of ‘bist_fail’ signals are compacted by an OR gate, and all ‘bist_end’ signals are compacted by an AND gate. The compaction leads to some information loss. As a result, although it is known at least one of the grouped memory arrays fails, it is hard to tell exactly which memory array fails. To solve this problem, the MBIST results of each memory array are also fed into registers. By connecting these registers to a scan chain, the output of these registers will be shifted out, when the failure memory need analysis. By this way, it can exactly tell that in which memory array has defects.

3. Hierarchical Scan Compression with Hybrid Structure

There are about 106,500 flip-flops in the microprocessor. 98% of these flip-flops are scannable D-Flip-Flops (DFF), the other 2% non-scannable DFFs are distributed in JTAG and some control logic. To decrease test data volume, scan compression is used in this DFT strategy. A good scan compression design needs to get a better trade off between compression rate, test coverage loss tolerance, physical implementation and test flexibility. In this design, Hierarchical scan compress (HSC) structure is applied to the nine logic blocks in order to shorten physical design time. A single scan chain, which connects the clock chains in the two PCC units, is proposed together with the HSC logics to constitute a hierarchical scan compress with hybrid structure (HSCHS) scheme, as shown in Figure 13.

Figure 13 

There are three major reasons why the HSCHS scheme adopts a bottom up approach of scan insertion. Firstly, each physical block can be laid out individually such that the schedules of the physical design will not be impacted. Secondly, inserting scan compress (SC) logic for each block can ease the floor planning and routing, and avoid the wire cluster at the top level. Finally, it is flexible to apply scan test for each block with its own SC logic, since blocks with different clock domains could be tested individually without confusing.

As shown in Figure 13, MUXs and XORs networks are used as data compressing and decompressing logic in each SC design. Each scan in/out port of an IOsubblock is connected to the PI and PO through the ports of IO block. Due to some physical design reasons, the paths between the scan out ports of some small blocks to the PO must go through the IO block. Such a bottom up scan insertion makes it more complex to trace scan chains. More convenient SC design is needed in future plans.

Figure 13 also indicates that the short scan chains in the compress placement can be connected together to form a long chain. The long chain, indicated as broken lines in Figure 13, is called internal scan chain. Therefore, the scan chains can work in SC mode or in internal scan mode. All the scan chains in SC mode have the same length of 80 flip-flops. The lengths of internal scan chains are different. The longest one is composed of about 1100 flip-flops, while the shortest one has slightly more than 800 flip-flops. A dedicated PI ‘test_compress’ is used to switch between SC mode and internal scan mode. In the internal scan mode, some test coverage lost by SC mode could be gained. The compress ratio of HSCHS is at least 10 times more than normal scan insertion.

With the help of hierarchical SC design and individual test control signal settings, it is easy to run a pseudo partition scan test policy. For example, ATPG engine can be run to generate test patterns only for IOsubblock_1 while leaving other blocks untouched. For the blocks that are not test target, constant ‘0’s or ‘1’s or other values can be injected, which puts these blocks in a least active state. Thus, the communication logic between IOsubblock_1 and other blocks can also be tested with this method. This divide and conquer method could prevent a power burst of the whole chip. Meanwhile, test coverage lost by partition test can be gained by adding test patterns, which could be accepted by ATE memory. The shortcoming of the pseudo partition scan is the inrescent test time, which is more than twice that of normal scan test.

4. Test Patterns Generation

The test goal of high-volume manufacturing (HVM) is to detect as many defects as possible with fewer test patterns. Defect detection of a complex design needs quite a lot test resource. With the increasing number of test patterns, the capacity of tester memory become far from enough. Embedded test source insertion with traditional pattern generation flow cannot satisfy the whole HVM test goals. To achieve higher test coverage with lower test data volume, a smart pattern generation flow is proposed and given in this section. This flow also considers power consumption during test, especially during delay test. A smart method is used to produce delay test patterns using complex test sequences from PCC unit.
4.1 Test Pattern Generation Flow

The test patterns for transition fault (TF), path-delay fault (PDF), and bridge fault (BF) testing can also detect a large number of stuck-at faults. To minimize the scan test patterns for different fault models, the following ATPG flow is used to generate test patterns.

1. Generate TF and PDF test patterns with the method described in subsection 4.2. The final TF test coverage is 85%. PDF test patterns are generated for selected critical paths.

2. Generate BF test patterns based on node pair files. The final BF test coverage is 85%.

3. Use the TF, PDF and BF test patterns obtained in the previous steps to run fault simulation based on single stuck-at fault (SSAF) model. The remaining undetected SSAF fault will be considered in step 4.

4. Generate the incremental patterns for the undetected SSAFs. Since the SC mode may lead to some coverage loss, ATPG is run in both SC mode and internal scan mode. Sequential ATPG is also run to enhance test coverage. The circuit could achieve more than 99% test coverage for SSAFs.

5. $I_{DQ}$ test is also conducted. The $I_{DQ}$ test patterns are chosen from SSAF test patterns. The $I_{DQ}$ test coverage by these selected patterns can reach 90% with 10 patterns.

Since most test patterns are generated in compress mode, the test data volume is only 14% that of test generation in internal mode with the same test coverage. In addition, 6% of test data volume is reduced by executing the given ATPG flow. To sum up, it takes less than 65ms to apply delay test patterns at 1GHz without using pseudo partition test strategy, while it takes totally about 30ms to apply other scan patterns at 50MHz. All the delay test patterns are shift in/out at 33MHz, and the other scan patterns are working at 50MHz.

4.2 At-Speed Delay Test Pattern Generation

As mentioned in section 2, there are two problems to generate delay test patterns going through memories. The first issue is how to model memories properly. To solve this issue, memory models are provided as simple as possible for the ATPG tools to recognize and generate suitable test patterns. To solve the second one, PCC unit is designed to support complex and flexible launch and capture sequences. After delay test generation, exhaustive simulation is executed to verify that the patterns can be applied at 1GHz. The patterns that result in timing violation are discarded manually. Pattern generation processes for core clk, ddr_clk and pci_clk domains are activated in parallel.

Power dissipation during test is an important issue that must be addressed. The previous mentioned pseudo partition pattern generation strategy is used to prevent power burst. The experiments fall into three categories.

- If we synthesize the MBIST logic at 1GHz and run TF ATPG without using pseudo partition test strategy, the TF test patterns can get more than 90% test coverage easily. At the same time, the test power during TF test is about twice that of function by simulation. In addition, if MBIST is run at 1GHz, the embedded memories might be destroyed due to the high switch ratio.

- If we synthesize the MBIST logic at 1GHz and use pseudo partition TF ATPG strategy, the test power during TF test is decreased sharply. But at-speed test time is increased to about 100ms to gain 90% TF test coverage. Further more, the memory test power is not reduced during high speed MBIST test.

- If we synthesis the MBIST logic at 600MHz and run TF ATPG without using pseudo partition test strategy, there are X bits output from the MBIST logic during TF ATPG in the core clk domain. To avoid these X bits aliasing, clock gating is used to disable the MBIST logic during TF ATPG at core_clk, as described in
subsection 2.3. The TF test coverage only achieves 85% which is 5% lower than the previous result. Since the MBIST logic is run at 600MHz, therefore, the test power for both memory test and delay test can be endured by the die without increasing TF test time.

By trading off test power with test time, the third method is adopted in the DFT flow. The test power consumption during TF test for the global die is no more than 5W in our simulation result.

4.3 IDDQ Test

The microprocessor is designed based on ST 90nm CMOS technology. During the entire project, the design-for-manufacturability (DFM) rules are strictly followed to avoid any violation which may cause a high static leakage current. The simulation indicated the leakage current can be more than 0.5 Ampere. From our test result of a former design of this microprocessor with the same 90nm technology, the $I_{DDQ}$ current is only about 30% of the simulation result. This test report shows some interesting information. The $I_{DDQ}$ current changes distinctly with the changing of wafer lots. Even in the same wafer, the value of the $I_{DDQ}$ current appears as normal distribution. The current peak of each wafer is dissimilar, and all the peaks fluctuate in a 20mA zone. Hence, it is much difficult to set current threshold during production test. In the recent version of the microprocessor, although it is hard to set threshold value for electrical wafer sort (EWS) and final test, $I_{DDQ}$ test is still in plan, and dynamic leakage current needs further research.

5. Future Work on Speed Grading

[7, 8, 11, 12] point out that speed grading with functional test alone is not an advisable way. It is very time consuming to produce and debug functional patterns for a complex high performance design. Using structural test patterns in speed grading is an appropriate way, and PDF test is a preferable choice for speed grading. Since all kinds of paths should be included to indicate the speed of the design [9], such as the paths going through the memory array, complex test sequences are required.

In this project, not only the paths selected by STA, but also the paths going through memories are considered as paths that need to be tested. These paths work at different speeds and can indicate the real performance of the chip. The planned speed grading scheme includes 1) a functional program that is executed from the embedded caches, and 2) delay test patterns that cover all kinds of paths. The functional program is generated for paths that disperse on the whole chip as much as possible. For the paths not covered by the finite functional test patterns, delay test patterns will reinforce.

Although this paper provides solutions to most challenges that have been discussed so far, there are still design and test limitations in this work.

- Since we synthesis the MBIST logic at 600MHz, we can’t use structural patterns to test memory at speed. Only when functional test is executed, memory is working at speed.
- With the increasing complexity of future designs, more flexible scan compression structure is needed.
- There still lacks of an effective strategy to find a better trade-off between test time and test power for at-speed test.
- A lot of work should be done to achieve speed grading in an economic way with structural and/or functional test patterns.
- Since it is difficult to set $I_{DDQ}$ current threshold for a 90nm-technology design, more attention should be paid to study the static and dynamic current during $I_{DDQ}$ test.

7. Conclusions

These DFT features provide an efficient test strategy with low test power dissipation, low test data volume, and comprehensive at-speed test. The main features and gained experiences are concluded as follows:

- The design of the PCC units to afford complex and flexible cycles is an effective method for at-speed test. At the same time, the delay test pattern generation flow was a tough task and took two engineers more than one month to finish.
- Simplifying the relationship between control signals can improve the flexibility of testing, and increase the test coverage as well. With the method adopted in the project, it also makes it easy for pseudo partition test, and can turn off untested logic to save test power.
- A 10 times compress ratio is obtained by using the hierarchical scan compress with hybrid structure scheme. The bottom up scan insertion flow is used to meet physical design requirement.
- The MBIST with scan collars implemented in the microprocessor eases the process of functional test, failure analysis and debugging.
- The given test compression logic and ATPG flow reduce 87% test data volume.

The DFT logic and test strategies used in this microprocessor design can satisfy the HVM test requirements and gain very good test quality. The test policy is also useful for reference in future nano-technology designs.
8. Acknowledgements
The authors would like to acknowledge the help provided by the design and verification teams of the microprocessor. Also thanks to ST Computer Periphery Group staffs for the valuable feedback. Thanks to Dr. Yinhe Han in our laboratory, and technical support engineers of EDA vendors for their valuable suggestions.

9. References