Qi ZC, Liu H, Li XK *et al.* Design for testability features of Godson-3 multicore microprocessor. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(2): 302–313 Mar. 2011. DOI 10.1007/s11390-011-9437-2

Design for Testability Features of Godson-3 Multicore Microprocessor

Zi-Chu Qi (齐子初), Hui Liu (刘　慧), Xiang-Ku Li (李向库), and Wei-Wu Hu (胡伟武)

Key Laboratory of Computer System and Architecture, Chinese Academy of Sciences, Beijing 100190, China

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Loongson Technologies Corporation Limited, Beijing 100190, China

E-mail: {qizichu, liuhui, lixiangku, hww}@ict.ac.cn

Received July 30, 2009; revised November 29, 2010.

Abstract This paper describes the design for testability (DFT) challenges and techniques of Godson-3 microprocessor, which is a scalable multicore processor based on the scalable mesh of crossbar (SMOC) on-chip network and targets high-end applications. Advanced techniques are adopted to make the DFT design scalable and achieve low-power and low-cost test with limited IO resources. To achieve a scalable and flexible test access, a highly elaborate test access mechanism (TAM) is implemented to support multiple test instructions and test modes. Taking advantage of multiple identical cores embedding in the processor, scan partition and on-chip comparisons are employed to reduce test power and test time. Test compression technique is also utilized to decrease test time. To further reduce test power, clock controlling logics are designed with ability to turn off clocks of non-testing partitions. In addition, scan collars of CACHEs are designed to perform functional test with low-speed ATE for speed-binning purposes, which poses low complexity and has good correlation results.

Keywords DFT (design for testability), TAM (test access mechanism), multicore processor, low power test

1 Introduction

The Godson-3 microprocessor is a high-performance low-power multicore processor, which adopts a scalable mesh of crossbar (SMOC) network connecting 1 to 16 nodes and supporting 4 to 64 cores[1-2]. The 4-core Godson-3, as shown in Fig.1, connects four identical MIPS64 compatible RISC CPU processor ($P_0 \sim P_3$) and four identical globally addressed L2-cache ($S_0 \sim S_3$) through the level-1 crossbar (X1 switch). A directory-based cache coherence protocol keeps the coherence between CPU processor and L2-caches[3]. Two identical HyperTransports (HT) are employed to provide high speed inter-chip accesses. The HT links are full custom Serdes pins, each of them is designed with a throughput of 1.6 Gbps. Two identical DDR2/3 memory controllers (MC_0 and MC_1), one PCI controller, one LPC controller, and some low-end IO controllers are connected to the L2-cache through a level-2 crossbar (X2 switch). To improve yield and save power consumption, the chip is designed to support parts of the processor being in active, and the large capacity memories in the L2-cache have redundancy rows and support repairable feature.

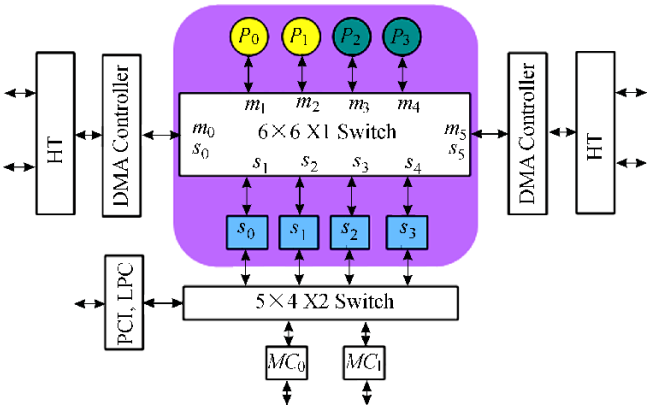

Fig.1. 4-core Godson-3 architecture.

The 4-core Godson-3 chip has been fabricated with ST 65 nm CMOS technology. It contains 425 million transistors and 650 thousand flip-flops. The die size is 174 square millimeters, and the max frequency is 1 GHz with about 10 W power consumption. A full hierarchical physical design flow is employed by assembling multiple copies of identical components. Those

Regular Paper

Supported by the National High-Tech Research and Development 863 Program of China under Grant Nos. 2008AA010901, 2009AA01Z125, 2009AA01Z103, the National Natural Science Foundation of China under Grant Nos. 60736012, 60921002, 60803029, 61050002, the National Basic Research 973 Program of China under Grant No. 2005CB321600, the Important National Science and Technology Specific Projects under Grant Nos. 2009ZX01028-002-003, 2009ZX01029-001-003.

components, like the CPU processors, the L2-caches, the DDR2/3 interfaces and the HT interfaces, are all designed in the SoC flow independently to form the reusable IPs. This makes identical cores have the same test feature and facilitates the reusing of test vectors.

The large scale and scalability of Godson-3 pose great challenges on the DFT design. Firstly, the number of transistors of Godson-3 increases dramatically, while I/O resource remains relatively unchanged. Elaborate DFT and test strategies must be adopted to reduce test time and test data volume while maintaining high test quality under the limitation of I/O resource. Secondly, test power at the chip level is very high. Without test power control, the maximum test power of the whole chip reaches about 50 W, which is unacceptable and may cause damage to the chip. Especially for transition test and path delay test, excessive power dissipation may make these delay tests invalid[4-5]. Thirdly, test access mechanism (TAM) should be scalable and flexible to facilitate the DFT design of the coming 8-core and 16-core and future 64-core chips. Thus, redefinition and redesign of DFT architecture will be minimized.

For large processors and complex SoCs, test partition[6] and modular SoC testing strategies[7] have been proposed in past few years. With a general wrapper methodology such as IEEE 1500[8-9], embedded cores are isolated and have no interaction with the outside, which gives us the opportunity to test cores one by one or test them simultaneously. The UltraSPARC processor[10], which has two cores, uses the lockup mode to give the two identical cores the same scan inputs. The responses of the two cores are compared internally and the mismatch is reported by an external pin. In the AZSCAN architecture[11], identical cores are tested by broadcasting the same test inputs to all of them. The responses of the cores are compared with the expect data which directly come from the external tester. For the quad-core AMD Opteron processor, an elaborate TAM with an on-chip comparison is presented[12]. It can enable comparing each core's responses to either the expected responses from the tester or to each other's responses. By loading complex commands, operations like scan shift, capture, and mask are executed within each core. It is scalable and more flexible, but the biggest problems are to prove it on silicon and to develop the corresponding software[13]. Though the large amount of identical cores in the multicore processor give a chance for test time and data volume reduction by testing identical cores concurrently, it makes the scan architecture more complex and limits the test diagnosis property of the chip.

On the other side, multicore processor like the Cell processor[14], the Niagra2 processor[15], and the UltraSPARC CMT processor[16] use the dedicated scan inputs and outputs for each core. This approach enables the diagnosis and determination of defective cores easier but leads to more test time since the length of scan chains is much longer. For the Godson-3 multicore processor, our purpose is to take advantage of both of them by trade-off among the ease of implementation, test costs and test throughput together.

The rest of the paper is organized as follows. Section 2 gives a brief overview of DFT techniques which meet the challenges mentioned above. The scalable TAM architecture is described in Section 3. In Section 4, the clock control scheme and timing issues for MBIST and scan partition test is addressed. Section 5 describes the scan architecture, including scan modes, scan test partition and on-chip comparison scheme for identical cores. Section 6 shows MBIST schemes for memories. Section 7 states the functional test application for speed binning. Conclusions are drawn in Section 8.

2 DFT Overview

As the high-end processor of the Godson series, Godson-3 multicore chip inherits many effective test techniques from our previous design[17]. Three on-chip clock (OCC) controllers are embedded in the chip to generate the high-speed capture clocks for delay test by using the internal PLLs for core clock, DDR2/3 clock and HT clock. During capture cycles, they are able to generate up to 4 high-speed clock cycles, which guarantee the effectiveness of delay test. To decrease test time and meet the restriction of given I/O pins, scan compression technique is employed to provide more than 10× compression ratio for each partition. The direct RAM access and debug mode, scan collar[17], is also devised for all memories by sharing the MBIST registers. Test patterns targeted for stuck-at faults, transition faults, path delay faults, bridging faults, and IDDQ faults are generated with high coverage to guarantee high test quality.

Besides the above features, Godson-3 employs a set of novel DFT techniques and test strategies to address the big challenges in such a large scale multicore chip with millions of transistors and multiple asynchronous clocks. The major techniques are listed as below.

• A scalable and hierarchical TAM is implemented by designing a flexible TAP architecture. The traditional IEEE1149.1 compliant TAP is exploited to support more than 60 instructions in multiple test modes.

• Partition scan test and partition ATPG are applied in Godson-3 to reduce test power and pattern generation time. We reuse boundary flip-flops of each partition as wrapper cells to isolate them from the rest of the design.

• To shorten test time and reduce I/O port requirement, we design data synchronous comparators (DSC) with X-mask features to compare scan test responses among identical cores.

• To lower MBIST power, self test of RAMs can run serially. The clocks of non-MBIST logics are all turned off. Only the clocks of MBIST logics and corresponding RAMs are turned on.

• Functional testing is applied on ATE for speed binning purposes. By shifting functional instructions and data through scan collar of L2 cache, real application instructions are executed and real-application critical paths are checked.

3 Scalable TAM Architecture

Though current Godson-3 is a 4-core version, the scalable SMOC architecture of Godson-3 imposes a need for a scalable and configurable TAM for 8-core, 16-core, etc., without changing the TAP structure. Fig.2 shows the proposed scalable TAM architecture with four nodes. For the 4-core Godson-3, only one node (with dark circle) exists, while 8-core and 16-core version are constituted by two nodes and four nodes respectively. Each node in the mesh includes four processor cores, four shared L2-caches, one DDR2/3 memory controller, one level-1 crossbar (X1 switch) and one level-2 crossbar (X2 switch)[1−2]. In the 4-core Godson-3, two HT links are connected to the level-1 crossbar, and two DDR2/3 memory controllers are connected to the level-2 crossbar to improve IO performance.

In Fig.2, TAP and TCU (test control unit) are located in the center of chip to control the test modes and allocate global test accessing channels. The test instruction register (TIR), which is accessed through the TAP, is conveyed from TCU to NCU. NCU (node control unit), sited in the node center, receives test in signals of the node from TCU and distribute them to certain block or collect test-out signals of certain block to TCU. BCU (block control unit), located in each partition, is used to control the switch of the partition's clock and buffer test in/out signals. For the purpose of the scalability and reusability, BCU in each partition is designed with the same structure. Because BCUs have the ability of buffering, the test in and test out data can be transferred in a pipelined way, which can avoid the broadcasting wire clusters and routing congestions throughout the whole chip.

By sharing scan channels in the hierarchical TAM architecture, each partition can be tested separately, and identical cores can also be tested concurrently with internal on-chip comparison DSC. To reduce the pipeline stages, partitions in one node are divided into two groups to receive/transmit test data from/to NCU by considering their locations in the floorplan. As shown in Fig.2, the sequences of the partitions in the two groups are depicted with the two arrow lines in the large circle. It further facilitates the physical design to transmit test in/out signals just between adjacent partitions. One long scan chain, debug chain, is designed to connect all scan registers of the whole chip for the purpose of

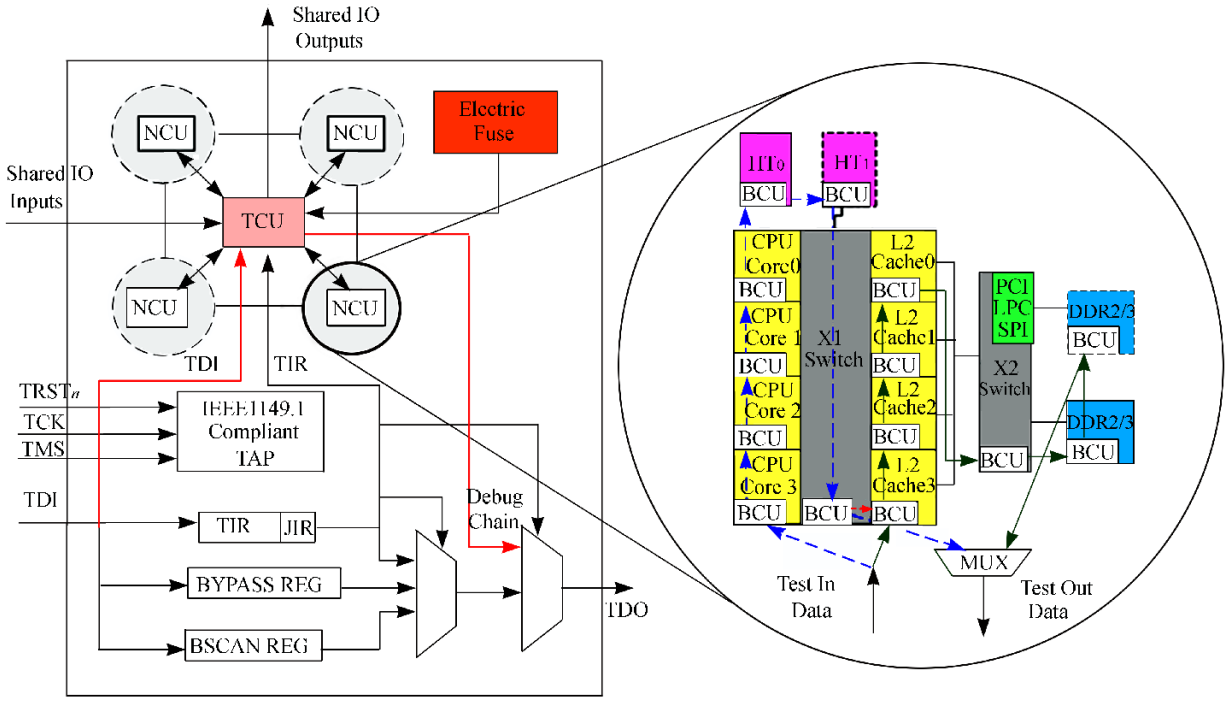

Fig.2. TAM architecture.

debugging and diagnosis. This debug scan chain can be accessed through IEEE1149.1 standard ports (TDI, TMS, TCK, TDO), and provides a direct way to observe all scan registers by shifting them out with the TCK clock. In the chip, electronic fuses are used to reconfigure the number of the running cores and deactivate the defective cores. The electronic fuses are also used for repairing RAM with redundancy rows by programming in the RAM fail addresses.

3.1 Boundary Scan Architecture

Compared with the traditional boundary scan architecture, the boundary scan circuitry in our design can provides more test instructions, and one additional data register called debug chain which includes all the scan registers can be accessed by the test access port (TAP). Besides the traditional TAP which consists of TDI, TDO, TMS, TCK and TRST, the instruction register (IR) and its associated decoder, our TAP also includes a TAP controller (TAPC) and several other data registers which is compatible with IEEE1149.1 standard.

The instruction register in boundary scan architecture is extended to have more bits to support more test instructions of scan test and MBIST. By loading IR with the test instruction through IEEE1149.1 standard ports, the chip can be set in corresponding test mode (scan mode or run BIST mode). The first 4 bits of IR are used as JTAG instruction register (JIR). The mandatory test instructions (BYPASS, SAMPLE, PRELOAD and EXTEST) which IEEE Std. 1149.1 defines are loaded into JIR. The left bits of IR are used as test instruction register (TIR). Scan test and BIST test related instructions are loaded into TIR. The instruction in IR is decoded by the decoder to generate the required control signals to properly configure the test logic in TCU, NCU and BCU.

The additional data register, debug chain, is similar with other data register like boundary-scan register and bypass register, and can be accessed by TAP directly. It does not affect the other data register. Debug chain provides the ability of control and observes the chip in on-line test and diagnosis.

3.2 Instruction Definition in TIR

TIR in our design has 40 bits to provide enough instructions for this multicore design. In 4-core Godson-3, approximate 60 instructions are used, while other instructions are reserved for future expanding uses. TIR is divided into 5 parts as shown in Fig.3. Node ID, Group ID and Partition ID are used to configure which partition in which group of which node will be tested. The parts of global test mode and local test mode are used to control which test mode the chip will enter into. For example, global test mode bits can be assigned to make the chip be in different test mode which can be scan test, MBIST test, boundary scan test, or functional test. Once the chip is in scan test, local test mode bits can be assigned to make the design in compress mode, internal mode, long-long mode, or debug mode (these test modes will be explained in Subsection 5.1).

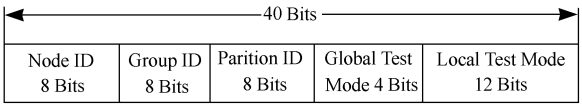

Fig.3. TIR definition.

There is a special instruction with all bits of TIR and JIR being logic "1", which indicates the chip in the functional mode and JTAG in bypass mode. After the system reset, all TIR and JIR bits are set to "1", and then the chip is ready to run the functional application. Test instruction must be loaded into TIR and JTR through test access port before corresponding test begins.

4 Clock Control

As is well known, test power has been a critical point in the multicore design. In our design, delay test, MBIST and functional test are all run at speed, which further aggravates the power consumption. To minimize test power, three-level clock gating cells are inserted in each partition for eliminating the switching power of non-testing logics. As shown in Fig.4, the level-1 (L1) clock gating cell is used to control whether to open or close the root clocks of the partition. Only the clocks of the partition under test are turned on, while all clocks of other partitions are closed. MBIST and functional logics have self-governed level-2 (L2) clock gating cells. The L2 clock gating cells are used to turn on/off the clocks of MBIST logics and the clocks of functional logics respectively. When MBIST runs, the clocks of functional logics are closed, and only the clocks of MBIST logic and the associated RAMs are pulsed. In the functional mode, the clocks of MBIST logics are keeping in "0" state and no extra switching power is consumed. Thus the test power is reduced significantly in both MBIST and functional mode. The level-3 (L3) clock gating cells are inserted globally in the chip during physical design. These clock gating cells can be controlled by ATPG tools which can use its power-aware ATPG feature to generate low power patterns. By using this hierarchy clock gating structure, power dissipation in functional mode and MBIST mode and scan mode can be reduced greatly.

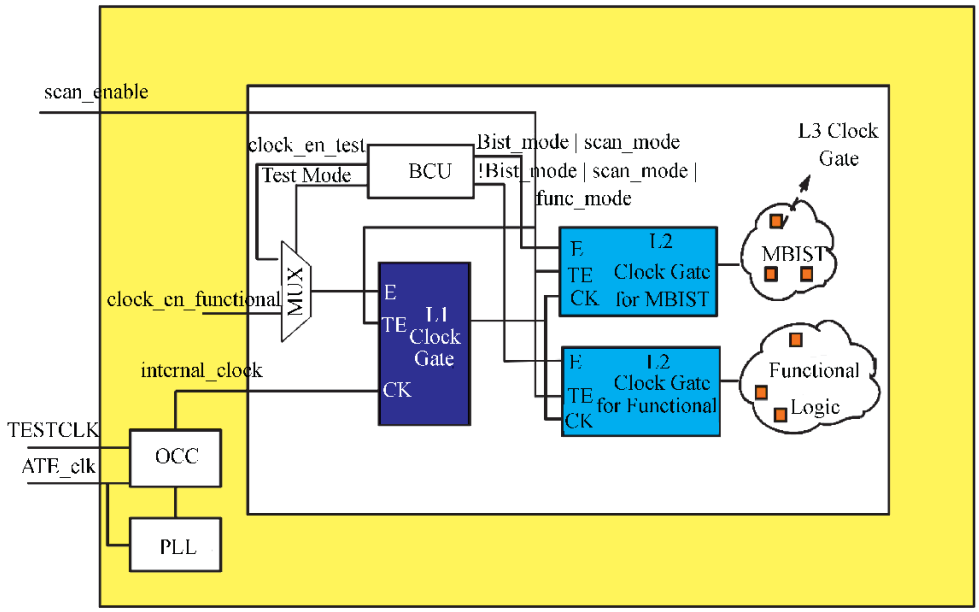

Fig.4. Clock controlling topology.

Timing issue especially hold-time constraint during scan shift period is also a big challenge in our design. Our chip has more than 10 asynchronous clocks and one partition usually includes multiple clocks. In clock tree synthesis and optimization process of physical design, the timing across clock domains is not optimized. The latency in two asynchronous clock trees can be very large, which means hold-time violations exist across timing domain. In order to make scan chains more balance, some scan chains include flip-flops of multiple clock domains. Timing violations across these clock domains must be fixed to guarantee the correctness of shift operation. Usually, fixing hold-time violations needs to insert delay cells or buffers. But if the latency among different clock is very big, the number of delay cells or buffers will be substantial, and extra area overhead will be incurred. To avoid fixing timing violations across clock domains, we use a dedicated clock "TESTCLK", which comes from the ATE directly, to shift scan data into all partitions. "Lockup" cells are inserted between flip-flops across clock domain in the same scan chain to prevent the hold-time violation. "Lockup" cells are negative latches which can provide half cycle hold-time margin. Thus, hold-time violations introduced by the different clock latency in scan shift operation can be eliminated. In this way, "TESTCLK" can drive all the scan flip-flops of different clock domains and guarantee the correctness of timing in scan shift period.

5 Scan Architecture

Scan test in Godson-3 is applied in a partition way. With this method, each partition of the chip can be tested separately by using dedicated wrapper chains. Since there are still some logics at top level that cannot be included into any partition, testing each partition separately will cause a little test coverage drop. To compensate for the coverage drop, multiple scan modes are designed in Godson-3.

5.1 Scan Modes

Each partition in the design supports four scan modes. These modes are compression mode, internal mode (IN mode), long-long mode (LL mode) and debug mode. In the compression mode, scan chain is designed to have less than 100 flip-flops to decrease scan shift cycles. Since scan compression structure can cause somewhat test coverage loss, internal mode is applied to improve test coverage based on the compression mode by bypassing the compression logics. In addition, from our experimental results, path-delay faults are hard to detect in compression mode. So, internal mode is used to generate path-delay patterns for detecting more critical paths. In the LL mode, internal chains are connected together to form longer scan chains with fewer scan input and output. Therefore, in the LL mode, dedicated scan in and scan out are possible for each partition. This provides a way to test the whole chip simultaneously with limited I/O resources, and the inter-partition logics are also tested in LL mode. In LL mode, 50 scan channels are used for testing all partitions in one node simultaneously. Those 50 scan channels are transferred to each core in a pipelined way. Each core occupies parts of 50 channels and does not share with the other cores. In LL

mode, the number of scan channels occupied by each partition is fully depended on the number of the scan registers it has. As mentioned before, there is a debug mode, in which all flip-flops in the whole chip are stitched into one debug scan chain. This mode provides a convenient way for detecting the failure point while running functional applications. If a failure occurs in a functional operation, debug mechanism will stop the clock and then shift out the values of the flip-flops to a register file. Checking the values of some key registers will give useful information and help to find the failure causes.

Fig.5 shows the four scan modes and the connections between two partitions. The dash lines that connect two short scan chains show the connecting of internal scan chains. Another kind of dash lines, which cross the MUX and XOR networks, show the way how internal scan chains form the LL scan chains. The scan chain "scan0" shows the scan connection in debug mode between two partitions. In the debug mode the previous scan out is selected for scan input of the next core. While in other test modes the global pipelined scan input channels are selected for scan inputs of the next core. The scan outputs of the two adjacent cores are selected in both internal and compression modes, while in the LL mode, scan outputs of each core are joined together.

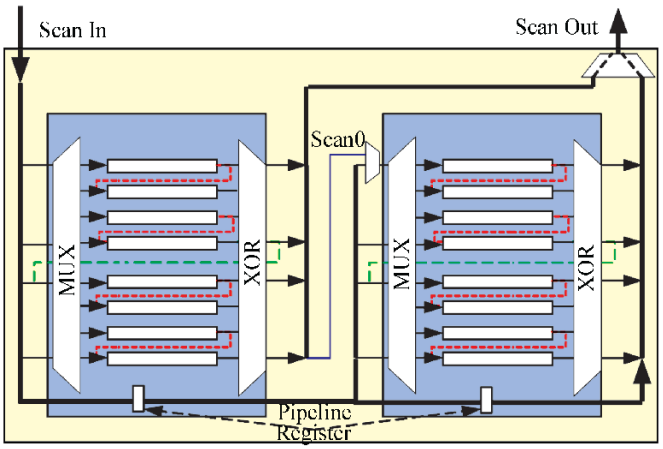

Fig.5. Four scan modes.

Different test modes are also designed for keeping low test power, especially during electronic wafer sort stage (EWS). In this stage, at-speed testing in LL mode could cause large switching activity at the whole chip level. From the power analysis tools, the TF patterns in the LL mode cause about 50 W power consumption in capture cycles. Such large power consumption is more likely to cause damage to the chip and cannot be tolerated. Therefore, test patterns are applied to one partition once a time, and a sequential test schedule is applied to avoid the over-heating of local region. In the final test, if there is a good condition for heat elimination, identical cores can be tested concurrently to reduce test time, and low-speed testing in the LL mode is also possible.

5.2 Scan Partition

It is clear that scan partition can help reduce the test time, test data volume and test power[6-7]. However, partitioning methods usually lead to a large area overhead by adding wrapper cells. The area overhead will be increased linearly with the partition number. In the Godson-3, the scan partition is applied in a low area cost way. By taking the advantage of the natural logic partitions, wrappers are inserted by reusing the existing flip-flops in each partition. This is possible because most partition pins connect directly to flip-flops. To further lower area overhead introduced by the wrapper cells, special wrapper cells are designed by adding capture enable pins to the boundary registers of partitions.

The wrapper cell used is composed by a common flip-flop with enable pin and an additional MUX as shown in Fig.6. A script is used to identify the boundary flip-flops of each partition, and change those cells to the special wrapper cells. The multiplexer added before the enable pin is used to select the functional enable signal and scan capture enable signal. The wrapper operates in INTEST, EXTEST and BYPASS modes. In INTEST mode, the input wrapper cells of the partition do not capture values and keep the shift-in value to provide the stimuli for the partition. While the output wrapper cells are enabled to capture the inner responses and then shift them out. In EXTEST mode, which indicates testing logics outside the partition, the input wrapper cells observe responses and the output wrapper cells provide stimuli for outside logics. In functional, LL and debug mode, the wrapper cell is in the BYPASS mode. All of the controlling signals are decoded from test instructions in each BCU.

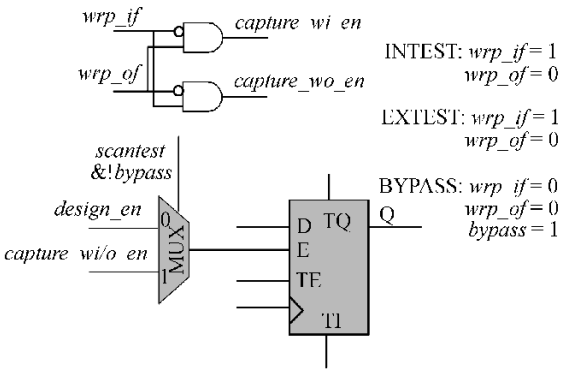

Fig.6. Wrapper cell.

Moreover, the input wrapper cells of one partition are reused as the output wrapper cells of the adjacent partition. In this way, the number of the wrapper cells is decreased significantly. In the 4-core processor, only 10 K flip-flops of the total 650 K flip-flops are transformed into the wrapper cells. Compared with the flip-flop without enable pin, the flip-flop with enable pin only adds 24% area. Together with the MUX added, about 90% of one flip-flop area is added in one special wrapper cell. Therefore, the 10 K wrapper cells area overhead is about 2% of the total 650 K flip-flops area.

5.3 ATPG and Power Results of Partitions

The Godson-3 multicore design is divided into 6 partitions, which are CPU core, L2-cache, HT, DDR2/3, X1 switch and X2 switch. Among them, only the CPU core, L2-cache, HT module and DDR2/3 module are inserted with wrapper cells. The two AXI switches share the wrapper scan chains with other partitions. Table 1 presents the scan chains and wrapper cells information of these partitions. Compression ratio is also given.

Table 1. Wrapper Information for Partitions

Partition	No. FF	No. I/Ps	No. O/Ps	No. Chains (IN/LL)	Chain Length (CP)	CMP* Ratio
CPU Core	63 152	202	364	48/6	90	12.2
L2-Cache	17 933	545	514	16/2	95	10.4
X2 Switch	48 725	1 506	954	28/3	90	16.1
X1 Switch	38 461	2 327	2 748	24/3	87	15.1
DDR2/3	32 522	338	154	20/3	87	15.0
HT	76 988	376	167	32/4	90	22.0

CP: compression mode, CMP ratio: compression ratio

Test patterns of the 6 partitions are generated independently in core level with a smaller design scale. Thus, the ATPG run time is reduced greatly. Since each partition always finishes their physical design before integrated into the top chip in full hierarchy physical design flow, their pattern generation time could overlap with the top physical design cycle. Chip design cycle is also decreased. A tool is developed to translate the core-level patterns to chip-level patterns. So pattern generation time and simulation time are decreased significantly. Patterns generated on each partition are simulated again at the chip level to guarantee their correctness. To speed up pattern simulation, partitions that are not tested are made empty modules. Thus, memory used and run time is further decreased.

Since only the clock of the partitions under test is open as described in Section 4, partition test power can be controlled. Scan test patterns for each partition include the stuck-at test pattern, transition test pattern, path-delay test pattern and bridging test pattern. Those patterns only target the faults in each partition. For the stuck-at faults in inter-partitions, additional patterns are generated in LL mode to improve test quality. Fault-grading procedure is adopted in scan test pattern generation flow to further reduce test data volume. Table 2 presents the total pattern counts and pattern test time for each partition in compression mode. The test time of each partition is calculated according to pattern counts, the length of scan chain and the shift clock period. The time of loading test data into ATE is not included.

For each partition, high test coverage is achieved. The stuck-at fault coverage of each partition ranges from 98.3% to 99.3%, and the transition converge ranges from 84.4∼90.4%. 10 K∼30 K critical paths are selected for each partition, and there are totally more than 18 K critical paths are detected in the internal mode. The power values listed in Table 2 are estimated with commercial tool Prime Power in the fast corner for transition patterns during capture cycles, which determine max power consumption. Moreover, test pattern volume still remains the same as shown in Table 2 with the increase of the number of nodes. If testing identical cores concurrently, the test time will be further reduced.

Compared with the whole chip test, test power and pattern generation time are decreased dramatically. As mentioned in previous subsection, we also estimate the TF patterns in LL mode during capture cycles, and find

Table 2. ATPG Results for Each Partition

	Pattern Count	Test Time (ms)	Max Power (W)	STA Cov. (%)	TF Cov. (%)	PD Paths (DT/TOT)
CPU Core	10 639	43.3	7.76	98.9	88.4	6235/30000
L2-Cache	6 985	28.1	4.96	98.3	84.4	2030/9971
X2 Switch	6 528	32.1	3.18	98.4	85.5	4771/10000
X1 Switch	5 841	29.1	5.98	99.3	88.9	3873/10000
DDR2/3	9 634	38.2	4.94	98.9	90.4	ND
HT	16 360	60.8	5.43	99.1	86.3	1873/10000
Summary	55 987	545/232*	7.76**	98.8	86.2	18782/69971

* 545 ms is test time of the 4-core chip with each identical core testing independently, 232 is the total test time of the partitions.

** 7.76 W is the max power of partition scan test.

the max test power is 50 W, which is about 7 times of the partition test listed in Table 2. Furthermore, test pattern generation time for such a large scale design at top level is tremendously long even with very high performance machine. Such ATPG time cannot be tolerated. All these problems are well resolved in our partition test.

5.4 Data-Synchronous-Comparator (DSC) for Identical Cores

Based on the fact that the isolated identical cores should have the same responses to the same test stimuli, the output responses can be compared with each other to reduce test time and test pattern volume significantly. In [12], a test access mechanism, which supports three test modes for multiple identical cores, the full-rate self-compare mode, the interleaved self-compare mode and inter-core compare mode, is described. Although it is flexible and scalable, pattern transformation and issuing proper TAM instructions is complex, and the support for X-masking will cost extra test time. In our design, an easily implemented DSC architecture is presented to reduce the pattern generation and pattern translation complexity.

In the DSC architecture, every identical core receives the same scan in data at the same time, and outputting the responses of any core needs the same number of cycles. All the identical cores are equivalent from outside and are data-synchronous, so we call the on-chip comparator adopted in our design data-synchronous-comparator (DSC). Only one core's output responses can be shifted out and observed directly on ATE. Responses of other cores are compared with that of the given core whose "*core_sel_out*" signal is 1, and the corresponding "*Err_flg*" signals give the comparison results. The comparison is proceeding in a cycle-by-cycle basis. If any core mismatches with the direct observe core during shift cycles, "*Err_flg*" signal gives out a high value at that cycle.

Fig.7 depicts the DSC architecture for four identical cores. Logics around the cores are the major part of the BCU which controls scan input and scan output of the cores. The same BCU structure is instantiated in each core, which provides the flexibility for extending the number of cores. In the debug mode, all flip-flops in the chip form one debug scan chain and the input stimuli are selected from the output of previous core. In other test modes, each identical core in the group shares the scan-in data in a pipelined way, and on-chip comparisons are launched between the core and the directly observed core. The signal "$core_x\_sel\_out$" controls whether test responses of the core are shifted out to ATE. If it is 1, the core's scan out data can pass the "AND" gate. If it is 0, the responses of the core are blocked with 0. So the responses of the core whose "*core_sel_out*" is 1 can pass all "AND" and "OR" gates and be directly observed on ATE. Meanwhile, the responses can be compared with those of other cores as a criterion.

In order to reduce design complexity and avoid tough control, extra complex control logics are excluded. Unlike [12], we use balance registers to achieve straightforward on-chip comparisons. As shown in Fig.7, there are four stages of scan input balance registers (SIBR) and zero stage of scan output balance registers (SOBR) for core_0, and three stages of SIBRs and one stage of SOBR for core_1, and so forth. Generally, for n identical cores, core_m has $n-m$ SIBRs and m SOBRs. Thus all the cores have the same number of scan input pipeline registers and scan output pipeline registers. In this way, all the identical cores are equivalent from the view of the tester. They receive the same stimulus at the same cycle and enter the capture procedure simultaneously. Therefore commands which control the specific operations in [12] are not needed any more. In addition, test patterns of testing one core separately are the same with that of testing all the cores concurrently. That is to say, only one set of patterns are needed to test any of the identical cores individually and test all the cores simultaneously.

Another issue to discuss is that which core should be directly observed. In our design, it depends on the connecting sequence of the identical cores. For instance, in Fig.7, core_0 should be the first to be observed directly. If core_0 fails in the test process, core_0 is bypassed and

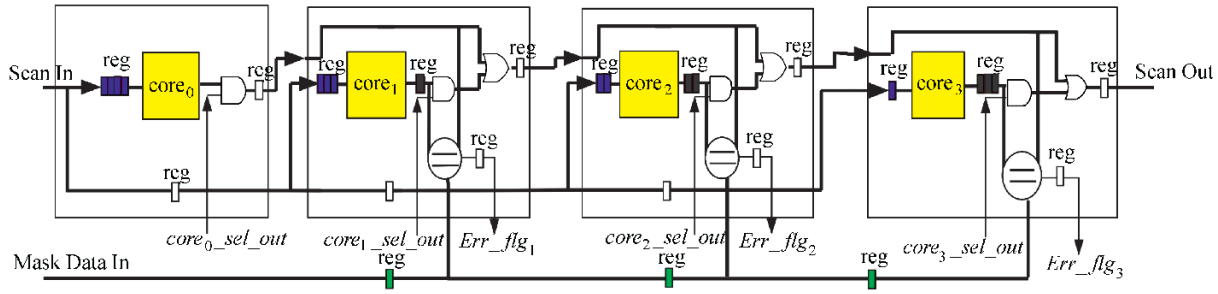

Fig.7. DSC architecture.

$core_1$ is then observed from ATE, and so on. The "*Err_flg*" signal of the directly observed core does not reflect the comparison results and is masked in the process of test. The average number of experiments to determine pass/fail status of each core is same as the inter-core compare mode in [12], which is a function of the per core yield.

The DSC techniques are adopted in Godson-3 quad-core processor, and the test time is reduced from 545 ms to 232 ms. The power consumption of testing 4 identical cores concurrently for stuck-at patterns is 8 W, which is still acceptable. However, transition test patterns and path delay test patterns are carefully considered in implementation of concurrent testing to avoid high power consumption.

5.5 *X*-Mask and Pattern Translation

In our DSC architecture, the X-mask feature is easily achieved by increasing extra scan input channels. Mask bits are inputted into the comparators through these channels to mask the comparisons with X bits. Although the additional channels cost extra I/O resources, the cost can be compensated by using asymmetric hardware compression structures with less scan output channels which determine the number of mask channels. This feature has been supported quite well by commercial EDA tools without lowering test coverage much. Because the responses as the criterion arrive at different cores at different clock cycles, the start time of comparison for each core is different. In Fig.7, $core_1$ is the first to start comparison and $core_3$ is the last. SOBRs are also used to adjust the comparison time. In general, $core_m$ starts comparison after m clock cycles, which requires that the mask bits should be "1" in the first m cycles for $core_m$ to mask the corresponding comparisons. It is too late to scan input these mask bits because pipeline registers must be passed through, and scan in mask bits cannot arrive at the comparator directly. Special pipeline registers which are shown in Fig.7 are adopted to tackle this problem. These special registers are in the "TESTCLK" domain and can capture a value of "1" in the "test setup" and capture procedure (or when the "*scan_enable*" signal is not asserted), which can be easily realized by adding an MUX. Thus in the first m cycles, the contents of these special registers are used to carry out the function of mask; after m cycles, the scan input mask bits are used. Compared to [12], X-mask in our design does not cause extra test time and a high tolerance of X bits is achieved.

Test pattern generation can also benefit from the DSC scan structure. Patterns can be generated by commerce ATPG tools and simulated at core-level, which can avoid the biggest challenge of developing software. Then the validated patterns are translated to chip-level to test all the identical cores concurrently or any of them separately by simply adding some pipeline stages. The ATPG time and pattern debug time can be reduced greatly in this way. An example of pattern translation for 4 identical cores is represented in Fig.8. After every shift procedure, the value of "1" is loaded into every special pipeline registers to mask the first four scan out data in the next pattern. If the following capture procedure pulses "TESTCLK", these special registers can capture "1", otherwise the contents are kept. In a word, the contents of the special registers can be guaranteed to "1" before the next shift procedure to make sure the correctness when compared with the value stored in the SOBRs.

6 Memory BIST

There are totally 496 memories of different sizes and different types embedded in the 4-core Godson-3. The mMarchLR14 and March 17N algorithms are adopted to test the various memories. The two algorithms are shown in the Fig.9.

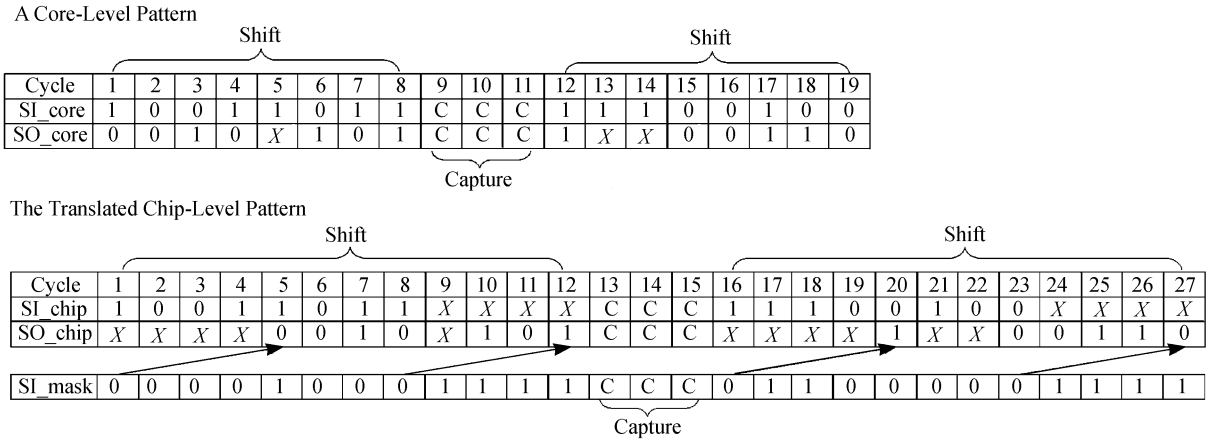

Fig.8. Pattern translation and mask inputs.

$\{\Updownarrow (w_0); \Uparrow (r_0, w_1); \Uparrow (r_1, w_1, r_0, w_1); \Downarrow (r_1, w_0); \Downarrow (r_0, w_1, r_1, w_1); \Uparrow (r_0);$

(a)

$\{\Updownarrow (w_1); \Uparrow (r_1, w_0, w_1, r_1, w_0); \Uparrow (r_0, w_1, w_1); \Downarrow (r_0, w_1, w_0, w_1); \Downarrow (r_1, w_0, w_1)$

(b)

Fig.9. (a) mMarchLR14 algorithm. (b) March 17N algorithm.

Our design supports a set of BIST instructions which need to be loaded into TIR to set the chip in the corresponding test mode. With these instructions, BIST can run both in chip level and in partition level. The clocks of partitions without memory macro like $X1$ switch, $X2$ switch and DDR are turned off in all MBIST test modes. Under low power budget, memories can be tested sequentially one partition by one partition. Moreover, memories inside one partition can also be tested in serial or parallel. If test time is a main concern, MBIST of all partitions can run concurrently.

We use SYNOPSYS Prime Power tool to accurately analyze power dissipation in different test mode. The power consumed in different MBIST test mode is shown in Table 3. From Table 3, we can see the power dissipation in partition level is low, and power dissipation increases linearly as the number of partitions increases. Memories can be tested in serial, in partial parallel, or in full parallel mode according to power and test time budget. All BIST modes make memory test in our design more flexible.

Table 3. Power of MBIST

Partition	Leakage Power (W)	Peak Power (W)	Total Power (W)
CPU $\text{core}_{0\sim3}$ independently	1.68	5.87	4.64
CPU $\text{core}_{0\sim3}$ concurrently	1.68	17.08	13.36
L2-$\text{cache}_{0\sim3}$ independently	1.68	5.19	4.53
L2-$\text{cache}_{0\sim3}$ concurrently	1.68	15.51	12.48
$\text{HT}_{0\sim1}$ independently	1.68	3.07	2.57
$\text{HT}_{0\sim1}$ concurrently	1.68	4.46	3.65

The results of MBIST are stored in certain flip-flops, and can be directly observed through the external pins by reusing the functional I/Os. Since there is limited number of reusable I/Os, MBIST shares the same I/Os with scan test in the defined test mode. If further diagnosis is needed, the results of MBIST can be scanned out through scan chains. The memories in the L2-cache have two redundancy rows for every 128 rows. BIST for these kinds of RAMs can record fail addresses and shift them out. The shifted fail addresses are then programmed into electric fuses to make redundancy rows replace the defect rows. BIST for all memories runs at the same frequency as the functional mode, which aims to detect at-speed delay faults. All memories in the processor are equipped with the scan collar engines[17]. Through scan collar, direct memory access and diagnosis are achieved.

7 Functional Test

Though at-speed structural test provides an efficient way for testing delay faults[18], at-speed functional test is still required for speed-binning in our processor. Functional testing has good portability properties especially for scalable multicore design. The functional testing programs developed on one core can be easily applied to the multicore design with a little change and still remain good correlation results. Moreover functional testing is fully reflecting the real application running on the processor.

There are some concerns to address in implementation of the functional test. One is how to perform functional testing and observe the results with low speed ATE. Scan collar chains are used to resolve the problem. With the help of scan collar, we can shift the functional instructions and initial data into the L1-cache and L2-cache at low scan shift speed. After the instructions and the data are ready in caches, the instructions in I-cache can be executed at the functional speed. Because the memory is not available at this time, all the instructions and data the functional program needs must exist in caches, which mean cache miss must not occur in the functional program. The program length is constrained by the size of caches. When the functional program finishes, the results are written into L2-cache and internally checked. In case of correct results, one output pin is asserted to "1", while in case of wrong results, this output pin is asserted to "0". ATE can check this status pin to decide whether the chip can run at certain frequency. Furthermore, the program results can be downloaded to the tester from L2-cache through scan collar chains for diagnosis and debug purposes. Generally, the time of loading required instructions and data into caches through scan collar chains is much longer than the running time of the associated functional program. In our design, they are 30 ms and 1 ms respectively.

Another concern is how to evaluate the effectiveness of the functional program. We use the percentage of critical paths the program can detect to measure the effectiveness. We use the static timing analysis tool Prime Time to generate 500 K critical paths and develop a tool to evaluate whether the program can cover these paths. Experimental results show that the functional program we develop can trigger and cover about 200 K critical paths, and most of these paths are memory-related paths which cannot be detected by

path delay patterns. As shown in Fig.10, comparing the real application results, we find the at-speed functional testing has a good correlation results. In most cases, the frequency difference of functional testing and real application are within 80 MHz. To get a more accurate speed-binning result, the correlation like that in [19] is further needed.

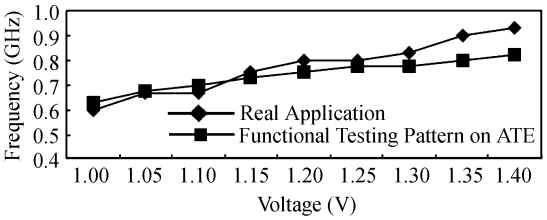

Fig.10. Functional testing correlation.

8 Conclusions

As the number of the cores increases in the multicore chip, test power and test time have been a major concern. To reduce power consumption, partition scan test and hierarchical clock gate controlling techniques are employed. With these techniques, the maximal test power is reduced from 50 W to about 8 W. To reduce test time, a simple and easy to implement DSC architecture is proposed to realize on-chip comparisons between identical cores. With DSC, test time can be reduced by almost 40%, while test data volume remains unchanged. Meanwhile, to minimize redesign and re-verification cost, a highly configurable and scalable TAM architecture with extendable test instructions is adopted. The flexible TAM can be easily carried on to the 64-core Godson-3 design. Thus, test target with shorter time, lower power and less test data volume at low test cost for the Godson-3 multicore chip is achieved.

References

[1] Hu W, Gao X, Chen Y *et al.* Micro-architecture of Godson-3 multi-core processor. In *20th Hot Chips*, Stanford, USA, Aug. 24-26, 2008, Slides.

[2] Hu W, Wang J, Gao X *et al.* Godson-3: Scalable multicore RISC Processor with x86 emulation. *IEEE Micro*, 2009, 29(2): 17-27.

[3] Hu W, Shi W, Tang Z. JIAJIA: An SVM system based on a new cache coherence protocol. In *Proc. the High-Performance Computing and Networking Europe* (*HPCN 1999*), Amsterdam, The Netherland, Apr. 12-14, 1999, pp.463-472.

[4] Liu H, Li H, Hu Y *et al.* A scan-based delay test method for reduction of overtesting. In *Proc. the 4th IEEE International Symposium on Electronic Design, Test & Applications*, Hong Kong, China, Jan. 23-25, 2008, pp.521-526,

[5] Saxena J, Butler K M, Jayaram V B *et al.* A case study of IR-drop in structured at-speed testing. In *Proc. IEEE International Test Conference*, Charlotte, USA, Sept. 28-Oct. 3, 2003, pp. 1098-1104.

[6] Sehgal A, Fitzgerald J, Rearick J. Test cost reduction for the AMD^{TM} Athlon processor using test partitioning. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 21-26, 2007, Article No.1.3.

[7] Waayers T, Morren R, Grandi R. Definition of a robust modular SOC test architecture, resurrection of the single TAM daisy-chain. In *Proc. IEEE International Test Conference*, Austin, USA, Nov. 8, 2005, pp.610-619.

[8] DaSilva F, Zorian Y, Whetsel L *et al.* Overview of the IEEE P1500 standard. In *Proc. IEEE International Test Conference*, Charlotte, USA, Sept. 28-Oct. 3, 2003, pp.988-997.

[9] Sehgal A, Goel S K, Marinissen E J *et al.* IEEE P1500 compliant test wrapper design for hierarchical cores. In *Proc. IEEE International Test Conference*, Charlotte, USA, Oct. 26-28, 2004, pp.1203-1212.

[10] Parulkar I, Ziaja T, Pendurkar R *et al.* A scalable, low-cost DFT architecture for UltraSPARC chip multiprocessors. In *Proc. IEEE International Test Conference*, Baltimore, USA, Oct. 7-10, 2002, pp.726-735.

[11] Makar S, Altinis T, Patkar N *et al.* Testing of Vega2, a chip multi-processor with multiple cores. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 21-26, 2007, Article No.9.1.

[12] Giles G, Wang J, Sehgal A *et al.* Test access mechanism for multiple identical cores. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 28-30, 2008, Article No.2.3.

[13] Wood T, Giles G, Kiszely C *et al.* Test features of the quad-core AMD Opteron^{TM} microprocessor. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 28-30, 2008, Article No.2.1.

[14] Riley M, Bushard L, Chelstrom N *et al.* Testability features of the first-generation CELL processor. In *Proc. IEEE International Test Conference*, Austin, USA, Nov. 8, 2005, pp.111-119.

[15] Molyneaux R, Ziaja T, Kim H *et al.* Design for testability features of the SUN Microsystems Niagara2 CMP/CMT SPARC Chip. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 21-26, 2007, Article No.1.2.

[16] Parulkar I, Anandakumar S, Agarwal G *et al.* DFX of a 3rd generation, 16-core/32-thread UltraSPARC^{TM} CMT microprocessor. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 21-26, 2008, Article No.2.2.

[17] Wang D, Fan X, Fu X *et al.* The design-for-testability features of a general purpose microprocessor. In *Proc. IEEE International Test Conference*, Santa Clara, USA, Oct. 2006, Article No.9.2.

[18] Belete D, Razdan A, Schwarz W *et al.* Use of DFT techniques in speed grading a 1 GHz+ microprocessor. In *Proc. IEEE International Test Conference*, Baltimore, USA, Oct. 7-10, 2002, pp.1111-1119.

[19] Zeng J, Abadir M, Vandling G *et al.* On correlating structural tests with functional tests for speed binning of high performance design. In *Proc. IEEE International Test Conference*, Charlotte, USA, Oct. 26-28, 2004, pp.31-37.

Zi-Chu Qi received her B.S. and M.S. degrees from Zhejiang University in 1999 and 2002, and her Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2010. She is now a lead DFT designer in the Godson projects. Her research interests include FPU design, DFT design and VLSI design.

Hui Liu received her B.S. degree in computer science from Shandong University in 2005, and her M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2008. She is now a senior DFT engineer in the Institute of Computing Technology. Her research interests include design for testability, overtesting and delay testing.

Xiang-Ku Li received his B.S. degree from Jilin University in 2006, and his M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2009. His research interests include VLSI design, and DFT design.

Wei-Wu Hu received his B.S. degree from the University of Science and Technology of China in 1991 and his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 1996, both in computer science. He is currently a professor in the Institute of Computing Technology. His research interests include high performance computer architecture, parallel processing and VLSI design.