Design-for-testability for system LSIs

Masanori Ushikubo

A system LSI is a large-scale circuit comprising a logic core, memory core, analog core, and the like, which means that ensuring sufficient test quality in LSI function tests is a difficult challenge. At Oki, we have adopted Design-for-testability with the object of increasing test quality whilst reducing the test design TAT (Turn Around Time). Recent progress in miniaturization, on the other hand, has led to sharp increases in the number of embedded cores and the size of LSI circuitry, and this has in turn given rise to longer test times.

We have developed and implemented design automation tools to support our Design-for-testability and achieve more than 50% reductions in test times for system LSIs. These tools not only cut test time, but also comprise functions for suppressing increases in chip area and power consumption, and speed reductions, all of which form barriers to design-for-testability. They also have functions for optimizing the number of test pins. This essay reports on out Design-for-testability for reducing test times, the design automation tools and design flow we have developed and implemented to put these procedures into effect, and the corresponding reductions we have achieved in test time.

Design-for-testability for cutting test time

As shown in Fig. 1, the problem in testing system LSIs is that the test time expands as the scale of the circuitry increases. Oki has managed to develop a procedure for reducing test times, without sacrificing test quality for system LSIs, by expanding the functionality of existing design-for-testability techniques (Table 1), to enable time savings of over 50% (Fig. 1). Below, we shall look at two of the related techniques: (1) scan rationalization, and (2) concurrent test design.

(1) Scan rationalization

A scan 1) is a process which facilitates the generation of a test pattern by adding paths (hereinafter, called “scan chains”) which directly control and observe the flip-flops inside the circuitry (hereinafter, called “scan flip-flops”) via pins. Fig. 2 shows a scan circuit and scan test sequence.

A scan test consists of a scan shift operation for activating the scan chains in order to control and observe the scan flip-flops directly via the scan input and output pins, and a parallel operation for controlling and observing the combinational circuit via the user input and output pins. The scan test

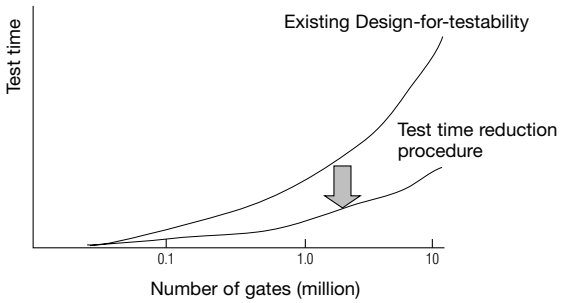

Fig. 1 Gate number vs. Test time

Table 1 Design-for-testability

Design-for-testability	Test object		Test time reduction procedure
	Circuit	Test item	
Full scanre	Random logic, logic core	Functional test	Scan rationalization
Memory BIST	Memory core	Functional test	Concurrent test design
Isolated test	Logic core (unscanned), memory core, analog core	Functional test	Concurrent test design

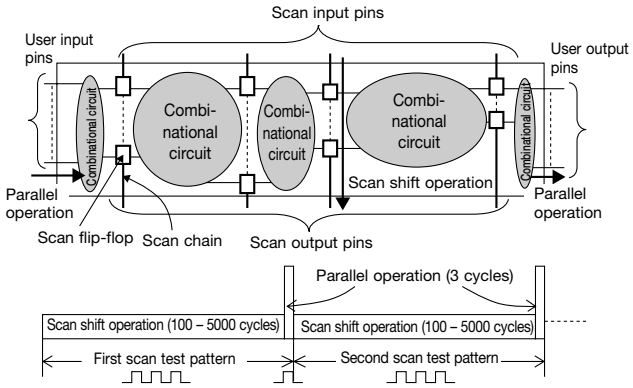

Fig. 2 Scan circuit and scan test sequence

sequence involves repetition of the scan shift operation and parallel operation, and the corresponding number of repetitions gives the number of scan test patterns. The total number of test cycles of the scan circuit (hereinafter, called the "number of scan test cycles") is effectively the product of the number of scan test patterns, and the number of scan shift cycles. Therefore, by reducing either the number of scan test patterns or the number of scan shift cycles, it is possible to cut the number of scan test cycles, and consequently, the test time.

The design automation tools developed and introduced by Oki generate circuits that reduce the number of scan test patterns and the number of scan shift cycles. By adopting these scan circuits, scan test time can be decreased without affecting test quality. The circuit to be scanned has a random logic and scannable logic core (hereinafter, called "scan core").

(2) Concurrent test design

Concurrent test design is a method for testing a number of cores and circuits in the LSI, in parallel.

In order to reduce the functional test time of the logic core, which cannot be scanned, (hereinafter, called "unscanned core"), and the memory core, concurrent testing is applied to the existing memory BIST (Built-In Self-Test) 2) and isolated test techniques used for these cores. Concurrent testing is not only applied to the cores, but also to the scan circuit combining the scan core and the random logic, alongside the unscanned cores and memory cores.

We have developed and introduced a design automation tool for concurrent testing, which generates memory BIST circuits and isolated test circuits (for controlling and monitoring the cores directly via pins). Use of this test circuit also permits test time reductions without loss of test quality. Another tool developed by Oki generates test patterns for realizing concurrent testing.

Design automation tools for aiding scan rationalization

In order to rationalize scanning, we have developed and introduced two design automation tools: a tool for inserting circuits to reduce the number of scan test patterns, and a tool for balancing the scan chains.

Fig. 3 is a model view of a scan circuit generated using these tools.

The main function of each tool is listed below.

(1) Tool for inserting circuits to reduce number of scan test patterns

This tool inserts scan test pattern reducing circuits in a circuit previously converted to a scan circuit, at deep logic levels of the combinational circuit.

- Considering internal structure of hard core

If there are deep logic levels in the combinational circuit between the interior of the hard core and the other scan circuits (random logic, other scan cores, etc.), then a scan test pattern reducing circuit is inserted outside the hard core (Fig. 3).

- Limiting increase in the chip surface area

Increase in the chip surface area can be suppressed by indicating the gate size of the scan test pattern reducing circuit. However, even if no such indication is made, the increase in the number of gates will be around 1%.

- Limiting speed reduction

Since the scan test pattern reducing circuit is connected to the net between cells, rather then being inserted into the net, the fanout number of the net

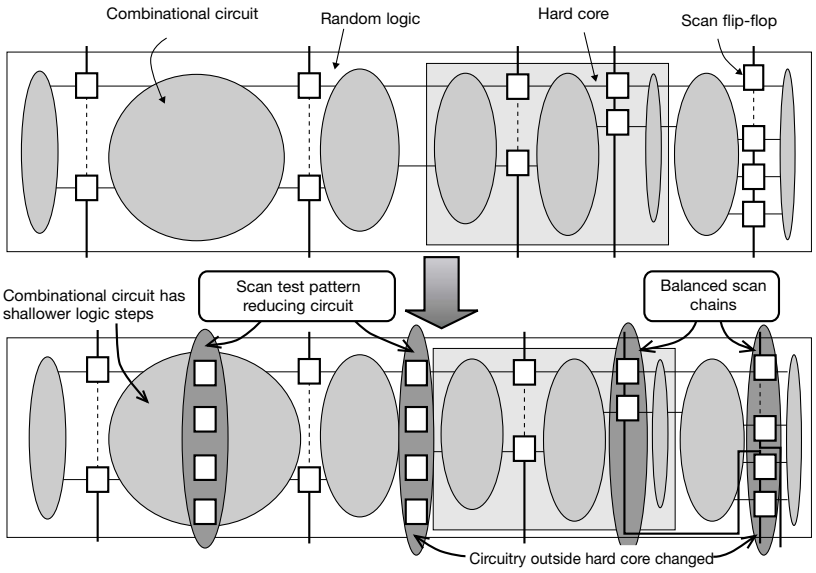

Fig. 3 Circuit model of scan rationalization

only increases by one, so it causes minimal reduction in the system speed.

- Limiting increase in power consumption

By inserting a circuit for stopping the operation of the scan test pattern reducing circuit, it is possible to restrict increase in power consumption.

(2) Scan chain balancing tool

This tool is used to cut the number of scan shift cycles. The number of scan shift cycles is equal to the maximum number of scan flip-flops per chain. Therefore, by averaging the scan flip-flops in each scan chain, it is possible to reduce the number of scan shift cycles. In order to balance the scan chains, this tools reconstructs the scan chains of circuits that have already been converted to scan circuits.

- Considering internal structure of hard core

Scan chains inside the hard core are included with the other scan circuits for balancing (Fig. 3).

- Multiple-block compatibility

A circuit is inserted to avoid timing problems if scan flip-flops of different clock systems are included in the same chain.

Design automation tool for concurrent test design

In order to achieve concurrent testing of different cores, we have developed and introduced three design automation tools: a memory BIST circuit and BIST startup test pattern generating tool, a memory BIST circuit and isolated test circuit inserting tool, and a concurrent test pattern generating tool.

Fig. 4 shows a circuit model where a memory BIST circuit and isolated test circuit for concurrent testing have been inserted using these tools.

The main functions of the respective tools are as follows.

(1) Memory BIST circuit and BIST startup test pattern generating tool

By specifying the word and bit structure of the memory, this tool generates an RTL (Register Transfer Level) memory BIST circuit for achieving concurrent testing, and a test pattern for starting up the BIST circuit.

- Compatibility with memory cores of various structures

The tool is able to generate BIST circuits for concurrent testing of several memory cores having different word and bit configurations.

- Limiting increase in power consumption

A circuit is inserted which halts operation of the BIST circuit when the system is running.

(2) Memory BIST circuit and isolated test circuit inserting tool

This tool inserts a memory BIST circuit and an isolated test circuit for concurrent testing of embedded cores, in an RTL description circuit.

- Limiting increase in chip surface area and reduction in speed

Since the test circuit is inserted in an RTL description circuit, the logic can be synthesized simultaneously with the system circuit, which means that the chip area and speed can be optimized.

- Limiting increase in power consumption

Isolated test circuits for sequential testing of particular cores can be inserted, to account for the increased power consumption involved in running a number of cores.

- Optimizing the number of test pins

When inserting isolated test circuits for concurrent testing, the total number of ports of the unscanned cores plus the number of memory BIST ports is kept to a value less than the number of LSI pins.

(3) Concurrent test pattern generating tool

This tool inputs the test pattern generated by the BIST startup test pattern generating tool, and the unscanned core test pattern and scan test pattern, and edits the patterns on the basis of wiring information for the inserted test circuits, to create a concurrent test pattern. The unscanned core test pattern is supplied by the core designer, and the scan test pattern is generated by the scan test pattern generator tool.

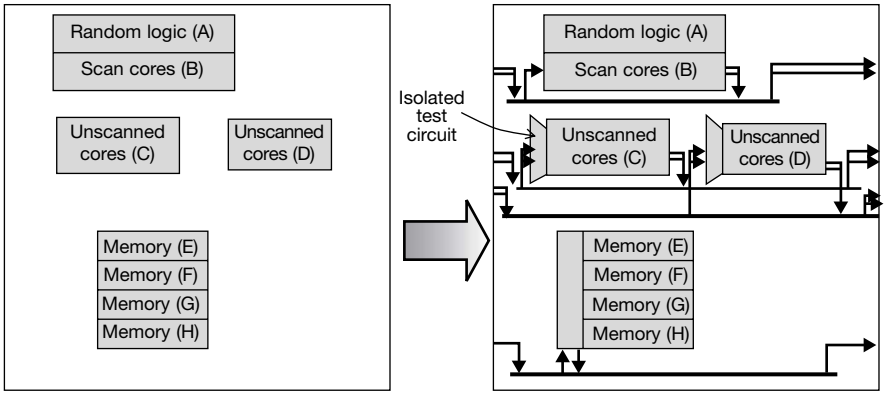

Fig. 4 Schematic circuit for concurrent testing

Design flow using design automation tools for reducing test time

Fig. 5 illustrates a design flow incorporating the design automation tools developed and introduced by Oki. The hatched areas indicate tools we have developed, which can be adopted easily without altering the conventional design flow.

With regard to the TAT for test design, although the scan rationalization process is a new design task compared to previous scan design methods, we have been able to restrict the corresponding increase in TAT to about 1 day, by employing our two design automation tools. Furthermore, concurrent testing, based on the three automated design tools described above, allows design tasks that previously took 1 to 2 weeks to be completed within 2 days.

Test time reductions

Fig. 6 shows the actual reductions in test time achieved when the procedure developed and introduced by Oki is applied. Scan rationalization allows scan test time to be halved. In Fig. 6, the test times for random logic (A) and scan cores (B) are both reduced by 50%. Furthermore, functional test times can be reduced by using concurrent testing of scan circuits combining random logic (A) and scan scores (B), as well as unscanned cores (C), unscanned cores (D) and memory cores, whilst taking account of the number of test pins, and the like. If scan rationalization is combined with concurrent testing, then test time reductions exceeding 50% can be achieved.

Conclusion

The design-for-testability we have applied in Oki has allowed us to halve test times without losing quality in system LSI testing, by introducing modifications into our scan circuits, memory BIST circuits and isolated test circuits. In order to adopt this method, we have developed design automation tools with functions for reducing the problems which might be expected to arise with test-oriented design. These tools have not only enabled easy application to system LSIs, but have also prevented any increase in test design TAT, due to their good affinity with existing design flows.

The procedure and tools we have devised are already being used in design operations, with excellent results. Oki is committed to reducing test time even further, in the future.

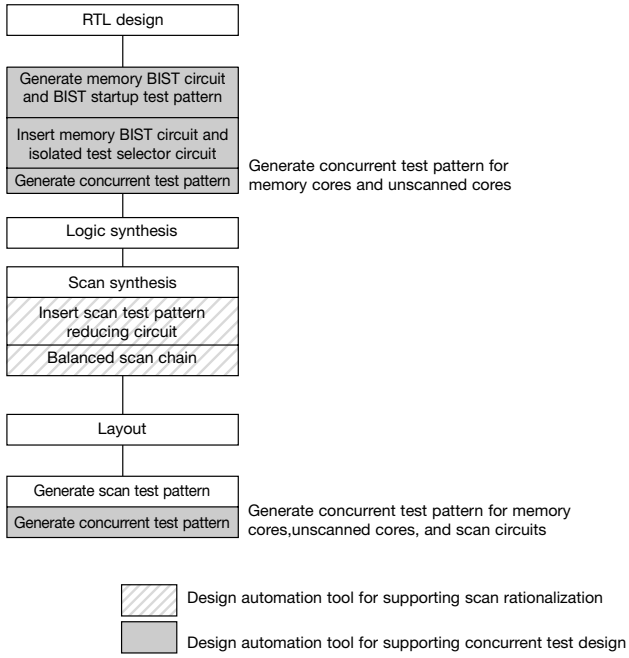

Fig. 5 Design flow using design automation tools for reducing test time

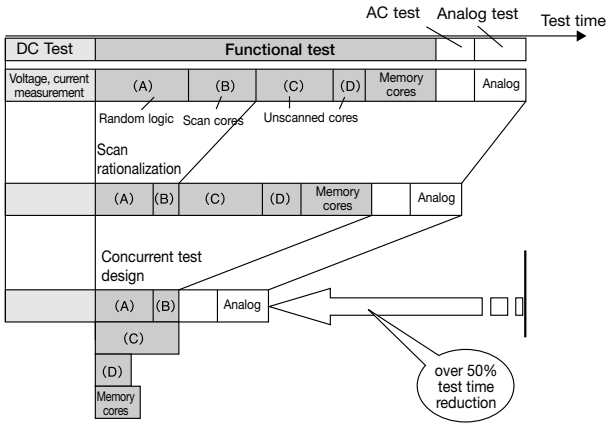

Fig. 6 Test time reduction results

References

1) Hatayama: “Basic information for designers on design-for-testability”, Design Wave Magazine, pp. 47 – 54, March 2001. [in Japanese]
2) Sato: “Basic information for designers on BIST technology”, Design Wave Magazine, pp. 55 – 67, March 2001. [in Japanese]

Authors

Masanori Ushikubo: Silicon Solutions Company, LSI Div., Test Engineering Dept.