DFT Timing Design Methodology for At-Speed BIST

Yasuo Sato[1], Motoyuki Sato[1], Koki Tsutsumida[1], Masatoshi Kawashima[1], Kazumi Hatayama[2], and Kazuyuki Nomoto[3]

1) Device Development Center
   Hitachi, Ltd.
   Ome-shi, Tokyo, Japan, 198-8512
   Tel: +81-428-33-2017
   Fax: +81-428-33-2157
   e-mail: ysatoh@ddc.hitachi.co.jp
           motoyuki@ddc.hitachi.co.jp
           k-tsutsu@ddc.hitachi.co.jp
           masato-k@ddc.hitachi.co.jp

2) Central Research Laboratory
   Hitachi, Ltd.
   Kokubunji-shi, Tokyo, Japan, 185-8601
   Tel: +81-42-327-7877
   Fax: +81-42-327-7736
   e-mail: k-hataya@crl.hitachi.co.jp

3) Semiconductor & Integrated Circuits
   Hitachi, Ltd.
   Kodaira-shi, Tokyo, Japan, 187-8588
   Tel: +81-42-320-7300
   Fax: +81-42-327-8638
   e-mail: nomoto-kazuyuki@sic.hitachi.co.jp

Abstract

Logic BIST is well known as an effective method for low cost testing. However, it is difficult to realize at-speed testing, as it requires a deliberate timing design in regard to logic design and layout of the chip. This paper presents a timing design methodology for at-speed BIST, using a multiple-clock domain scheme. Some experimental test results of large industrial designs using our custom tool "Singen", will also be shown.

1. INTRODUCTION

The increase of timing-related failure becomes a crucial issue in the deep sub-micron (DSM) technology. Fig.1 shows the distribution of defect [1][2], which shows that potential of failure increases as particle (defect) size decreases. Moreover, small defects, which had been benign in the conventional process, tend to cause a fatal timing failure in high speed LSI's. To detect them, at-speed testing has been investigated intensively [3]-[7].

Logic BIST is well known as an effective method for low cost testing because it enables us to test a high-speed design chip with a low speed ATE. Some papers in regard to at-speed BIST have been published. They show multiple-clock domain schemes to test DUT (device under test) at system cycle [6][7].

However, it is difficult to realize at-speed BIST, as it requires a deliberate timing design in regard to logic design and layout of the chip. Few papers have reported in regard to timing design of DFT circuits or clock design [8]. Ad hoc approaches have been adopted in industrial design. We need special care to satisfy restrictions such as set-up time or hold-time. Clock design is the most difficult one. Clock network should be designed to guarantee that any logic gate should operate properly in every testing mode (at-speed, medium speed, and slow speed).

In this paper, we will show our DFT timing design methodology for at-speed BIST using a multiple-clock domain scheme. We introduce the layout design of the DFT circuits and the clock network. They were realized with small modification of the original layout of user logic. We applied this methodology to our industrial design chips using our custom tool "Singen", and confirmed their short design term.
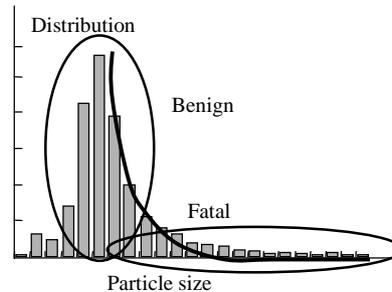

Fig.1 The distribution of defect

2. BIST DESIGN FLOW

2.1 The DFT structure

Fig.2 shows our DFT structure, which is based on STUMPS [9]. TPG is a test pattern generator, which is based on LFSR (Linear Feedback Shift Register). MISR (Multiple Input Signature Register) is used as a pattern compressor. The length of scan chain is reduced to 200-300 for realizing short testing time, and it is independent of input pin number. Three types of testing clock resources are available.

(1) PLLIN: the clock input of PLL (Phase Lock Loop)

   It is used for at-speed testing.

(2) TCK: the clock input for boundary scan test

   It is used for slow-speed testing (DC-BIST), and is also used for slow scan shifting.

(3) C1,C2: the clock inputs for debugging

   They are used for fast BIST (AC-BIST), which may not be at-speed. However, it is faster than TCK. Test timing is controllable according to the difference between C1 and C2. It is known that the skew of two pins on ATE

can be adjusted to fair level ordinary. As PLL doesn't operate well at slower speed than the specification, this function is essential for debugging.

CIF (external clock interface) generates the test clocks from (C1,C2) or TCK. TGN (test clock generator) generates at-speed test clocks from PLL. TCU (test control unit) controls them.
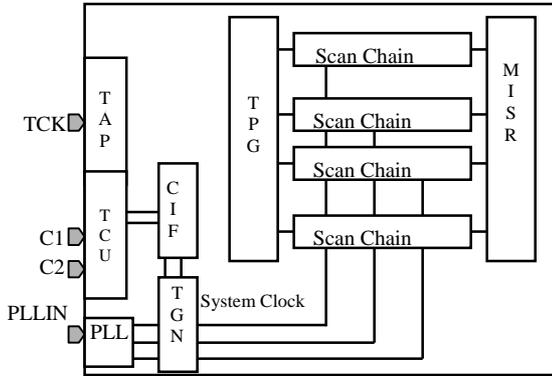

Fig.2 The DFT structure

2.2 The BIST design flow

Fig.3 shows our BIST design flow. It is mainly applied to ASIC's in 0.18um technology. Therefore, short design-term is strongly required. Their frequencies are ordinary from 30MHz to 500MHz. The DFT is constructed of logic BIST, memory BIST and the boundary-scan [10]-[15].

Firstly, DFT rule checker will find design rule violations. For example, a gate-loop, an asynchronous "set" or "reset" signal, a gated-clock, and a negative-edged flip-flop should be modified to satisfy the rules.

Secondly, the test point insertion (TPI) is applied. TPI will improve the fault coverage with small number of additional gates. It was developed to minimize the gate overhead and delay overhead [12].

Then, DFT synthesis inserts several control blocks (TAP, TCU, CIF, TGN), scan chains, TPG and MISR into the original logic. At the same time, it will output the timing script file, which will be used for timing driven layout (TDL) and static timing analysis (STA). To realize this function, all of the test signals were categorized into the following three levels:

Level-1: The signals that should operate at the speed of system cycle (at-speed):

TGN, CIF, test clocks, and scan enable signal

Level-2: The signals that should operate at the speed of scan shifting cycle:

JTAG signals, scan chains, TPG, and MISR

Level-3: The signals that need only DC-level speed:
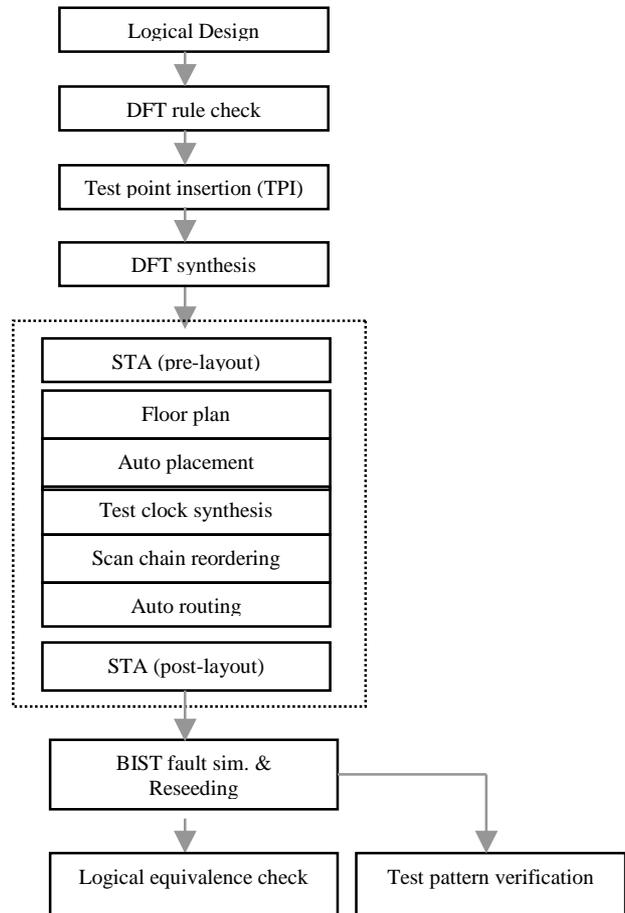
Mode control signals

Fig.3 BIST design flow

The signals in Level-1 should be treated carefully. For instance, some of them need manual layout and others use timing driven layout (TDL) and clock-tree synthesis (CTS). The signals in Level-2 will be realized using TDL or manual floorplan of blocks. The signals in level-3 don't need any special care for their timing. However, the number of fan-out should be optimized.

After TDL is completed, BIST pattern generation and fault simulation will be performed. It is consisted of the random pattern-based BIST and reseeding-based BIST (NPG: neighborhood pattern generation [11][15]). The detail of at-speed BIST scheme will be described in the following section.

3. AT-SPEED BIST SCHEME

3.1 Multiple-clock domain testing scheme

Fig.4 and Fig.5 show our at-speed BIST scheme for multiple-clock domain. Fig.4 shows a case when TI clock launches a pulse, and the same clock (TI) captures it. Fig.5 shows a case when TI launches a pulse, and TJ captures it. Thus, each clock pair is tested respectively, whereas other clocks are frozen in a capture window. TI and TJ operate at rather slow speed in scan-in mode or scan-out mode. The

timing between the launch and the capture (Tij) is the same as system clock cycle (i.e. at-speed).

This scheme has the following features:

(1) The test control is simple.

(2) The two delay testing methods (the skewed-load test [16][17][18], and the broad-side test [19]) are available.

(3) Pair of clocks, for instance, one of which is from a PLL and the other is from an external clock pin, doesn't synchronize with each other. So they can be tested at slower speed (AC-BIST or DC-BIST).

(4) The power and noise during scan shifting are reduced.

(5) The debugging and diagnosis of testing is viable.

The only drawback of this method is an increase of testing time. However, a design chip usually consists of a few main clocks and many sub-clocks. If so, the testing time mostly owes to the main clocks, and others contribute little. Our experiment [14] also confirmed this phenomenon.

Fig.6 shows our multiple-clock domain scheme. There are many clock domains, which have different delay length (Di). The depth of each cone shows Di. Each clock is supplied from a PLL or an external clock pin. In at-speed testing, the clock in a capture window is supplied from a PLL (bold line) as the system clock operation. Each path from domain-I to domain-J should be designed to operate at the speed of Tij.

However, when the clock is supplied from TCK or C1-C2 in DC-BIST or AC-BIST, the clocks go through other paths (Fig.6). So the clock skew from domain-I to domain-J can be as large as ΔIJ (=Di−Dj). Therefore, test timing ($T^{DC\text{-}BIST}$ or $T^{AC\text{-}BIST}$) should be greater than Tij + ΔIJ. From Fig.4, 5 and 6, we know that SEN (scan enable) should be enabled between launch and capture. We generate SEN from a clock resource and treat it like another clock. According the discussion above, we derive the following restrictions;

$$TI \text{ (launch)} < SEN \text{ (low)} < TJ \text{ (capture)}$$

$$= TJ \text{ (launch)} + Tij \qquad (1)$$

$$T^{DC\text{-}BIST}{}_{IJ} \geq Tij + \Delta IJ \qquad (2)$$

$$T^{AC\text{-}BIST}{}_{IJ} \geq Tij + \Delta IJ \qquad (3)$$

$$T^{at\_speed} = Tij \qquad (4)$$

We should remark that relation (1) is needed for the skewed-load test. If we only use the broad-side test for delay testing, relation (1) can be neglected. In our example in section 4, we have used both methods combined to get high delay fault coverage.

From (1)-(4), we conclude that reducing ΔIJ is crucial in timing design. It is also effective to reduce hold-time violations during scan shifting between different clocks as shown in Fig.7. The delay from FF2 to FF3 (Dij) should be larger than the hold-time of FF3 (T^{hold}).
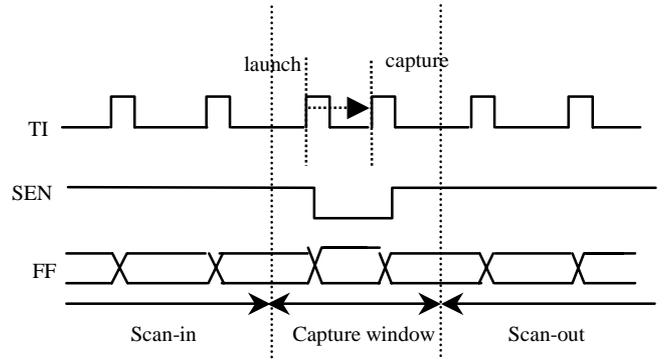
$$Dij \geq T^{hold} + \Delta IJ \qquad (5)$$
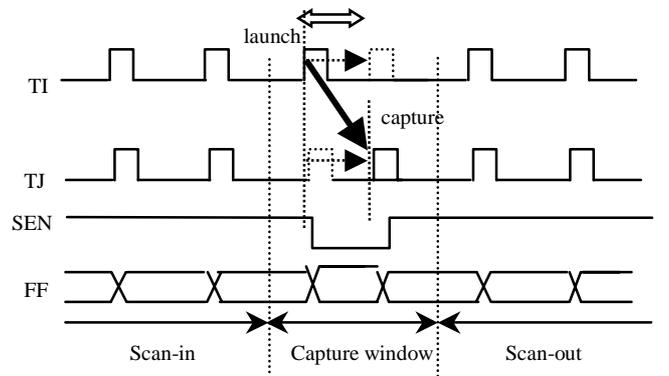

Fig.4 Test timing (TI-TI)
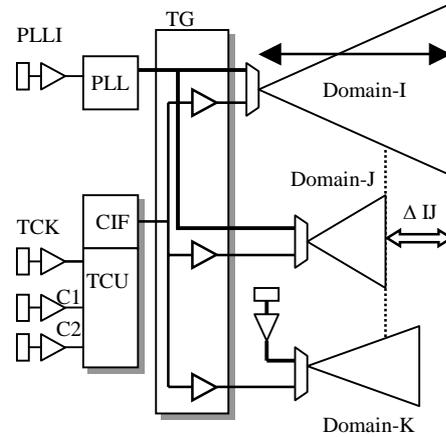

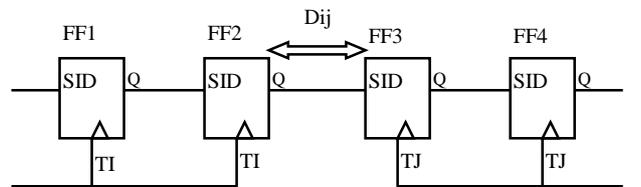Fig.5 Test timing (TI-TJ)

Fig.6 Multiple-clock domain model

Fig.7 scan-chain timing

3.2 Reducing clock skew

In the previous section, we have shown that all clocks should be treated as if they were in a domain during AC-BIST or DC-BIST mode. To ensure the signal propagation between different clock domains during AC-BIST or DC-BIST, clock skew ΔIJ (I=1 to N, J=1 to N) should be minimized at reasonable level.

Fig.8 shows our concept of reducing ΔIJ. We insert delay gates (ΔDj) between test pins and the selectors that switch PLL clock and test clock. This process is performed after generating system clock domains. Therefore, it doesn't effect system clock delay or skew at all. The layout procedure using a commercial CTS (clock tree synthesis) tool will be as follows;

1st step: create system clock domains.

    (specify clock delay and skew)

2nd step: create test clock domain (all clocks are treated

    as a domain) preserving each system clock domain-I.

    (specify the longest delay of Di + α)

3rd step: create scan enable trees corresponding to each

    system clock domain-I.

    (specify the delay as Di + α, the skew as SKi)

Fig.9 shows a layout of Fig.8. The clocks of domain-I and domain-J are supplied from PLL. The clock of domain-K is supplied from a clock pin (T-K). The revised clock skew ($\Delta IJnew$) will be as follows;

$$\Delta IJnew = (Di + \Delta Di) - (Dj + \Delta Dj)$$
$$= (Di - Dj) - (\Delta Dj - \Delta Di)$$
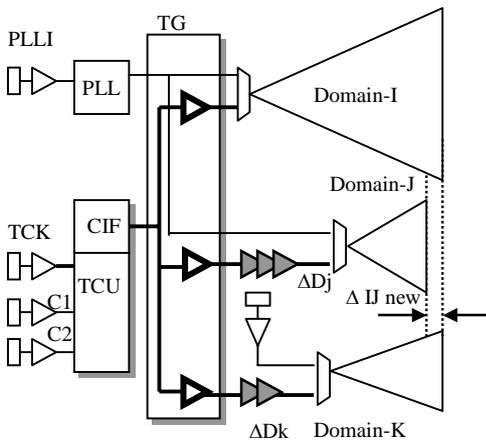$$= \Delta IJ - (\Delta Dj - \Delta Di) \qquad (6)$$

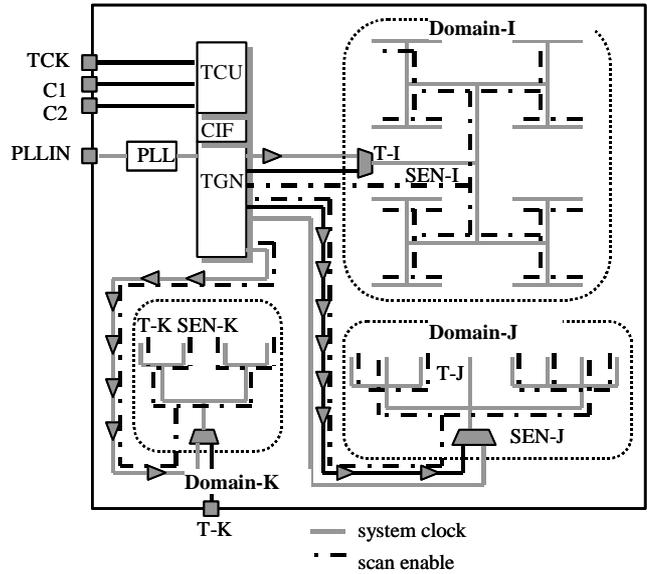Fig.8 Multi-clock domain (optimized)

Fig.9 Multi-clock domain layout

After the layout design is completed, STA (Static Timing Analysis) in regard to the timing restrictions described in section 3.1 is performed. The STA script is made by DFT synthesis automatically. It will be as follows;

(a) Script for scan enable to check restriction (1)

  - Define a clock that starts from port C2 to TI (I=1,N).

  - Check setup and hold time at each flip-flop considering scan enable to be a data path triggered by the clock (Fig.10).
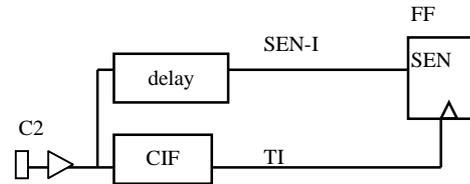

Fig.10 timing check of scan enable

(b) Script for capture window to check restriction (2)& (3)

  - Set scan enable (SEN-J) as '0' (capture mode).

  - Define a clock that starts from port C2 to TI (I=1,N).

  - Check setup and hold time between TI and TJ.

  Other clock pairs are defined as false paths, which will not be checked. (See section 3.1 (3) and Fig.5)

(c) Script for scan shifting to check restriction (5)

  - Set scan enable as '1' (scan in or scan out mode).

  - Define a clock that starts from port TCK.

  - Check setup and hold time at each flip-flop.

4. EXPERIMENTAL RESULTS

Fig.11 shows the distribution of delay and clock skew for 8 clock domains of 700k gate ASIC. We applied the method introduced in section 3.2 using a commercial CTS tool. Preserving the original clock domain cones, the delay from test clock pin (TCK or C1-C2) was leveled around the length of 10.58 ns with 2.189 ns skew (all clocks were treated as one domain). This level was enough for our 54Mhz design.

Fig.12 shows the analysis of turn around time. For each design step, we need some manual operation such as optimizing the parameters for the first time. So we needed 17 hours (excluding layout time). In layout design, CTS needed 5 hours. On the final design stage, manual work should be almost negligible, and TPI will need less time.

Fig.13 shows the results of evaluation data. Their stuck-at fault efficiency was 99.67%, 99.97% and 99.98%, respectively. Their transition fault efficiency was 96.94%, 94.67% and 98.35%, respectively. They were acquired using the random-based BIST and reseeding-based BIST (NPG: neighborhood pattern generation). The BIST pattern count of ASIC3 was reduced less than others with optimization.

clock	delay (ns)	skew (ns)
T0	7.79	0.318
T1	1.64	0.003
T2	2.17	0.035
T3	2.10	0.042
T4	3.44	0.04
T5	3.15	0.146
T6	2.85	0.161
T7	1.23	0.000

Fig.11 clock skew distribution

Item	Manual (hr)	CPU (hr)
DFT rule check	0.3	0.2
TPI	1.0	8.0
DFT synthesis	0.5	0.2
Formality	0.5	0.2
BIST sim.	1.0	3.3
Verification	1.0	1.0
Sum.	4.3	12.9

Fig.12 Turn around time

ASIC3 was tested at 400MHz (launch-capture speed). It consists of 3 test clocks:

- CK1: 400MHz main clock

- CK2: 100MHz sub-clock

- TT: 50MHz dedicated test clock that is used for the boundary-scan test, and memory BIST

The clock layout was performed manually using the tree buffering technique and its skew was reduced to less than 100 ps. The scan enable signals were designed in the same way. The scan shift worked at the speed of 20ns, and the length of scan chain was within 300 flip-flops. The timing violation of hold-time, most of which depend on MUXD-scan structure, occurred frequently. However, timing tuning gates were inserted automatically and their layout was performed incrementally in several hours.

Item	ASIC1	ASIC2	ASIC3
Gate count	1495K	1050K	1272K
Frequency(MHz)	51	136	400
Scan chain length	300	214	300
BIST pattern count (DC)	510K	510K	133K
BIST pattern count (AC)	510K	510K	314K
Fault Efficiency (BIST-DC)	98.30%	99.74%	99.88%
Fault Efficiency (reseed-DC)	99.67%	99.97%	99.98%
Fault Efficiency (AC)	96.94%	94.69%	98.35%
BIST sim time (hr)	11.0	4.6	8.5

Fig.13 Implementation results

Item	ASIC3
TPG	2.2%
MISR	0.8%
Scan chain	44.7%
Boundary Scan	2.8%
Scan enable, TT	29.2%
Control	19.9%
TPI	0.4%
else	1.6%
Sum.	100%

Fig.14 Wiring overhead evaluation

The gate overhead of ASIC3 was 8.5%. This includes the overhead of scan-chain, TPG, MISR, boundary scan, TAP,

TCU, TT-related gate, and fan-out gates for test control signals. The wiring overhead of ASIC3 is shown in Fig.14. As today's LSI has many wiring-layers and their capacity defines its chip size, we are more interested in the wiring overhead than the gate overhead. DFT synthesis tool made the signal names to correspond to the items in Fig.14. Therefore, it is easy to extract them from layout.

5.   CONCLUSIONS

In this paper, we have shown a timing design methodology for at-speed BIST, and some experimental test results of industrial designs using our custom DFT tool "Singen".

(1) An at-speed BIST scheme was presented. It tests DUT for each clock pair. Timing restrictions for this scheme were extracted. Reducing the clock skew between different clock domains was crucial.

(2) We showed a systematic layout approach to reduce the clock skew described in (1). Actual experimental data was shown. Short design time was confirmed.

(3) Implementation results for three ASICs were introduced. 400Mhz at-speed test was achieved, and high fault efficiency of 99.67 - 99.98% was acquired.

REFERENCES

[1] C.H.Stapper, "Modeling of defects in integrated circuit photolithographic patterns", *IBM J. Res. Develop,* Vol.28 No.4 July 1984

[2] International Technology Roadmap for Semiconductors, 1999 Edition

[3]   T.G.Foote, D.E.Hoffman, W.V.Huott, T.J.Koprowski, B.J.Robbins, M.P.Kusko, "Testing the 400MHz IBM generation-4 CMOS chip", *Int. Test Conf.*, 1997, pp. 106-114

[4] P.S.Gillis, T.S.Guzowski, B.L.Keller, R.H.Kerr, "Test methodologies and design automation for IBM ASICs", *IBM J. Res. Develop,* Vol.40 No.4 July 1996

[5] J.Braden, Q.Lin, B.Smith, "Use of BIST in SUN FIRE^{TM} servers", *Int. Test Conf.*, 2001, pp. 1017-1022

[6] G.Hetherington, T.Fryars, N.Tamarapalli, M.Kassab, A.Hassan, J.Rajski, "Logic BIST for large industrial designs: real issues and case studies", *Int. Test Conf.*, 1999, pp. 358-367

[7] B.N.-Dostie, "Design for at-speed test, diagnosis and measurement", *Kluwer Academic Publishers*, 1999

[8] X.Gu, et al., "An Effort-Minimized Logic BIST Implementation Method", *Int. Test Conf.*, 2001, pp. 1002-1010

[9] P.Bardell, W.McAnney, J.Savir, "Built-in test for VLSI: pseudorandom techniques", *John Wiley and Sons,* 1987

[10] Y.Sato, T.Ikeya, M.nakao, T.Nagumo, "A BIST approach for very deep sub-micron (VDSM) defects", *Int. Test Conf.*, 2000, pp. 283-291

[11] M.Nakao, Y.Kiyoshige, K.Hatayama, S.Fukumoto, K.Iwasaki, "Deterministic built-in test with neighborhood pattern generator", *IEICE TRANS. INF.&SYST.,* Vol.E85-D, No.5, 2002, pp.874-883

[12] N.Nakao, S.Kobayashi, K.Hatayama, K.Iijima, S.Terada, "Low overhead test point insertion for scan-based BIST", *Int. Test Conf.*, 1999, pp. 348-357

[13] M.Nakao, Y.Kiyoshige, K.Hatayama, Y.Sato, T.Nagumo, "Test generation for multiple-threshold gate-delay fault model", *Asian Test Symposium*, 2001, pp. 244-249

[14] K.Hatayama, M.Nakao, Y.sato, "At-speed Built-in Test for Logic Circuits with Multiple Clocks", *Asian Test Symposium*, 2002, in press.

[15]   K.Hatayama, M.Nakao, Y.Kiyoshige, K.Natsume, Y.Sato, T.Nagumo, "Application of High-Quality Built-in Test to Industrial Designs", *Int. Test Conf.*, 2002, pp. 1003-1012

[16] K.-T Cheng, S.Devadas, K.Keutzer, "Delay-fault test generation and synthesis for testability under a standard scan design methodology", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*, 12(8), 1993, pp. 1217-1231

[17] J.Savir, "Skewed-load transition test: part I, calculus", *Int. Test Conf.*, 1992, pp. 705-713

[18] J.Savir, "Skewed-load transition test: part II, coverage", *Int. Test Conf.*, 1992, pp. 714-722

[19] J.Savir, "On broad-side delay testing", *VLSI Test Symposium*, 1994, pp. 284-290