

# Gate Strength Aware DC Coverage Improvement

Boon Chong Ang and Eng Lian Goh  
 eASIC Corp  
 Email: bang@easic.com, elgoh@easic.com

**Abstract**—A study[1] shows that the data came from operational tests of systems between 1985 to 1990 and 1996 to 2000, the percentage of systems meeting reliability requirements decreased from 41 percent to 20 percent. As system complexity increases, testability is alarming in almost every applications development. There is a need to put more efforts to address the issues of testability at the device, board and system level in order to deliver more consistently reliable and cost effective products to the market. In the current industry, the highest acceptable defects parts per millions, DPM is 500 DPM or lower. To achieve 200 DPM in matured process that typically yield 99%, the test coverage requirement is 98%. This paper will address the DC coverage improvement through the proposed gate strength aware modeling.

**Index Terms**—ATPG modeling, ASIC, DC coverage

## I. INTRODUCTION

Study shows that the percentage of systems meeting reliability requirements decreased from 41 percent to 20 percent for operational tests of systems from year 1985 to 1990 shown in Fig. 1 and year 1996 to 2000 shown in Fig. 2.

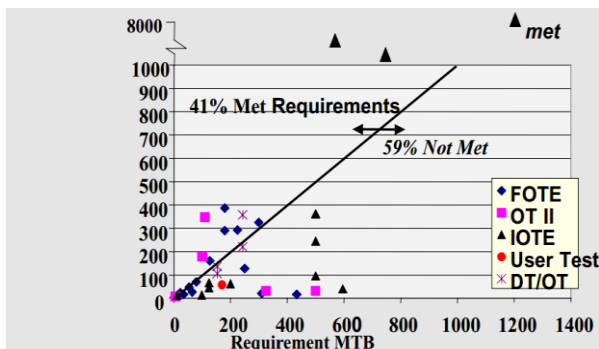


Figure 1. Demonstrated reliability versus requirements for operational tests, 1985-1990[ 1]

Fig. 1 and Fig. 2 show the reliability versus the mean time between (MTB) all sort of failures in follow-on test and evaluation, (FOTE), operating test II, (OT II), initial operational test and evaluation, (IOTE), developmental test, (DT), operating test, (OT), limited user test, (LUT), follow on test, (FOT) and initial operational test, (IOT). From Fig. 1, it is concluded that a large percentage of systems failed to meet the required operational reliability and the trend is

worsened in year 1996 to 2000 demonstrated operational reliability versus requirements. Hence there is a need to address system testability in every application developments at the device, board and system levels in order to deliver consistently reliable and cost effective products to the market place.

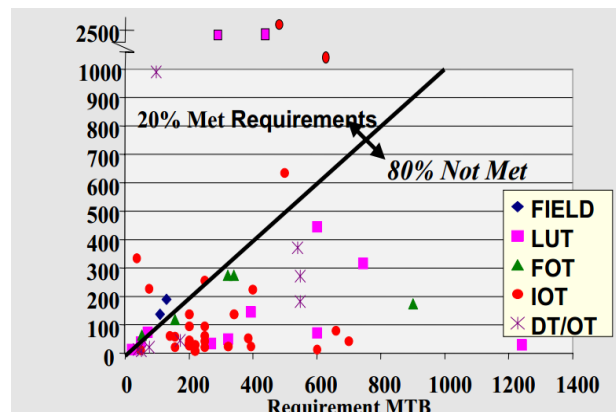


Figure 2. Demonstrated reliability versus requirements for operational Tests, 1996-2000[1]

The failures in integrated circuits, IC can be group by physical failure, electrical failure, in-process failure and reliability failure shown in Fig. 3.

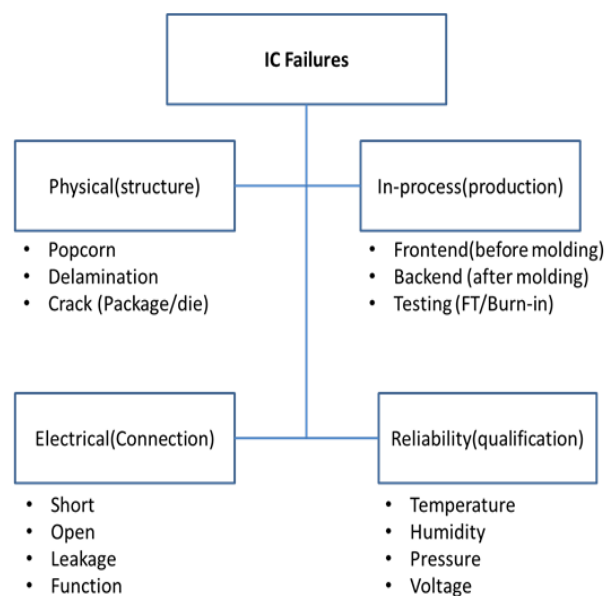


Figure 3. IC failure classification[2]-[15]

From Fig. 3, the electrical failure can be properly verified through design for testability,(DFT) technique. Design for testability(DFT) technique is divided into ad-hoc methods and structured methods shown in Fig. 4.

From Fig. 4, for the ad-hoc DFT technique, the required DFT guidelines are:

- Partition large circuits into smaller subcircuits to reduce test generation cost
- Insert test points to enhance controllability and observability
- Design circuits to be initializable
- Provide logic to break global feedback paths
- Avoid the use of redundant logic
- Keep analog and digital circuits physically apart
- Avoid the use of asynchronous logic

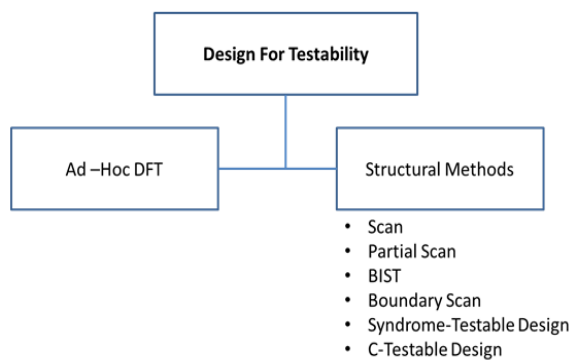


Figure 4. Design for testability techniques

For ATPG deployment, there are 2 basic restrictions that users must aware[16]:

- ATPG tool cannot handle bidirectional devices such as tranif in Verilog primitive gate
- Faults are considered at the ports of the lowest module in the netlist

Besides the 2 basic ATPG restrictions, a design has to comply with scan design rules to utilize the scan structure and to achieve the target fault coverage goal. The details of basic scan design rules are shown in Table I.

TABLE I. SCAN DESIGN RULES

Design Style	Scan Design Rule	Solutions
Tri-state bus	avoid during shift	Fix bus contention during shift
Bidirectional I/O ports	avoid during shift	Force to input or output mode during shift
Gated clock	avoid during shift	Enable clocks during shift
Derived clock	avoid	Bypass clocks
Combinational feedback loop	avoid	Break the loops
Asynchronous set-reset signal	avoid	Use external pins
Clocks used as data	avoid	Block clocks to the data portion
Floating buses	avoid	Add bus keepers
Floating inputs	not recommended	Tie to power or ground
Cross-coupled NAND/NOR gate	not recommended	Use standard cell for sequential elements
Non-scan storage elements	not recommended for full scan design	Initialize to known states or bypass or make transparent

In this paper, the discussion of stuck-at-fault test will

compare the default stuck-at-fault test flow coverage against

- 1) Cell aware stuck-at-fault test coverage improvement[17]-[20].
- 2) The proposed's gate strength aware stuck-at-fault test coverage improvement.

Hopefully, the readers can benefit from the sharing.

## II. DEFAULT DC COVERAGE FLOW

For the default automatic test pattern generator, ATPG flow for stuck-at-fault test, users are allowed to increase the test abort level as well as to complement the basic scan with N cycle of sequential capture in order to improve stuck-at-fault coverage at the expense of ATPG runtime. The typical automatic test pattern generator, ATPG flow is shown in Fig. 5.

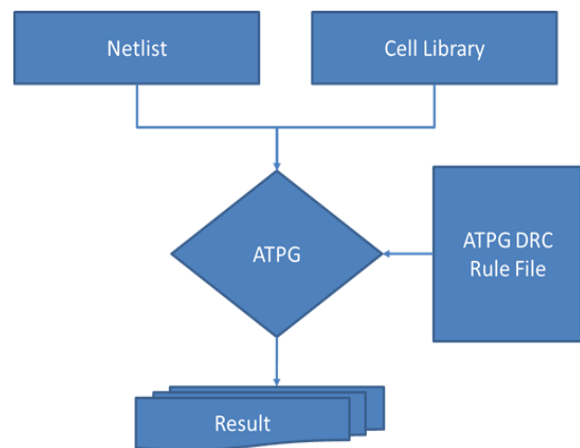


Figure 5. ATPG flow

As the ATPG library modeling does not reflect the impact of gate's drive strength, the default approach to improve the test coverage is to collapse the fault count hence reduce the total fault count and the untestable fault nodes. The difference between the collapsed fault and uncollapsed fault is shown in Table II.

TABLE II. COLLAPSED VERSUS UNCOLLAPSED FAULT REORT

cell function	collapsed_faults	uncollapsed_fault	Fault Reduction(%)
2 input AND	4	6	33
2 input OR	4	6	33
Buffer	2	4	50
Inverter	2	4	50
2to1 Mux	8	8	0
2 input NOR	4	6	33
2 input NAND	4	6	33
2 input XNOR	6	6	0
2 input XOR	6	6	0

From Table II, the collapsed's fault is 50% of uncollapsed fault for buffer and inverter logic gate, and 33% of uncollapsed fault for AND gate, OR gate, NAND gate and NOR gate. The difference between AND gate's collapsed and uncollapsed fault can be explain as follow:

- Stuck at zero (SA0) at any inputs is equivalent to stuck at zero (SA0) at the output.

- Stuck at one (SA1) at output is dominated by stuck at one (SA1) at both inputs.

Hence, for AND gate collapsed's fault, the stuck at zero, SA0 will result in 3 faults while the stuck at one, SA1 is dominance fault which can observe directly from the gate's output. The resulting total collapsed's faults for 2 input AND gate is 4 faults.

As buffer, inveter, AND, OR, NAND and NOR gate made up for majority of IP cell counts, the collapsed's fault reporting approach greatly reduces the total faults' counts and improves the effective fault coverage. The result of full chip collapsed's coverage report versus uncollapsed's coverage report is shown in Table III.

TABLE III. DEFAULT ATPG FAULT COVERAGE REPORT

Fault Class	collapsed_fault	uncollapsed_fault	Fault reduction(%)
Detected	1855789	2933063	37
Possibly detected	124	223	44
Undetectable	38231	63844	40
ATPG untestable	10929	19122	43
Not detected	1667	3188	48
Total Faults	1906740	3019440	37
Test Coverage	99.32%	99.24%	

From Table III, the collapsed-fault-coverage report improves the fault coverage from 99.24% to 99.32% by reducing the total faults' count. To achieve a low defect per million, 0.1% improvement in fault coverage is a significant improvement.

### III. CELL AWARE ATPG MODELING

For cell aware ATPG modeling introduced by Mentor[17]-[20], the primary objective is to improve the bridging coverage and it involves library modeling for Mentor's ATPG tool. It can be illustrated with 3- to-1 input MUX logic based pattern set required. To test 3-to-1 input MUX, the logic based pattern required is shown in Table IV.

TABLE IV. 3-TO-1 MUX LOGIC BASED PATTERN REQUIRED

S0	S1	D0	D1	D2	Z
0	0	0	?	?	0
0	0	1	?	?	1
?	1	?	?	0	0
?	1	?	?	1	1
1	0	?	0	?	0
1	0	?	1	?	1

TABLE V. 3-TO-1MUX LOGIC BASED PATTERN REQUIRED WITH BRIDGING FAULT

S0	S1	D0	D1	D2	Z
0	0	0	?	1	0
0	0	1	?	?	1
1	1	?	1	0	0
?	1	?	?	1	1
1	0	?	0	1	0
1	0	?	1	?	1
0	1	1	?	0	0

To detect the bridging fault from S1 to D2 node, the test pattern required is shown in Table V.

By comparing Table IV and Table V, to cater for the additional bridging fault, additional 1 pattern is required. Bridging fault can be detected through the capacitive coupling report for bridging test shown in Fig. 6 while the library modeling remain unchanged.

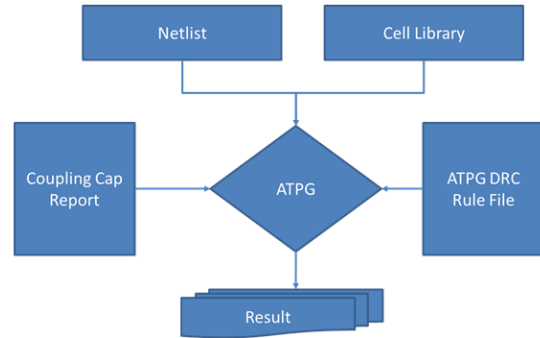


Figure 6. Bridging test flow

For the default 3-to-1 Mux modeling, the typical fault count reported from automatic test pattern generator (ATPG) tool is 12 faults, as 3-to-1 multiplexer has 5 input ports and 1 output ports. By leveraging the cell aware technology to model the 3-to-1 multiplexer, MUX, the library modeling can be modeled as Fig. 7 to reflect the physical construction.

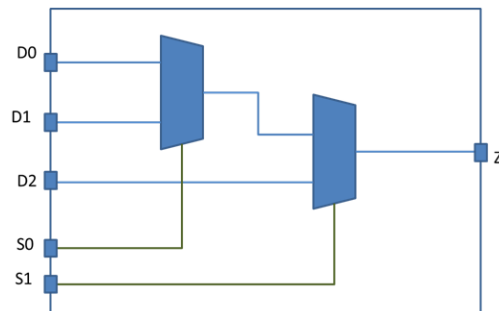


Figure 7. 3-to-1 mux remodel

From the ATPG tool, the fault count and test pattern required is shown in Table VI.

TABLE VI. CELL AWARE FAULT REPORT

cell	Original			Cell Aware Modeling		
	collapsed_faults	uncollapsed_fault	Pattern	collapsed_faults	uncollapsed_fault	Pattern
3to1MUX	12	12	6	28	28	6

```

module MUX3_abc (I0, I1, I2, S0, S1, Z);
input I0, I1, I2, S0, S1;
output Z;
mux_abc U1 (I0_out, I0, I1, S0);
mux_abc U2 (Z, I0_out, I2, S1);
Endmodule
  
```

```

module mux_abc (q, d0, d1, s);
output q;
input s, d0, d1;
assign q=s?d1:d0;
endmodule
  
```

Figure 8. 3-to-1 mux cell aware modeling

From Table VI, it is observed that the pattern count for cell aware modeling remain unchanged, however the detectable fault counts increase from 12 to 28 faults. Hence cell aware modeling may result in better fault coverage report based on the cell implementation. The cell aware library modeling for 3-to-1 Mux is shown in Fig. 8.

The ATPG flow in Fig. 5 is repeated by replacing the default library modeling with cell aware library modeling and the ATPG result is shown in Table VII.

TABLE VII. CELL AWARE ATPG REPORT

Fault Class	Default		Cell Aware	
	collapsed_fault	uncollapsed_fault	collapsed_fault	uncollapsed_fault
Detected	1855789	2933063	1852773	4178555
Possibly detected	124	223	127	224
Undetectable	38231	63844	66504	104399
ATPG untestable	10929	19122	13935	24961
Not detected	1667	3188	1463	4083
Total Faults	1906740	3019440	1934802	4312222
Test Coverage	99.32%	99.24%	99.17%	99.31%
Pattern Count	4269	4269	4382	4382

From Table VII, cell aware library modeling produces better uncollapsed fault coverage report than the default library modeling approach, due to increase of detectable fault counts. It is observed that the collapsed fault coverage report for cell aware library modeling does not guarantee to produce better test coverage than uncollapsed fault coverage report, unlike the default DC coverage flow’s observation. The total collapsed faults reported in cell aware library modeling approach, is slightly higher than the default library modeling’s total collapsed faults. This observation is expected, as not all cell functions will reduce the fault counts in collapsed format. The pattern counts’ increment with cell aware modeling is negligible.

IV. GATE STRENGTH AWARE ATPG MODELING

For a standard cell physical library, it is constructed with fixed physical’s height such as 9 tracks cell or 12 tracks cells, as standard offerings from the foundry. As a result of fixed physical height, the width of the transistor is fixed. To cater the need of drive strength’s variation through uniform width and height of transistor, the transistor outputs are connected in parallel to improve the drive strength. Hence a buffer with higher drive strength can be viewed as super buffer shown in Fig. 9.

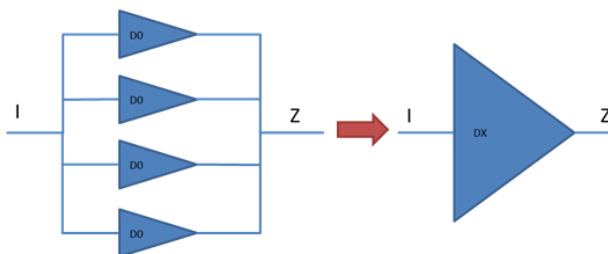


Figure 9. Gate strength’s physical implementation in ASIC standard cells

The library modeling of gate strength can be equated with either of the following approaches:

- The ratio of output driver’ transistor counts to the smallest drive strength of equivalent cell.

- The total transistor’s width and length of output driver to the transistor’s total width and length of the equivalent cell with smallest drive strength.

For the ATPG modeling to reflect the gate drive strength, the cell has to connect in series shown in Fig. 10, to avoid output contention violation.

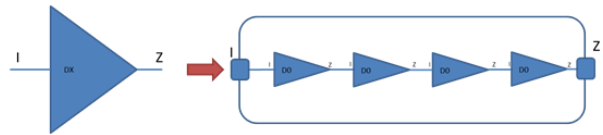


Figure 10. Gate strength aware ATPG modeling

From Fig. 10, the collapsed-fault report for buffer and inverter with gate strength aware modeling will remain unchange while the uncollapsed faults for buffer and inverter will be greater than 4, pending on the gate strength of logic cell.

The ATPG flow in Fig. 5 is repeated by replacing the default library modeling with cell aware library modeling and gate strength aware library. The ATPG result is shown in Table VIII

TABLE VIII.GATE STRENGTH AWARE ATPG RESULT

Fault Class	Default		Cell Aware		Cell Aware+Gate Strength Aware	
	collapsed	uncollapsed	collapsed	uncollapsed_fault	collapsed_fault	uncollapsed_fault
Detected	1855789	2933063	1852773	4178555	1853362	5496970
Possibly detected	124	223	127	224	154	745
Undetectable	38231	63844	66504	104399	66497	115107
ATPG untestable	10929	19122	13935	24961	13295	29378
Not detected	1667	3188	1463	4083	1657	4738
Total Faults	1906740	3019440	1934802	4312222	1934965	5646938
Test Coverage	99.32%	99.24%	99.17%	99.31%	99.20%	99.38%

From Table VIII, the uncollapsed –fault report with cell aware and gate strength aware library modeling provides the best coverage report.

V. CONCLUSION

The summary of Gate Strength Aware ATPG modeling is shown in Table IX.

TABLE IX. ATPG MODELING SUMMARY

Library Modeling	Summary
Original	Improved coverage by reducing fault points based on functional truth table.
Cell Aware	Improved the coverage by reflecting the actual fault points
Cell Aware +Gate Strength Aware	Further Enhance the coverage coverage by actual fault points

From Table IX, it concludes that cell aware and gate strength aware modeling will improve the uncollapsed fault coverage report, at the cost of additional effort in library modeling. For gate strength aware ATPG, it can be implement through library modification or enhanced made by the ATPG tool vendor, to allow users to provide the uncollapsed-fault’s weight for standard cells’ ports with different drive strength.

ACKNOWLEDGEMENTS

Thanks to eASIC management, Lai Kok Keong, Massimo Verita, Lai Sheng Tat, Kenneth Chen and Michael Basanty for the support given.

REFERENCES

- [1] "Reliability performance today," *ATEC/PEO*, July 27, 2001.
- [2] B. G. West, "Accuracy requirements in at-speed functional test," in *Proc. Int. Test Conf.*, 1999, pp. 780-787.
- [3] J. T. Y. Chang, C. W. Tseng, Y. C. Chu, S. Wattal, M. Purtell, and E. J. McCluskey, "Experimental result for IDDQ and VLV testing," in *Proc VLSI Test Symp.*, 1998, pp. 118-123.
- [4] Y. Zorian, "Testing the monster chip," *IEEE Spectrum*, pp. 54-60, 1999.
- [5] B. I. Dervisoglu and G. E. Stong, "Design for testability: Using scanpath techniques for path-delay test and measurement," in *Proc Int. Test Conf.*, 1991, pp. 365-374.
- [6] S. Kim, M. Soma, and D. Risbud, "An effective defect-oriented BIST architecture for high-speed phase-locked loops," in *Proc. VLSI Test Symp.*, 2000, pp. 231-236.
- [7] S. C. Ma, P. Franco, and E. J. McCluskey, "An experimental chip to evaluate test techniques experimental results," in *Proc. Int. Test Conf.*, 1995, pp. 663-672.
- [8] (2008). The Evolution and Future of DfX In Modern ICs. Future Fab. [Online]. Available: [www.future-fab.com](http://www.future-fab.com)
- [9] S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, Addition-McGraw Hill, 2003
- [10] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC scan: The war against unmodeled defects," in *Proc. Int. Test Conf.*, 1996, pp. 250-258.
- [11] P. Nigh, W. Needham, K. Butler, P. Maxwell, R. Aitken, and W. Maly, "So what is an optimal test mix? A discussion of the sematech methods experiment," in *Proc. Int. Test Conf.*, 1997, pp. 1037-1038.
- [12] S. Eichenberger, J. Geuzebroek, C. Hora, B. Kruseman, and A. Majhi, "Towards a world without test escapes," in *Proc. of IEEE Int'l Test Conf.*, 2008, paper 20.1
- [13] K. Y. Cho, S. Mitra, and E. J. McCluskey, "Gate exhaustive testing," in *Proc. of IEEE Int'l Test Conf.*, 2005, paper 31.3
- [14] J. Geuzebroek, E. J. Marinissen, A. Majhi, A. Glowatz, and F. Hapke, "Embedded multi-detect ATPG and its effect on the detection of unmodeled defects," in *Proc. IEEE Int'l Test Conf.*, 2007, paper 30.3
- [15] S. Spinner, I. Polian, P. Engelke, B. Becker, M. Keim, and W. T. Cheng, "Automatic test pattern generation for interconnect open defects," in *Proc. VLSI Test Symp.*, 2008, pp. 181-186.
- [16] Accelerating Innovation. [Online]. [https://solvnnet.synopsys.com/dow\\_retrieve/H-2013.03/dce/dce\\_olh/Default\\_CSH.htm?param1=dceug/title\\_page.htm?otSearchResultSrc=advSearch&otSearchResultNumber=5&otPageNum=1](https://solvnnet.synopsys.com/dow_retrieve/H-2013.03/dce/dce_olh/Default_CSH.htm?param1=dceug/title_page.htm?otSearchResultSrc=advSearch&otSearchResultNumber=5&otPageNum=1)
- [17] F. Hapke, R. Krenz-Baath, A. Glowatz, J. Schloeffel, H. Hashempour, S. Eichenberger, C. Hora, and D. Adolfsson, "Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs," in *Proc ITC 2009*, 2009, pp. 1-10.
- [18] L. Y. Ko, S. Y. Huang, J. L. Chiou, and H. C. Cheng, "Modeling and testing of intra-cell bridging defects using butterfly structure," *VLSI Design, Automation and Test*, 2006.
- [19] H. Tang, G. Chen, C. Wang, J. Rajski, I. Pomeranz, and S. M. Reddy, "Defect aware test patterns," in *Proc. Design, Automation, and Test in Europe*, Munich, Germany, March 2005, pp. 450-455.
- [20] Cell-aware ATPG test methods improve test quality. [Online]. Available: <http://www.edn.com/design/manufacturing/4389613/Cell-aware-ATPG-test-methods-improve-test-quality>



**Ang Boon Chong** Obtained the B.Eng in electrical and electronic from University Putra Malaysia, Malaysia in 2002. He is currently pursuing part time MBA at Open University Malaysia. He started his career as an IC designer and work. He is currently a senior IEEE member and customer engineer at eASIC.



**Goh Eng Lian** received the B.E. degree in computer engineering from University of Technology Malaysia in 2002. He started his career as a product engineer and subsequently as a product development engineer in a network processor division. In 2008, he joined eASIC as test engineer. His specialties and interests include DFT, test program development and silicon debug using automated test equipment.