Scan Test of Latch-based Asynchronous Pipeline Circuits under 2-phase Handshaking Protocol

Kyohei Terayama
Hirosaki University
h14ms511@stu.hirosaki-u.ac.jp

Atsushi Kurokawa
Hirosaki University
kurokawa@eit.hirosaki-u.ac.jp

Masashi Imai
Hirosaki University
miyabi@eit.hirosaki-u.ac.jp

Abstract— Asynchronous MOUSETRAP pipeline circuit is a simple and fast circuit thanks to the 2-phase handshaking protocol which has no return-to-zero overhead. In this paper, we propose two scan D-latches in order to support its scan test since D-latches are used instead of flip-flops in the MOUSETRAP. We design some MOUSETRAP pipeline circuits with the ISCAS89 benchmark combinational circuits using 130nm process technologies and show some evaluation results of the overhead and the fault coverage under the single stuck-at fault model.

I. Introduction

Over the past few decades, a considerable number of studies have been conducted on the test method of synchronous circuits based on the global clock, and then some EDA (Electronic Design Automation) tools to support the full-scan test using scan flip-flops have been developed. On the other hand, the concern with asynchronous circuits based on the request-acknowledge handshaking protocol instead of the global clock has been growing since they have a great potential to resolve some of difficulties in synchronous circuit design, e.g., delay variations due to process variations and environmental changes, crosstalk, and EMI (Electromagnetic Interference). However, the test method of asynchronous circuits has not yet been established. Thus, in this paper, we focus on the scan test of asynchronous circuits under the single stuck-at fault model.

When designing an asynchronous circuit, one needs to choose a handshaking protocol with respect to a request signal and the corresponding acknowledge signal. The two main protocols used for handshaking are the four-phase handshaking (signaling) and the two-phase handshaking (signaling) [1, 2]. The former is often preferred by asynchronous circuit designers since it leads to simpler circuit while it suffers from the overhead due to the return-to-zero (initializing) phase. On the other hand, the latter requires no return-to-zero phase so that it has the potential to outperform the former. Actually, MOUSETRAP [3] proves that a high speed asynchronous pipeline can be implemented using the two-phase handshaking protocol, and several similar implementations have been proposed [4, 5, 6, 7]. However, few attempts have been made at test methods in the MOUSETRAP circuits. In [8], Gennette et al. proposed a test method for delay faults in asynchronous pipeline circuits which include the MOUSETRAP circuits. The fault model is different from our target fault model. Feng et al. introduced C^2MOS circuits to the MOUSETRAP scheme and then presented a test method to detect stuck-at faults [9]. However, it requires the C^2MOS circuits which are different from the normal D-latches. To our knowledge, a full scan test method in the MOUSETRAP pipeline circuits has not yet been clearly shown. Thus, in this paper, we propose two scan D-latches in order to support the full scan test, and evaluate the overhead and the fault coverage using the ISCAS89 benchmark circuits and the 130nm process technologies.

The rest of this paper is organized as follows. The MOUSETRAP circuit and constraints are explained in the next section. Section III proposes two scan D-latches and explains about the structure of a full scan MOUSETRAP circuit. Section IV shows some evaluation results. Finally, Section V describes our conclusions.

II. MOUSETRAP pipeline circuit and its scan test

Figure 1 shows the basic structure of a MOUSETRAP pipeline circuit [3] and the timing chart of signal transitions in the three pipeline stages. As shown in Fig. 1 upper part, the circuit can be separated as a handshake control block and a datapath block where its border is represented by the double-dotted line. The former is constructed by a D-latch and an XNOR gate per pipeline stage. A delay element is inserted between D-latches in order to generate the strobe signal which indicates the stable state of the corresponding datapath circuit based on the bundled-data transfer scheme [1]. The request signal and the corresponding acknowledge signal are performed based on the two-phase handshaking protocol as shown in Fig. 1 lower part, i.e., both the rising and falling edges of the signals are valid. The datapath block is constructed by combinational circuits and D-latches instead of flip-flops. The enable signal of D-latches are commonly used in both the control block and the datapath block. An important point to emphasize is the fact that both the handshake control block and the datapath block are constructed by normal D-latches and standard logic gates. There is no special asynchronous element like the Muller's C-element which is often used as a state holding element in asynchronous circuit design. Thus, in this paper, the handshake control block of a MOUSETRAP pipeline circuit is considered as one of datapath circuits in the viewpoint of a full scan test.

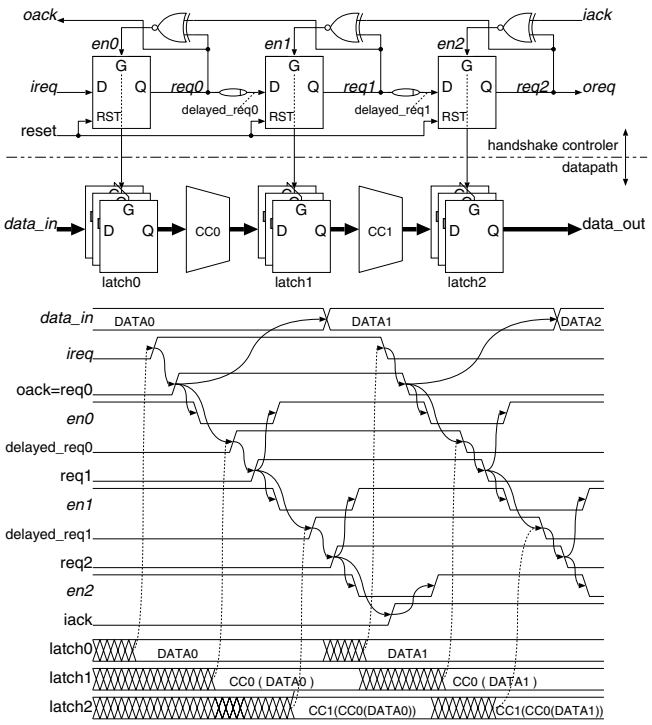

Fig. 1. Basic structure of MOUSETRAP pipeline circuit and its timing chart.

The MOUSETRAP pipeline circuit performs as follows. The D-latches in the control block are initialized by the reset signal. Then, all the enable (gate) signals *en** become 1, i.e., all the D-latches are in the transparent mode initially. When the initializing is finished, input data *data_in* is inserted and the corresponding request signal *ireq* is asserted by a rising edge. As a result, the acknowledge signal *oack* is asserted as a rising edge, and the enable signal of the first pipeline stage *en0* becomes 0, i.e., D-latches in the first stage are in the hold mode. The output request signal *req0* from the first stage is delayed by the delay element whose delay value matches to the critical path delay of the corresponding combinational circuit CC0. Then, the second pipeline stage captures the delayed request signal. As a result, the enable signals *en1* and *en0* become 0 and 1, respectively. It means that D-latches in the first stage are in the transparent mode again. Similarly, in the succeeding stages, the input data is sequentially propagated based on the above procedures as shown in Fig. 1 lower part. What is important is that a request signal and the corresponding data in the first pipeline stage cannot help going through all the pipeline stages at the normal operation period of the MOUSETRAP scheme. In the viewpoint of scan test, test patterns in the second stage and the succeeding stages must depend on the test pattern in the first stage. Thus, it is needed to control each stage at the normal operation period of a scan test.

On the other hand, at least two control signals are needed to transfer data between sequential latches. Typically, latches are separated into master latches and slave latches, and they are controlled by two enable signals whose valid periods are not overlapped. When the master enable signal is 1 and the slave enable signal is 0, a data transfer from the slave latches to the master latches is performed, and vice versa. As the result, the stored values in two sequential D-latches from the slave latch or the master latch must be the same value as shown in Fig. 2.

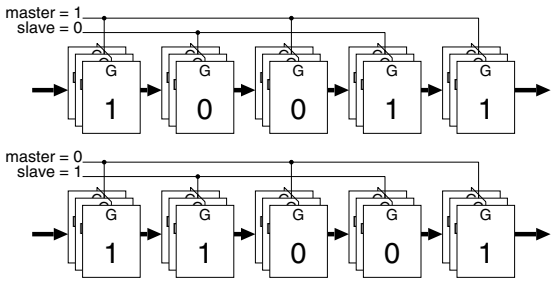

Fig. 2. Data transfer constraints in D-latches.

In this paper, pipeline stages are classified into master stages and slave stages. Then, the scan test for master stages is firstly performed. After that, the scan test for slave stages is performed. Note that all the master (slave) stages can be tested in parallel. In this case, when the input signals to master (slave) latches which belong to the master (slave) stages are issued by other master (slave) latches, additional slave (master) latches should be inserted between them. Thus, when the number of stages is odd, additional latches should be inserted between the last stage and the first stage since both the first stage and the last stage are classified as the same master stages, resulting in increasing the number of D-latches by the bit width. On the other hand, when the number of stages is even, the above additional D-latches are not needed.

Suppose that the number of D-latches in an original circuit is n. When the number of pipeline stages is even, the lengths of a scan chain is n. Generally, the input of a test pattern can be simultaneously performed with the output of the test result on the same scan chain. Figure 3 shows the test procedure of the proposed method when the number of test patterns is three. As shown in Fig. 3, the output of the last test result for the master stages can be simultaneously performed with the input of the first test pattern for the slave stages. Thus, the test time can be represented as follows;

$$(n \times D_{latch} + D_{ope}) \times \{\#\text{ of test patterns}\} \times 2 + n \times D_{latch}$$

where D_{latch} and D_{ope} denote the delay of scan D-latch and the delay of the normal operation, respectively.

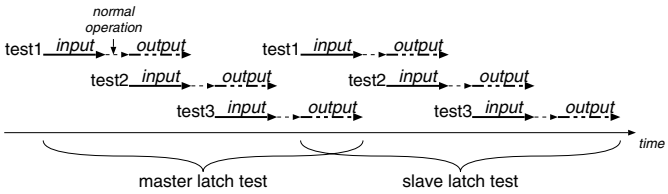

Fig. 3. Test procedure of the proposed method.

III. Scan D-latch for MOUSETRAP

We propose two scan D-latches which satisfy the following conditions in order to implement the proposed test method in the previous section.

Condition 1. It is necessary to keep specific D-latches the hold mode at the normal operation period by the control signals from the outside of a chip. It can stop the propagation of the test pattern which has been stored in the first stage through all the stages during the normal operation period.

Figure 4 shows the structure of our scan D-latch which satisfies the condition 1 and denotes *SD1*. Table I shows the truth table of the scan control logic in Fig. 4. The signal *SC* is 2bit control signals from the outside of a chip.

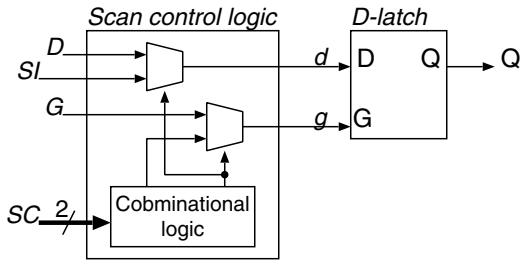

Fig. 4. Scan D-latch which satisfy the condition 1 (*SD1*).

TABLE I
Truth table of scan control logic in Fig. 4.

		SC		*d*	*g*
Normal mode	for normal operation	0	0	*D*	*G*
	for test operation	0	1	*	0
Scan mode		1	0	*SI*	1
		1	1	*	0

As shown in Table I, when *SC* = (0,0), the proposed scan D-latch operates as the normal D-latch in the normal mode. Then, *SC* should be (0,1) in order to keep its stored value at the normal operation period in the scan test scheme. It can properly stop the data propagation. *SC* = (1,0) and (1,1) are used to input test patterns and output the test results using the scan chain. When the mode is changed from the scan (normal) mode to the normal (scan) mode, the lower bit of the signal *SC* should be 1 so that the latch is in the hold mode, resulting in no redundant transitions.

Condition 2. In addition to the condition 1, it is necessary to operate as a simple buffer at the normal operation period by the control signals from the outside of a chip since it is a redundant circuit except for the scan test.

Figure 5 shows the structure of another scan D-latch *SD2* which satisfies the condition 2. Table II shows the truth table of the scan control logic in Fig. 5.

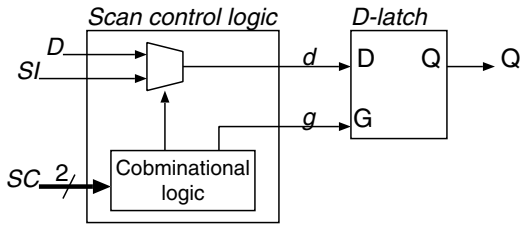

Fig. 5. Scan D-latch which satisfy the condition 2 (*SD2*).

TABLE II
Truth table of scan control logic in Fig. 5.

		SC		*d*	*g*
Normal mode	for normal operation	0	0	*D*	1
	for test operation	0	1	*	0
Scan mode		1	0	*SI*	1
		1	1	*	0

As shown in Table II, when *SC* = (0,0), the proposed scan D-latch operates as the simple buffer since the enable signal of the D-latch is always valid, i.e., the D-latch is always in the transparent mode. The difference between the above two scan D-latches is only the operation under the case where *SC* = (0,0).

Figure 6 shows the structure of the proposed scan MOUSETRAP circuit which contains three pipeline stages as shown in Fig. 1. The first stage and the third stage are defined as the master stages, and the second stage is defined as the slave stage. In the datapath block, the original D-latches are replaced with the proposed scan D-latches *SD1*s which operate as the normal D-latches in the normal mode. Note that the enable signal *g* in Fig. 4 can be shared in the same stage. The scan chain in the datapath block is composed from the least significant bit to the most significant bit as shown in Fig. 6. The *scan_in* is the input port of test patterns. Since the number of pipeline stages is odd, a normal D-latch is inserted between the scan D-latch whose bit number is *i* in the last stage to the scan D-latch whose bit number is $i + 1$ in the first stage. In the handshake control block, the original D-latches are also replaced with the proposed scan D-latches *SD1*s. Then, additional scan D-latches *SD2*s which operate as simple buffer gates in the normal mode are inserted between the replaced D-latch and the corresponding XNOR gate since the output signal of the original D-latch generates an input signal to the same D-latch. In addition, two additional scan D-latches *SD2*s is inserted for the input signal *iack* in order to determine the specific input signals. The scan chain in the control block is alternately connected as shown in Fig. 6.

The area of a normal MOUSETRAP pipeline circuit can be modeled as follows.

$$\{A_d \times (M+1) + A_g\} \times N + (A_{cc} + A_{dc}) \times (N-1) \quad (1)$$

where A_d, A_g, A_{cc}, and A_{dc} denote the area of a normal D-latch, an XNOR gate, a combinational circuit, and a delay element,

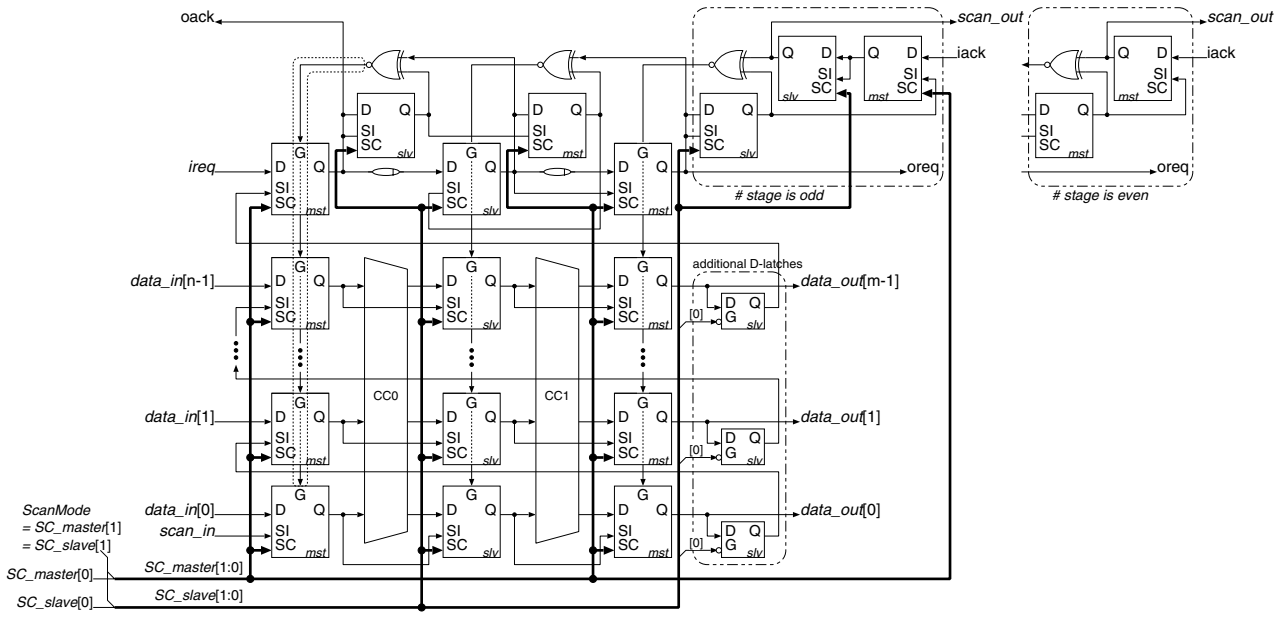

Fig. 6. The structure of the proposed scan MOUSETRAP (pipeline stage number is odd).

respectively. N and M denote the number of stages and the number of bit width in the datapath, respectively. Note that all the combinational circuits are assumed to be the same. The area of the proposed scan test circuit can be formulated as follows.

- The number of stages is even.

$$\{A_{SD1} \times (M+1) + A_g + A_{SD2}\} \times N \\ +(A_{cc} + A_{dc}) \times (N-1) + A_{SD2} \quad (2)$$

- The number of stages is odd.

$$\{A_{SD1} \times (M+1) + A_g + A_{SD2}\} \times N \\ +(A_{cc} + A_{dc}) \times (N-1) + A_{SD2} \times 2 + A_d \times M \quad (3)$$

where A_{SD1} and A_{SD2} denote the area of the proposed scan D-latch *SD1* and the proposed scan D-latch *SD2*, respectively.

The upper bit of the control signals *SC* indicates the normal mode or the scan mode. The mode is the common to both the master stages and the slave stages. Thus, the upper bit of the control signal *SC_master* can be shared with that of the control signal *SC_slave*. Consequently, three input bits are required to control the proposed scan D-latches. Figure 7 shows the signal patterns of the above three control signals. As shown in Fig. 7, *SC_master*[0] and *SC_slave*[0] mutually become valid when *ScanMode* = 1. When *ScanMode* = 0, i.e., in the normal operation period, either signal becomes valid. As a result, the test result is properly captured.

In the viewpoint of the fault coverage, a stuck-at-zero fault on the gate signal in the first stage, which is indicated as the rounded polygon with the dotted line, cannot be tested. This

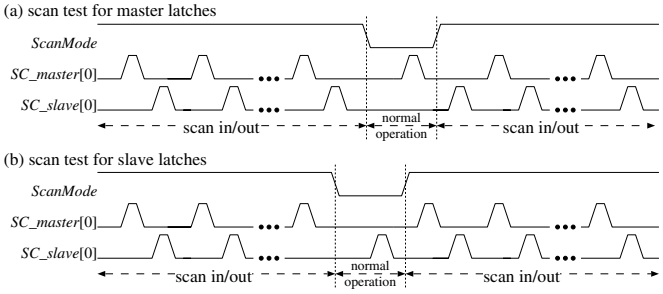

Fig. 7. Control signal patterns.

is because the input values of the first stage are undefined, i.e., the stored data in the first stage at the normal operation period cannot be judged whether it is correct or not when the stuck-at-zero fault occurs on the gate signal. It means that the test pattern in the last stage may be the same as the undefined input values.

IV. Evaluation

In this section, some evaluation results using 130nm process technologies and the combinational circuits of the ISCAS89 benchmark will be shown.

A. Cell characteristics

First, the characteristics of 1bit D-latches are evaluated. The proposed scan D-latches are synthesized by the Synopsys Design Vision. It is assumed that the process variation parameters are taken as the typical values, the voltage of circuits is 1.2V,

the temperature is 25 degrees Celsius, and the output load is FO4 inverters. Table III shows the characteristics of the normal D-latch, the proposed scan D-latch *SD1*, and the proposed scan D-latch *SD2*. In table III, each value represents the ratio value to the normal D-latch instead of the absolute values due to the NDA. The gate signal *G* of *SD2* cannot become the trigger to flip its output signal since the signal always valid. From Table III, it can be said that the overheads of the scan D-latch are large since standard cells are used in this evaluation. It is considered that they can be reduced by transistor level implementation. It is in the scope of our future work.

TABLE III
CHARACTERISTICS OF THE PROPOSED SCAN D-LATCHES.

		normal D-latch	scan D-latch (*SD1*)	scan D-latch (*SD2*)
Area		1.00	2.36	2.14
Delay	D→Q (rise)	1.00	1.56	1.61
	D→Q (fall)	1.00	1.55	2.09
	G→Q (rise)	1.00	1.43	–
	G→Q (fall)	1.00	1.18	–
Power		1.00	3.45	3.02

B. Area overhead

Suppose that the number of pipeline stages are 4 and 5. When no combinational circuit is assumed, the bit width is specified as 32 bits. When there are some combinational circuits, all the combinational circuits between pipeline latches are assumed to be the same circuits of the ISCAS89 benchmark. When the number of output port in the combinational circuit is lower than that of the input port, the shortage bits of the output latches are directly connected to the input latches without combinational circuits.

Table IV shows the area overhead which is represented as the ratio of the proposed scan MOUSETRAP circuits to the normal MOUSETRAP circuits. The overhead decreases as the amount of the combinational circuits increases. When the number of pipeline stages is 4, the area overhead is varied from 1.22 to 2.14 with an average value of 1.49. On the other hand, when the number of pipeline stages is odd, the area overhead is larger than that of the even stage circuits due to the additional D-latches in the datapath block.

C. Fault coverage

The Synopsys TetraMAX ATPG (Automatic Test Pattern Generation) tool is used for the evaluation of fault coverage under the single stuck-at fault model. The tool can generate the corresponding test patterns for a combinational circuit. It also produces fault coverage results as the numbers of *total faults*, *detected faults*, and *undetectable faults*. The fault coverage of a combinational circuit of the ISCAS89 benchmark circuit can be represented as follows;

$$\frac{detected\ faults}{total\ faults} \times 100$$

TABLE IV
AREA OVERHEAD COMPARISON.

circuit	# of ports	# of pipeline stages: 4 stages	# of pipeline stages: 5 stages
nothing	32	1.93	2.12
s27	7	2.14	2.22
s208_1	18	1.71	1.78
s298	20	1.61	1.67
s344	26	1.59	1.67
s349	26	1.60	1.67
s382	27	1.57	1.64
s386	13	1.52	1.57
s400	27	1.57	1.64
s420_1	34	1.59	1.69
s444	27	1.58	1.65
s510	25	1.42	1.47
s526	27	1.53	1.59
s526n	27	1.51	1.58
s641	54	1.65	1.75
s713	54	1.65	1.75
s820	24	1.36	1.41
s832	24	1.36	1.40
s838_1	66	1.53	1.62
s953	52	1.32	1.37
s1196	32	1.28	1.32
s1238	32	1.28	1.31
s1423	91	1.45	1.52
s1488	25	1.22	1.25
s1494	25	1.22	1.24
s5378	214	1.45	1.52
s9234	250	1.41	1.48
s9234_1	250	1.41	1.48
s13207	764	1.55	1.64
s13207_1	762	1.54	1.63
s15850	681	1.46	1.54
s15850_1	681	1.46	1.54
s35932	1763	1.46	1.54
s38417	1664	1.43	1.50
s38484_1	1578	1.36	1.42
s38584	1578	1.36	1.42
average		1.49	1.56

On the other hand, as mentioned above, the fault coverage of the scan block cannot become 100% since the stuck-at-zero fault on the gate signal in the first stage cannot be tested. Thus, the fault coverage of the remaining scan blocks can be represented as follows. Note that "×2" denotes a stuck-at-zero fault and a stuck-at-one fault on each node.

$$\frac{(\#\text{node} \times 2 - 1)}{(\#\text{node} \times 2)} \times 100$$

As the result, the total fault coverage can be calculated as follows;

$$\frac{\{detected\ faults\} \times (\#\text{stage} - 1) + (\#\text{node} \times 2 - 1)}{\{total\ faults\} \times (\#\text{stage} - 1) + (\#\text{node} \times 2)} \times 100$$

Table V shows the total fault coverage of the proposed MOUSETRAP circuits with the ISCAS89 benchmark combinational circuits. They are varied from 89.18% to 99.99% with an average value of 98.90%. From Table V, it can be seen that the fault coverage depends on the circuit structure of the combinational circuits regardless of the amount of the circuits.

TABLE V
FAULT COVERAGE [%].

	scan block	logic	total
only pipeline latch	99.87	–	99.87
s27	99.87	100.00	99.99
s208_1	99.87	83.33	96.20
s298	99.87	100.00	99.91
s344	99.87	99.00	99.57
s349	99.87	99.00	99.57
s382	99.87	100.00	99.91
s386	99.87	100.00	99.90
s400	99.87	100.00	99.91
s420_1	99.87	88.24	95.80
s444	99.87	100.00	99.91
s510	99.87	90.79	97.27
s526	99.87	100.00	99.91
s526n	99.87	100.00	99.91
s641	99.91	100.00	99.95
s713	99.91	99.48	99.74
s820	99.87	100.00	99.91
s832	99.87	100.00	99.91
s838_1	99.93	85.35	94.66
s953	99.91	68.92	89.18
s1196	99.87	96.03	98.34
s1238	99.87	96.03	98.34
s1423	99.95	100.00	99.97
s1488	99.87	100.00	99.91
s1494	99.87	100.00	99.91
s5378	99.98	99.65	99.83
s9234	99.98	99.90	99.94
s9234_1	99.98	99.80	99.90
s13207	99.99	95.96	98.29
s13207_1	99.99	96.06	98.33
s15850	99.99	99.14	99.63
s15850_1	99.99	99.14	99.63
s35932	99.99	99.99	99.99
s38417	99.99	95.56	98.08
s38484_1	99.99	99.37	99.73
s38584	99.99	99.37	99.73
average	99.91	96.86	98.90

V. CONCLUSION

We have presented the test method of asynchronous MOUSETRAP pipeline circuits under the single stuck-at fault model. We have presented two scan D-latches in order to support the full scan test of the circuits. In this paper, we have evaluated the overhead and the fault coverage using the ISCAS89 benchmark circuits and the 130nm process technologies. Our proposed method can control D-latch circuits which operate based on the 2-phase handshaking protocol, and observe them by the full scan scheme. In the proposed method, the proposed scan D-latches are controlled by the control signals from the outside of the chip instead of the internal handshaking signals. Thus, it can be said that the proposed scan D-latches can be used for the scan test not only of other asynchronous circuits, but also of synchronous circuits. Evaluation results show that the area overhead is varied from 1.22 to 2.14 with an average value of 1.49, and it becomes smaller as the amount of the combinational circuit increases. The total fault coverage is varied from 89.18% to 99.99% with an average value of 98.90%. It is in the scope of our future work to reduce the area overhead and improve the fault coverage.

ACKNOWLEDGEMENTS

We thank Tomohiro Yoneda and Satoshi Ohtake for technical discussions. This work is partially supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc. This work is also supported by the Telecommunication Advancement Foundation.

REFERENCES

[1] Scott Hauck. Asynchronous design methodologies: An overview. *Proceedings of the IEEE*, 83(1):69–93, January 1995.

[2] Editted by Jens Sparso and Steve Furber. *Principles of Asynchronous Circuit Design –A Systems Perspective*. Kluwer Academic Publishers, 2001.

[3] Montek Singh and Steven M. Nowick. MOUSETRAP: Ultra-high-speed transition-signaling asynchronous pipelines. In *Proc. International Conf. Computer Design (ICCD)*, pages 9–17, November 2001.

[4] Yijun Liu. An asynchronous pipelined 32x32-bit iterative multiplier using hybrid handshaking protocol. *14th UK Asynchronous Forum*, 2003.

[5] Gennette Gill, John Hansen, Ankur Agiwal, Leandra Vicci, and Montek Singh. A high-speed GCD chip: A case study in asynchronous design. *Proc. ISVLSI09*, pages 205–210, May 2009.

[6] Michael N. Horak, Steven M. Nowick, Matthew Carlberg, and Uzi Vishkin. A low-overhead asynchronous interconnection network for GALS chip multiprocessors. *Proc. NOCS2010*, pages 43–50, 2010.

[7] Masashi Imai and Tomohiro Yoneda. Improving dependability and performance of fully asynchronous on-chip networks. *Proc. Async2011*, pages 65–76, Apr. 2011.

[8] Gennette Gill, ANkur Agiwal, Montek Singh, Feng Shi, and Yiorgos Makris. Low-overhead testing of delay faults in hgih-speed asynchronous pipelines. *Proc. ASYNC06*, pages 46–56, Mar., 2006.

[9] Feng Shi and Yiorgos Makris. A transistor-level test strategy for C^2MOS MOUSETRAP asynchronous pipelines. *Proc. ASYNC06*, pages 57–66, Mar. 2006.