SCAN TESTING OF LATCH ARRAYS

Michael M.Y. Hui Benoit Nadeau-Dostie

Bell-Northern Research Ltd.

Abstract

A novel scan-based test method that allows latch-based arrays to be fully tested using conventional scan software tools is described. Very little support circuitry and simple modeling are required. Several useful applications of the method are described. The method was used on several production chips.

Introduction

A digital integrated circuit designed to be tested using scan design techniques must be decomposable into a set of registers and combinational blocks [1,2]. A popular style of scan, known as the Stanford style, requires the use of D-type flip-flops as the only valid storage element for registers. The registers are connected such that they operate as shift registers in a special mode of operation of the circuit called scan mode. In scan mode, the state of the registers can be controlled (observed) from a serial test input (output) pin. Figure 1 shows an example of such a circuit. Only one shift register (or scan chain) is shown for simplicity. The main advantage of this test method is that only the combinational part of the circuit needs to be considered during the test generation step. It is well known that the complexity of test generation for combinational circuits is significantly less than for sequential circuits.

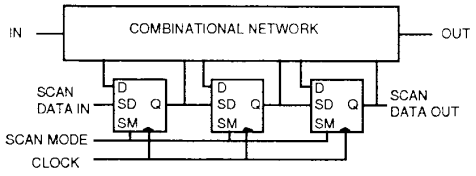


Figure 1

This test method can become expensive for register intensive designs. It would be nice to relax the requirement that imposes the use of full "scannable" D-type flip-flops for every storage element of the circuit under certain circumstances and yet be able to fully test these "illegal" storage elements using the same software tools. For example, small FIFOs, addressable control/status registers and other storage blocks are commonly found in digital circuits. Although these circuit structures can be built using arrays of "scannable" D-type flip-flops, it is more attractive to use transparent latches from a silicon area point of view (3 to 1 ratio on average). Embedded memories could also be used in some cases but they would require different test generation tools because of their specific failure modes [3-5]. Also, embedded memories might not be available for a particular technology, especially when gate-arrays are used.

We have devised a scan-based test method that allows latch-based arrays to be fully tested using conventional scan software tools. With the addition of support circuitry and some restrictions on how the latches may be connected, each latch may be modeled as a combinational gate to the pattern generation software.

This paper is divided as follows. The method is introduced in section II. Applications of the method are described in section III. Results are presented in section IV followed by the conclusion.

Method

The principle of the method is to make the transparent latches used to build the various registers and arrays look like combinational gates. This principle was used in [6]. In this paper, it was proposed to force all latches into their transparent mode for the whole duration of the scan test to verify the interface between the array and the rest of the circuit. However, this method can not test the ability for the latch to hold any data.

0-7803-0623-6/92 $3.00

To accomplish this, it is required to apply a sequence of patterns at both the gate (clock) and data input of the latch. For example, in order to test for the gate input SA1 fault (latch always transparent), we need to write a value into the latch (gate input = 1) with the first pattern and observe that this value doesn't change in a second pattern where the latch is in "hold" mode (gate input = 0) and the data input is set to the complement value.

In our method, we propose to preload all latches with a background value before each scan pattern is applied. By proper modeling (explained later), the background data is seen as a regular input to the combinational circuit.

The circuitry needed to load the background data for a small 2x2 latch-based array is shown in figure 2. The normal input data bus is intercepted with multiplexers that select between the background and normal data. For each row of latches, the write enable signal is connected such that it can be forced active by the FORCE WRITE signal. The sequence of events on the various input/output ports is described by the timing diagram of figure 3. For this first example, we suppose that the scan test is controlled by a Test Access Port (TAP) designed to support the IEEE 1149.1 standard [7]. The names shown at the top of the timing diagram reflect the state of the TAP. All flip-flops output new data on the falling edge of the clock and sample on the rising or the falling edge of the clock.

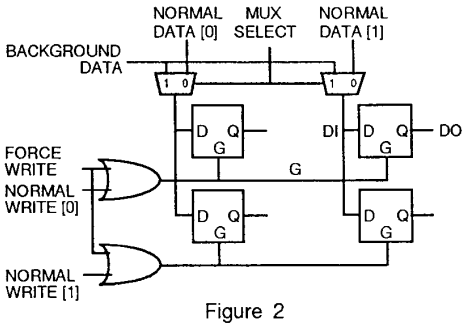


Figure 2

The latches are preloaded by briefly setting FORCE WRITE to logic 1, with the data multiplexers selecting BACKGROUND DATA, before entering the Capture-DR state. During the Capture-DR state, FORCE WRITE stays at logic 0 while the multiplexers select the NORMAL DATA input, hence the latch's stored state may be overwritten depending on whether the NORMAL WRITE input is active.

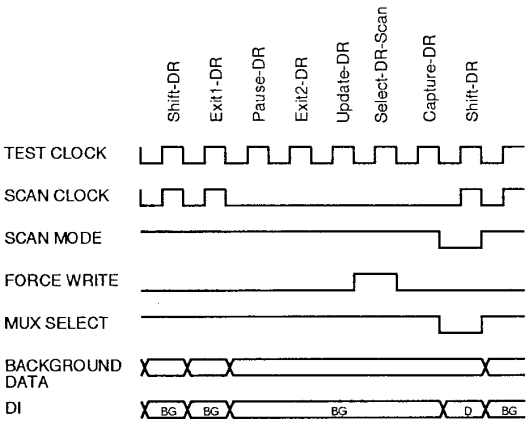


Figure 3

A variety of techniques may be used to generate the FORCE WRITE signal. We chose to assert FORCE WRITE during the Update-DR state. The select signal for the data multiplexers (MUX SELECT) must ensure that the background value is stable when FORCE WRITE is released. Connecting MUX SELECT to SCAN MODE is sufficient. The background value written into all the latches must be the "final" value present on that node when the Capture-DR state occurs, hence FORCE WRITE must occur after the last shift operation on the scan chain and no sooner.

Note that SCAN MODE is different from the Shift-DR signal normally generated by TAP. This is needed to accommodate the particular method we are using to test on-chip tri-state bus drivers [8]. We are able to adequately test bus drivers using a combinational test pattern generator.

The timing shown in figure 3 is not the only possibility to implement our method. For example, figure 4 shows a simpler scheme where the IEEE 1149.1 standard TAP circuit is not needed to control the scan sequence. All flip-flops on the scan chain now sample and output new data on the falling edge of the clock. This scheme only has a single period, corresponding to the Capture-DR state, where the SCAN MODE is at logic 0. The scan chain shifts both before and after this one period. Here, MUX SELECT cannot have the same timing as SCAN MODE any more. A new signal generated from SCAN CLOCK and SCAN MODE now drives both MUX

SELECT and FORCE WRITE. This timing variant is more suitable to perform "at-speed" testing and we actually modified our TAP to accommodate it as well.

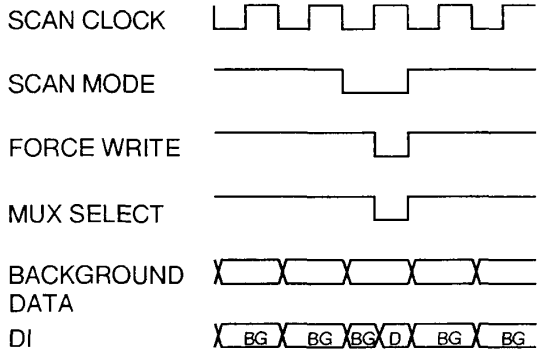
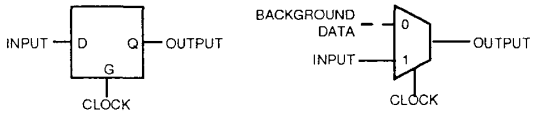


Figure 4

Substituting a combinational circuit for the latch allows an automatic test pattern generation (ATPG) program designed only for processing combinational circuits to be used to fully test the latch. When the support circuitry shown in Figure 2 is in place, each latch connected in that fashion will be modelled as a multiplexer, as shown in Figure 5, when the entire chip's net list is analyzed by the ATPG program. The BACKGROUND DATA input for the scan model will have to be supplied by modifying the net list accordingly before the ATPG program processes it (a few text editor commands will do for regular arrays). In order to guarantee the consistency of the functional models and the test generation models, the designers are required to generate the patterns using the test generation models and simulate the patterns using the functional models. This step is performed at the macro block level to avoid having to debug the scan patterns at the chip's top level.



Actual Circuit Scan Model

Figure 5

The multiplexers used to intercept the input data bus and the gates used to force a write operation are not explicitly considered during test generation. When the combinational part of the circuit is extracted using our rules checker [8], the Update-DR and Shift-DR signals are set to 0 and these constant values are propagated through the circuit. The implication is that FORCE WRITE and MUX SELECT are found to be 0 and the logic pruned accordingly. For example, the multiplexers and OR gates in figure 2 become buffers connecting the NORMAL WRITE and NORMAL DATA signals to the latch array. However, all the gates are implicitly tested because of the particular pattern sequencing used.

BACKGROUND DATA and NORMAL WRITE may be the output of any flip-flop on the scan chain or the output of any combinational logic. So far, we assumed that a single BACKGROUND DATA value was sufficient to initialize all transparent latches of the circuit. However, since the output of these transparent latches reconverge for most applications described in the next section, there exists cases where a different BACKGROUND DATA net is needed for some registers. A situation requiring more BACKGROUND DATA signals would be identified by having several faults being declared redundant by the ATPG.

Applications

In this section, we demonstrate the usefulness of the test method using 3 examples: a dual-port memory that can be used as a FIFO, a collection of addressable control/status registers and an orthogonal memory.

An array of latches, each with a tristate driver connected to its output, can form a static RAM structure as shown for a 3 words x 4 bits dual-port memory in Figure 6. The bottom left of the figure shows the internal structure of each RAM cell. The output bus (DO) is pulled up/down according to our test method for bus drivers described in [8]. Furthermore, all bus drivers are turned off when SCAN MODE is active so that the weak pull-up/pull-down can precharge the output bus. Each address decoder has at most one output active at a time.

If the read and write addresses are the same, and both address decoders are enabled, then one word (or row) of this RAM will drive the data output bus with the normal data input value during the Capture-DR cycle, since every latch in that word will be in transparent mode. If the addresses are different, the background value will be driven out by the word enabled for reading, since another word is being written with the normal data value. All latches can then be tested completely.

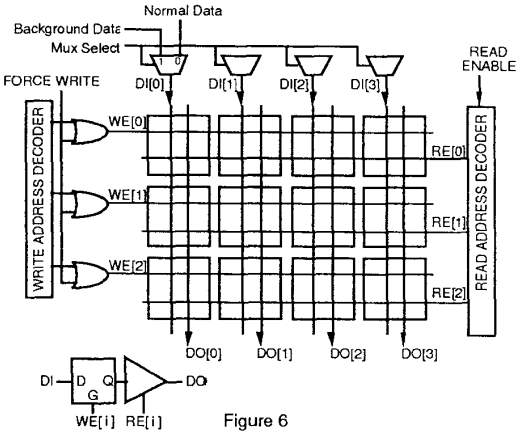


Figure 6

Faults in the address decoders can also be detected. If the write decoder is defective, wrong data will be sensed by logic connected to the DO outputs when the wrong word is written or no word is written. This includes the case where more than one word is written at a time.If the read decoder is defective and no words are placed in read mode, the logic connected to the DO outputs will sense the pull-up/pull-down value when the read and write addresses coincide. This is incorrect since that is when the value at the DI inputs should be present on the DO outputs. If the read address decoder is defective in a way that two or more words are read simultaneously, then a bus contention will arise on the DO outputs if one of these words is also enabled for write. Faults causing this behavior are reported as being potentially detected only. This problem goes away if the tri-state bus drivers are replaced by a multiplexer ring.

For the second application, we examine the case of addressable control/status registers. Typically, these registers are accessible by an external processor through a bidirectional bus and they are distributed throughout the chip to control and observe the various status bits of the blocks implementing the chip's function. The only relevant features of this set of registers are that a single address decoder is available to perform both the read and write operations and that the output of more than one word can be observed at a time. In the previous application, we relied on the two address decoders to test each other since the only observability mechanism we had was the output data bus DO. When only one decoder is used for both read and write, the correct operation of the decoder is not tested, unless other means allow monitoring the decoder's output. This is the case for the control/status registers discussed here because most of them are directly connected to combinational logic blocks that provide the additional observability.

There are some restrictions on how the registers' output can be connected to the combinational logic blocks. Since only one address decoder output may be active at a time, only one word will be written, and hence only one word's latches' outputs may be at a logic value other than the background data value. An example of a circuit that will not be fully tested is

$$(A \text{ xor } B) \bullet (C \text{ xor } D)$$

where A to D are individual outputs of latches that have common inputs. Since only one of A to D may be different from all others, the AND gate's output can not be driven to logic 1 during scan test. Some of the faults of that circuit would be declared redundant because of the reconverge of the BACKGROUND DATA signal that is common to all latches. This is only an artifact of the test method used and these faults must be tested for. There are 2 ways to get around this problem. The first one is to segment the input data bus such that A, B, C and D belong to a different group of registers that would share a common background data. The other one is to selectively replace some of the transparent latches by scannable D-type flip-flops.

The third application is an orthogonal memory. Figure 7 shows a memory where the data input lines are orthogonal to the data output lines. This orthogonal array is area-efficient for converting between serial and parallel data formats, with the added bonus of providing a space switch function. The separate flip-flops controlling the write enable lines allow full controllability for the array's data output bus, rather than restricting it to a 1 in N pattern only. This allows any combinational logic to be connected to the DO outputs and be fully tested by scan test.

Results

Several designs have used this new test method in the past three years, and volume production has shown that this technique does provide robust testing of the latch arrays. The area savings provided by this method are circuit dependant. It is most appropriate for chips that have a collection of small arrays that can't justify

the use of a custom memory (if available) but still large enough so that scannable flip-flops are not a reasonable option.

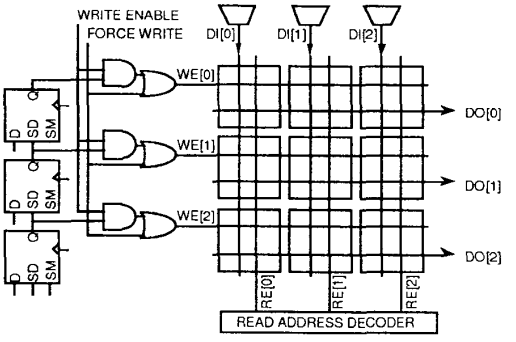


Figure 7

As an example, a scaled-down version of the first product chip (a gate-array with no option for custom memories) to use this method had the following memory configuration:

Dual-port memory	Orthogonal memory
two 9x9	one 21x9
one 29x30	one 14x18
two 9x5	
one 29x2	

for a total of 1485 bits of storage. The use of flip-flops was clearly not an option because the size of this 400x400 mils 1.5-micron gate-array would have been exceeded. The area was reduced by about 8% by using transparent latches to make the chip fit. Less than half of the arrays were accessible from the chip's pins. Providing external access for each array for testing would have increased chip size and required manual test generation.

The number of test clock cycles required to test this chip is not significantly different from other chips with a similar number of gates. This is about 1 million clocks cycles for 20K gates. There was 545 flip-flops in the scan chain, and 1969 scan patterns were generated using our deterministic test pattern generator. These patterns provided 97.8% definite stuck-at fault coverage and an additional 1.13% potential fault coverage because of bus conflicts. The 1.06% of faults remaining is redundant. Our test method increases the width and depth of the combinational circuit seen by the ATPG because the transparent latches do not provide the same level of controllability-observability than flip-flops. On the other hand, we have less flip-flops in the scan chain. So, on average, the two effects seem to cancel each other for the applications we have seen so far. In terms of random pattern susceptibility of the chip, 65536 random scan patterns are needed to achieve a coverage of 97.65% (96.5% definite and 1.15% potential). This satisfies our requirements for board and system-level testing [9].

In general, the number of words of each array should be kept small (say less than 64 words) if random pattern testability is required. This is because the number of patterns required to test the address decoders is $O(w^2)$ (w is the number of words). Otherwise, additional test points might be needed [10].

Conclusion

We presented a novel scan-based test method that allows latch-based arrays to be fully tested using conventional scan software tools. Very little support circuitry and simple modeling are required. Several useful applications of the method were described including dual-port memories, addressable control/status registers and orthogonal memories. The method was used on several production chips. In one example, it was shown that the silicon area required to implement the chip was reduced by about 8% compared to an implementation with scannable flip-flops.

Acknowledgements

The authors would like to thank their managers Mr. Jim Dilley and Mr. Phil Wilcox for allowing the product development schedule to stretch in order to incorporate this test method into the product as it was being devised and refined, Mr. Brian McKinney and Larry McNaughton for their help with the implementation of the first chip using this test method, Mr. Jean-Yves Ouellet for careful review of this paper and his helpful suggestions, and all the anonymous reviewers for their valuable suggestions.

References

[1] E.B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability", Proc. 14th Design Automation Conference, June 1977, pp. 462-468

[2] S. Funatsu, N. Wakatsuki and T. Arima, "Test Generation in Japan", Proc. 12th Design Automation Conference, June 1975, pp. 114-122

[3] E.K. Vida-Torku & al., "Test Generation for VLSI chips with Embedded Memories", IBM J. Research & Development, Vol. 34 No. 2/3, March/May 1990, pp. 276-287

[4] F.P.M. Beenker & al., "Macro Testing: Unifying IC and Board Test", IEEE Design & Test, December 1986, pp. 26-32

[5] B. Nadeau-Dostie, A. Silburt, V.K. Agarwal, "A serial interfacing technique for embedded memories", IEEE Design & Test of Computers, April 1990, pp. 56-64

[6] B. Nadeau-Dostie & al.,"Practical Scan Design for ASICs", New Directions for IC testing workshop, Halifax, Canada, 26-27 October 1988

[7] IEEE std 1149.1-1990, "IEEE standard test access port and boundary-scan architecture"

[8] B. Nadeau-Dostie & al.,"Scan Design Software for ASICs", Canadian Conference on VLSI'89, Vancouver B.C., Canada, 22-24 October 1989, pp. 3-8

[9] B. Nadeau-Dostie, P.S. Wilcox, V.K. Agarwal, "A Scan-Based BIST technique Using Pair-Wise Compare of Identical Components", Fourth CSI/IEEE International Symposium on VLSI Design, New Delhi, India, January 4-8, 1991, pp. 225-230

[10] B.H. Seiss, P.M. Trouborst, M.H. Schulz, "Test Point Insertion for Scan-based BIST", IEEE European Test Conference, Munich, Germany, 1991, April 10-12, pp.253-262