Extending Boundary-Scan to Perform a Memory Built-In Self-Test

HENNING BAHR, GORDON RUSSELL, YAJIAN LI

School of Electrical, Electronic and Computer Engineering University of Newcastle Upon Tyne UNITED KINGDOM

Abstract: We present a novel test architecture which combines IEEE 1149.1 Boundary-Scan with a Memory Built-In Self-Test. The TDI pin is used for serially shifting in the test data into a test data register which is connected to the memory. The finite state machine of the TAP controller performs the memory test algorithm. The test response is shifted out via the TDO pin for off-chip analyses. The test architecture offers small area overhead, acceptable test time, increased flexibility and analysis capabilities while maintaining compliance to the Boundary-Scan standard.

Keywords: Design for Test, Memory Test, Boundary Scan, IEEE 1149.1, JTAG, Built-In Self Test, BIST

1. INTRODUCTION

The growing cost of designing, producing and testing digital systems is increasingly alleviated by Designfor-Testability (DFT) methods such as Boundary Scan and Built-In Self-Test (BIST). These techniques improve the testability of a chip, make debugging and diagnosis easier, achieve higher fault coverage and therefore produce higher quality parts. Increasingly more functional components are moved from the board-level to the silicon die to build a System on Chip (SOC). This presents a test challenge due to the heterogeneous and complex structures of SOCs. The key factors in deciding which test methodology is the most suitable are area/performance overhead due to the added test logic, test application time and the limitations of automatic test equipment (ATE). Other parameters are power dissipation, routing impact and additional I/O pin count.

Boundary-Scan Test (BST, also called JTAG) is mainly used as a board level test for testing interconnections but it can also be used for on-chip testing and debugging [1]. It became the IEEE standard 1149.1 in 1990 and is now widely applied in the IC industry. The IC pads and the core logic are connected via Boundary Scan Cells (BSCs). All BSCs can be combined in a shift register with parallel inputs and outputs and a serial input and output. A test access port (TAP) is integrated into the chip to stimulate and read-out the cells. It only requires 5 (4 when the test reset is omitted) standardised test signals to operate the BST. The standard allows extension by adding user-defined instructions and user-defined register. As more and more memories are implemented in a SoC, embedded memory testing emerges as a key issue in VLSI design. It is estimated that embedded memories account for more than 60% of the silicon area of modern SOCs [2]. Basically three different schemes are employed for embedded memory testing [3]:

1. Direct Access Testing: The memory signals are routed to the I/O pins so that ATE can perform the memory test externally. The input test vectors and correct response data is stored in the ATE memory. This method becomes increasingly prohibitive when employing multiple memories because of the routing impact and the limited number of I/O pins. Moreover, this method depends on high performance and therefore expensive ATE.

2. Re-using existing on-chip resources: Hardware which already exists on the chip such as a microprocessor is employed to perform the memory test [4]. This method reduces the area and performance overhead on the expense of increased testing time. The test problem is partly shifted to the software development level and it is dependent on the resources on-chip which adds complexity to the design flow.

3. Stand-alone memory BIST: It is commonly a self-contained system where the test vectors are created and test responses are analysed on-chip. An external tester is only needed to initiate the test and to observe the final result (pass/fail signal). Although this method allows the use of inexpensive ATE, the drawback is the added test logic. This

increases the overall cost of the chip, dissipates additional power and limits the performance.

The new test architecture called BMTSCAN (Boundary Memory Test Scan) combines some of the advantages of the above testing methods with IEEE 1149.1 Boundary-Scan. The generation and analysis functions are moved off-chip while boundary scan is giving access and control to the memories via the TAP. A memory test data register generates the test vectors and collects the responses. This facilitates the implementation of a memory test only by using the standard Boundary-Scan interface signals. Low-cost ATE can be used because only these five signals need to be accessed. Assuming that JTAG is implemented on the chip in any case, it significantly reduces the area overhead. The serial nature of the architecture (similar to [6]) reduces the routing congestion at the expense of a longer test time. Since BST is a standard it is independent of the core logic and is fairly effortless to implement.

2. EMBEDDED MEMORY TESTING

The three main areas where memory failures occur are the address decoder logic, the memory cell array, and the read/write logic. The basic types of memory faults include stuck-at, transition, coupling (including inversion, idempotent, bridging, and state faults), address decoder and neighbourhood pattern sensitive faults. One of the most efficient test algorithms belong to a group called March tests. The March C- algorithm, illustrated in Table 1, is a fair trade-off between complexity and fault coverage [5]. It detects address, stuck-at, transition, coupling, and unlinked coupling faults. The complexity equals 10n (12n when the redundant operations Rx and Wx are included), where n is the number of locations in the memory.

Table	1:	March	C-	test	al	gorithm
-------	----	-------	----	------	----	---------

March Element	Memory Operation ^a
1	Rx Wo
2	R0 W1
3	R1 W0
4	R0 W1
5	R1 W0
6	R0 Wx
- D	

a. R = read operation, W = write operation, x = don't care

The realisation of the March C- algorithm in this architecture for testing embedded memories is as follows: 1. Read memory: The read data is stored in a data register which is part of the Boundary-Scan architecture.

2. Shift data: The stored data is serially clocked out via TDO for response analysis while a new data background is simultaneously clocked in via TDI.

3. Write memory: The content of the data register is written to the specified address.

4. Increment (decrement) the address.

5. Repeat step 1-4 until the highest (lowest) address and then start again with a different data background or a new March element until the March algorithm is finished.

It is feasible to implement other March algorithms without changing the test architecture, i.e. after the fabrication of the chip, by changing the BST program. Also, depending on the BST program, the March test can be performed bit-oriented or word-oriented. In the remaining test it is assumed that a word-oriented test with different data backgrounds is used.

3. TEST ARCHITECTURE

Figure 1 shows a simplified block diagram of the BMTSCAN architecture applied to a single SRAM of arbitrary size. All non-testing signals of the SRAM are omitted in the diagram. The block BMT_CONTROL consists of the normal 1149.1 modules, i.e. TAP, instruction register, instruction decoder and bypass register.



Figure 1 Block diagram of BMTSCAN

A new instruction RUN_MBIST is added to the Instruction Register decoder. If this instruction is loaded the signal mbist_on is set to '1'. This switches the test collar multiplexers of the memory (assumed to be integrated) to BIST mode, i.e. the BIST data and address signals and the BIST write enable signal are active.

The modules added to the Boundary Scan architecture are the boundary memory test (BMT) register and the address counter. The latter is an up/down binary counter with a size equal to the number of addresses of the SRAM. The counter incorporates a small builtin finite state machine to change the direction of the counter after three March elements. The state transition diagram is shown in Figure 2. Both the counter and FSM are returning to their initial state when the test reset nTRST is low.



Figure 2 State transition diagram of the address counter

The circuit diagram of the BMT register is illustrated in Figure 3. The data register cell is similar to a Boundary-Scan register cell. However, the Update Flip-Flop and the muliplexer selected by the signal Mode are omitted, i.e. only half of the normal BS-cell is employed. The cells are connected in series to form a shift register with a serial input/output and parallel inputs/outputs. The length of the register is equal to the length of a data word of the memory.



Figure 3 Circuit diagram of the BMT register

The BMT register can be placed close to the memory under test to reduce wire delay and routing congestion. The routing congestion is further reduced because only two signals (shiftDR and clockDR) need to be connected to the TAP controller.

The TAP is extended only by a memory write/read enable signal to control the memory and a further signal inc_addr to control the address counter. Figure 4 shows the state transition diagram of the data column of the TAP with the added MBIST operations printed in italics.)



Figure 4 State transition diagram of the TAP (only data column

Each round begins with a read operation while the state of the TAP is in CAPTURE-DR, i.e. the BMT register is parallel loaded with a data word of the memory. The next state is SHIFT-DR which shifts the read data serially out to TDO while a new data background is shifted in via TDI. Thus, the new data background can be freely chosen by the test engineer by controlling TDI. If a bit-oriented test is used only one bit would be shifted in and the state machine would immediately change to the next state EXIT-1-DR. This state asserts the write enable signal of the memory and writes the new data background to the specified address. Finally, in PAUSE-DR, the signal inc_addr is set high in order to increment the address by the address counter. The algorithm would then start again for the next address of the memory. Notice that it is possible to omit PAUSE-DR altogether and read and write again at the same address to accommodate different March tests. The architecture, however, allows only March elements with a least one read and write operation. Furthermore, the state PAUSE-DR can be repeated (TMS = 0) in order to skip to a specific memory address. This allows to analyse and debug a specific address of the memory. Ultimately the design also allows to load and unload the memory without the need for an additional bus for programming the memory.

Figure 5 shows an extract of a VHDL test bench as an example of how to operate the BST signals for memory testing. The second March element (read 0, write 1) of March C- is performed in this test bench on a 256 x 16 SRAM.

```
-- March Test Element 2: (R0, W1)
TDI <= TDI_TMS(1);
TMS <= TDI_TMS(0);
for i in 0 to 255 loop
   TDI_TMS <=
-- Select-DR:
   ('1', '1') after 1 ns,
-- Capture-DR:
   ('1', '0') after 101 ns,
-- Shift-DR:
-- shift in 16 '1s' into BMT_REG and
-- observe 0's at the output (TDO)
   ('1', '0') after 201 ns,
-- Exit1-DR: write enable = 1
   ('1', '1') after 1801 ns,</pre>
```

```
-- Pause-DR: increment address
  ('1', '0') after 1901 ns,
-- Exit2-DR:
```

```
('1', '1') after 2001 ns,
```

```
-- Update-DR:
```

```
('1', '1') after 2101 ns;
wait for 2.2 us;
end loop;
```

Figure 5 Extract of the VHDL Test Bench to perform the second March element (R0, W1)

The clock cycle of TCK is 100 ns. It takes 22 clock cycles to test one address. In general the number of clock cycles to perform a complete March C- test using the TAP controller is

$$6 \cdot (n \cdot (6 + w)) \tag{1}$$

where n is the number of addresses and w is the word length of the memory.

4. INTEGRATION INTO A SYSTEM ON CHIP

The BMTSCAN architecture has been implemented into a SOC for Diesel injection control. The chip is specified for high temperatures (temperatures above 200° C) and is fabricated in a 1 micron SOI technology. The digital core consists of approximately 7000 gates and 4 memories. Two of them are 256x16 single port SRAMs (SPRAM) and the other two are 32x16 dual-port SRAMs (DPRAM). Since the memories are relatively small, the area overhead is particularly critical and the testing time of less concern. Distribution and reuse of resources is essential to overcome this issue. Figure 6 shows a simplified block diagram to illustrate how the four memories are connected together to implement the BMTSCAN.



Figure 6 BMTSCAN implemented into a SoC testing multiple memories

The architecture employs six BMT registers and two address counters. Both SPRAMs are grouped together, i.e. both are connected with the same address counter (8 bit) and their BMT registers are connected in series. Hence the test is performed in parallel as if it is one memory with a 32 bit word length. The same procedure is applied to the two DPRAMs with a BMT register for every port and 5 bit address counter. Accordingly the test runs as if the test algorithm is performed on a 64 bit memory. Two new instructions are implemented RUN_MBIST_SP and RUN_MBIST_DP for both clusters of memories. Together with the obligatory Boundary-Scan modes EXTEST, SAMPLE_PRELOAD and BYPASS this demands a 3 bit Instruction Register/Decoder. It would be relatively simple to add more memories with the same address space and word length to one of these groups. A new group with a different specification could simply be formed by adding another instruction to the IR.

Table 2 lists the silicon area of the memories and the BMTSCAN modules. The added area to the Boundary-Scan control circuitry is negligible. All the test logic would encompass approximately 11% of the design. Considering that Boundary-Scan is employed for the SOC for a board and logic test in any case it is appropriate to calculate the are overhead only by taking account of the BMT registers and address counters. The silicon real estate of these modules equals 630,000 square micron and this leads to a silicon overhead of approximately 7%. The last row of the table presents the area overhead of a conventional MBIST architecture for comparison (the design is based on a finite state machine). This design consists of a finite state machine to run the March test and comparators built from EXOR trees for the evaluation of the response data. The area overhead of BMTS-CAN is about 40% less.

Module	Silicon overhead in μm^2
Total area of memories	8,400,000
BMT_CONTROL	400,000
6x BMT_REG	450,000
ADDR_CNT_SP	110,000
ADDR_CNT_DP	70,000
Total Area (TA): BMTSCAN	1,030,000
TA: MBIST modules only (excluding BMT Control)	630,000
TA: Conventional MBIST	1,090,000

 Table 2: Silicon area comparison

The conventional MBIST needs considerably less time to perform the test. It takes 300 us testing all memories in parallel with two different data backgrounds. Testing both SRAMs with BMTSCAN takes approximately 3 ms when the test clock TCK is clocked at 10 MHz. Performing the March C- test for all six memories takes 4 ms (word-oriented test, two different data backgrounds).

5. CONCLUSION

This paper presented a novel test architecture for integrating Boundary-Scan with a memory built-in selftest scheme. The standard Boundary Scan architecture can readily be converted to implement this test strategy by adding an additional test data register and an address counter. The test scheme, overall, is controlled by the finite state machine in the Boundary-Scan TAP controller. No extra test I/O pins are required other than the standard boundary scan interface signals. The standardised IEEE 1149.1 interface to perform the test allows remote testing and fault analysis/debugging. It is also possible to load and unload a memory via the BST port. The routing impact is limited since the BMT registers can be placed close to the memories and only two serial lines and 4 control signals are routed to a central controller. The silicon overhead is less compared to a standard MBIST circuit on the expense of extended test time. The implementation is independent of the digital core. The possibility to perform a memory test only by using the five Boundary-Scan interface signals should make this testing method particularly attractive for field testing and for testing with limited ATE capabilities.

Acknowledgements:

This research is part of the EU funded Advanced Techniques for High Temperature System on Chip (ATHIS) Project No. G1RD-CT-2002-00729.

References:

- [1] Parker, K. P., "The Boundary-Scan Handbook", Kluwer Academic Publishers, 2003
- [2] Rajsuman, R., "Design and Test of Large Embedded Memories: An Overview", IEEE Design and Test of Computers, May-June 2001
- [3] Crouch, A. L., "Design-for-Test for digital IC's and embedded core systems", Prentice Hall, 1999
- [4] Rajsuman, R. "Testing a System-on-Chip with Embedded Microprocessor", IEEE International Test Conference, p. 499-508, 1999
- [5] van de Goor, A. J. "Testing Semiconductor Memories", John Wiley & Sons, 1991
- [6] Nadeau-Dostie B., Silburt, A., Agarwal, V. K.,"Serial Interfacing for Embedded Memory Testing", IEEE Design & Test of Computers, April 1990
- [7] Abramovici, M., Breuer, M. A., "Digital Systems Testing and Testable Design", IEEE Press, 1995
- [8] Dekker, R., Beenker, F., Thijssen, L., "Realistic Built-In Self-Test for Static RAMs", IEEE Design and Test of Computers, February 1989