Military Technical College
Kobry El-Kobbah
Cairo, Egypt

10th International Conference
On Aerospace Sciences&
Aviation Technology

Architecture for the BIST Boundary Scan

Mohamed H. El-Mahlawy*

Abstract

The boundary scan (BS) technique, formally known as IEEE-1149.1 Standard, offers a convenient alternative to physical probing by effectively migrating the test probe circuitry into the chip which enables a non-contact method of accessing chip pins for testing. In this paper, the incorporation of Built-In Self-Test (BIST) capabilities into the boundary scan architecture is presented. The Boundary Scan Register (BSR) input cells have been configured to operate as a Test Pattern Generator (TPG) in the BIST mode. The BSR input and output cells have been configured to operate as an Muti-Input Shift Register (MISR) in the BIST mode. The Tape Controller (TAPC) controls the BIST process. Instructions for BIST process are proposed. This configuration supports BIST for both the cascaded and non-cascaded input and output cells of the BSR.

1. Introduction

Advances in VLSI technology have led to the fabrication of chips that contain a very large number of transistors, integrated on a single chip. The cost of testing such devices increases with complexity so the incorporation of BIST capability [1] inside a chip is increasingly desirable, which is a *design for testability* technique. BIST requires hardware overhead to incorporate a TPG [2-3] such as in Fig. 1, a test response compactor [2] [4] such as in Fig. 2, and a BIST controller into the system (core) logic to realize self test operations. (Fig. 1 and Fig. 2 have c_i's as binary constants, $c_i = 1$ implies that a connection exists, while $c_i = 0$ implies that there is no connection.)

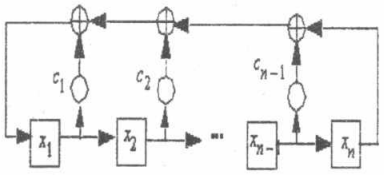

Fig. 1. Autonomous Linear Feedback Shift Register (ALFSR) as TPG.

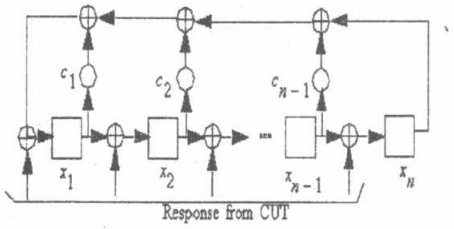

Fig. 2. Muti-Input Shift Register (MISR) as Test Response compactor

The BILBO register is one of the structures designed specifically for test-per-clock BIST schemes [1]. Fig. 3 and Fig. 4 show two alternative BIST flip-flops, referred to as BILBO cell1 and BILBO cell2, respectively [1]. These cells can be reconfigured using mode control signals B1, B2, and HOLD to perform the normal, test pattern generation (LFSR), MISR, scan, and hold modes. These control signals are generated from the BIST controller. The hold mode is used to freeze the resulting signature when a test session is finished. All cells are

*Egyptain Armed Forced

connected into a single scan chain in the scan mode either to provide access to the signature at the end of the test session or to seed the generator and the compactor (initial seed). These cells are considered the foundation of the user-defined data registers in IEEE-1149.1 standard architecture.

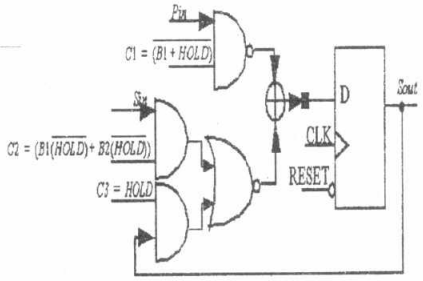

Fig. 3. BILBO cell1.

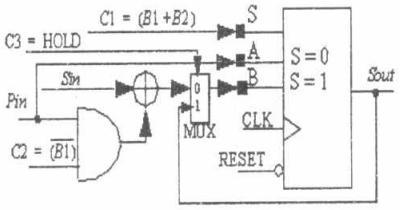

Fig. 4. BILBO cell2.

Assigned codes are chosen for B1, B2, and HOLD. They are based on the assumption that the normal mode and the scan mode operations of all BILBO registers are performed simultaneously and a BILBO register operates either in the TPG mode or the MISR mode for a particular test session. The binary values assigned to B1 and B2 need to have the same values in the normal mode (B1 = B2 = 0) and in the scan mode (B1 = B2 = 1). The binary values assigned to B1 and B2 need to have different values in the TPG mode and in the MISR mode. B1 = 1 and B2 = 0 in the TPG mode, and B1 = 0 and B2 = 1 in the MISR mode. In the hold mode, HOLD signal is high where, in the BILBO cell1, both B1 and B2 are X (don't care) and, in the BILBO cell2, either one of B1 and B2, or both of them need to be high. Every BILBO register needs one distributed decoder to decode the control signal (B1, B2, and HOLD) into other appropriate control signals for the BILBO cells. BILBO cell1 and BILBO cell2 will be compared as:

- The performance penalty in the normal mode is lower in the case of BILBO cell2 than in the case of BILBO cell1.
- BILBO cell1 is fully tested but input A of the BILBO cell2 is not tested in the BIST mode.
- The area overhead of BILBO cell1 is less than the area overhead of BILBO cell2. However, the area overhead of the distributed decoder in BILBO cell2 is less than that required for BILBO cell1 because the number of terms of the logic expression of the decoding signals are fewer in the case of BILBO cell2. In general, the area overhead of the BILBO register based on BILBO cell1 is less than the BILBO register based on BILBO cell2.
- From Fig. 3 and Fig. 4, the propagation delay for the signal in the BIST mode is less in the case of BILBO cell1 than in the case of BILBO cell2. So BILBO cell1 is faster than BILBO cell2 in the BIST mode.

Also, the *boundary scan* (BS) technique [5-6] is a *design for testability* technique. It offers a convenient alternative to physical probing by effectively migrating the test probe circuitry

into the chip. It enables a non-contact method of accessing chip pins for testing. The basic architecture of this standard is incorporated at the chip level and essentially consists of a protocol by which various test functions can be carried out [1, 5-6]. The basic concept behind boundary scan is to introduce a scan cell situated just inside the chip I/O pad ring at each I/O pin. The boundary scan cells are interconnected to form a scan chain. During normal system operation, the boundary scan cells are transparent, passing the I/O signals unaltered. However, when the test mode is entered, the logical connection from the I/O pads to the system logic is broken and signals may be injected and sampled via the BSR. Using an appropriate set of boundary scan test patterns, most of the interconnections on a circuit board can be checked for continuity and short circuits. The IEEE Standard 1149.1 defines an interface for boundary scan testing at device and assembled circuit board level. It defines four (or optionally, five) new pins, through which the boundary scan test features are controlled, forming the test access port (TAP). Two of them (test clock, TCK, and test mode select, TMS) are used to control the protocol, while the remaining two pins (test data input TDI and test data output TDO) are employed to shift data into and out of the chip serially. The optional pin (test reset input TRST*) forces the test logic into its reset state asynchronously to ensure normal system operation at power up when the test logic is not in use. The basic architecture of IEEE-1149.1 standard, incorporated at the chip level, is illustrated in Fig. 5. More details of the architecture of IEEE Standard 1149.1, are given in [1, 5-6].

Using the inherent BIST capabilities of the boundary scan architecture, defined in the IEEE-1149.1 standard, a new structure of the boundary scan architecture for BIST is introduced. This paper includes three section. The first section presents the design of the BIST boundary scan architecture. The second one presents test sequence of the BIST boundary scan. The last section is the conclusion.

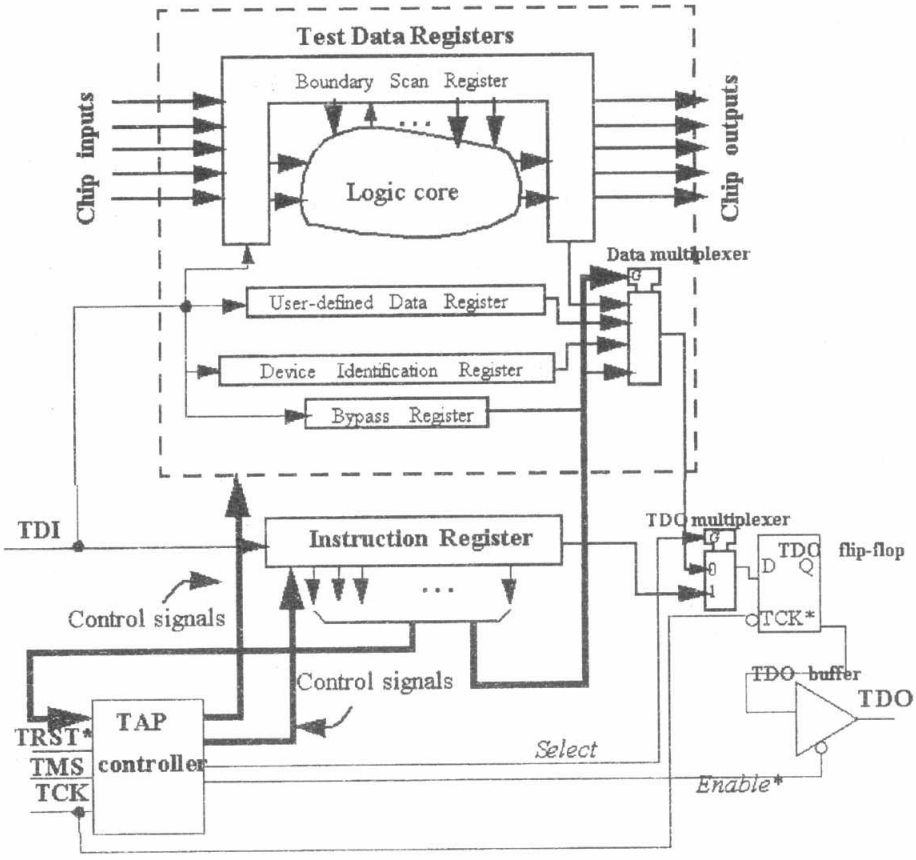

Fig. 5. Block diagram of a boundary scan of the chip (IC).

1. Design of BIST boundary scan architecture

This section relates to a BIST method for an IEEE-1149.1 boundary scan circuit. The BSR is designed to work in BIST mode. The instruction register supports new instructions to BIST operations. The TAP controller is designed to control the BIST process.

1.1 Design of BIST boundary scan register

This section presents BIST BSR input/output cells. The Update (UPD) flip-flops in the BSR input cells have been configured to operate as a TPG in the BIST mode. The Capture (CAP) flip-flops in the BSR input and output cells have been configured to operate as an MISR in the BIST mode.

Fig. 6 presents a BIST BSR input cell for an input pin. The input pin of the chip is connected to *Pin*, and *Cin* feeds the core (system) logic input. Comparing the circuit with that of the standard cell [1], two new input signals (*Cin_p, BIST_mode_I*) have been added plus one

multiplexer at the input of the UPD flip-flop. When *BIST_mode_I* is set high, the input cells are configured as part of a TPG in the BIST mode. Signal *Cin_p* is fed from signal *Cin* of the previous input cell and signal *Cin* feeds signal *Cin_p* of the next input cell. This cell requires 28.5 GE (Gate Equivalent), where a standard BSR input cell requires 26.5 GE so this cell requires more 2 GE to operate in BIST.

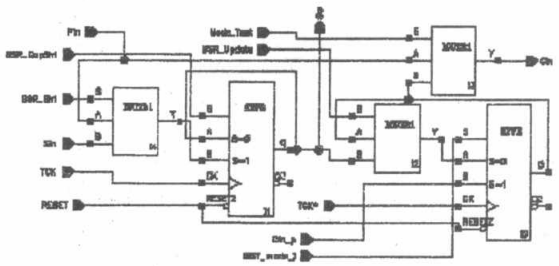

Fig. 6. Schematic diagram of the BIST BSR input cell.

Fig. 7 presents a BIST BSR output cell for a primary output pin. The output pin of the chip is connected from *Pout*, and *Cout* is fed from the core (system) logic output. Comparing the circuit with the standard cell [1], one new input signal has been added plus a multiplexer and an XOR gate at the input of the CAP flip-flop. When *BIST_mode_O* is set high, the cell is configured as part of an MISR in the BIST mode. This cell requires 32.25 GE, where a standard BSR input cell requires 26.5 GE so this cell requires more 5.75 GE to operate in BIST.

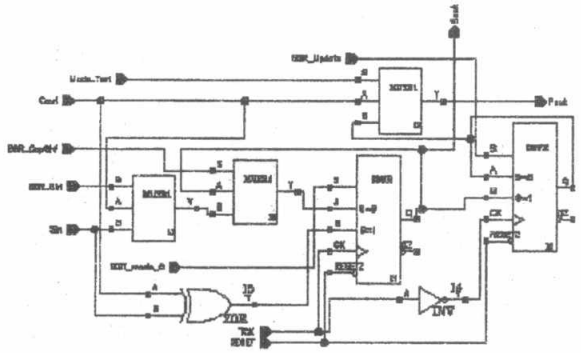

Fig. 7. Schematic diagram of the BIST BSR primary output cell.

1.2 Design of the TAPC controller

The TAPC consists of two main blocks a FSM (Finite State Machine) and, TAPC decoder. The FSM is driven by the signals TCK, TMS, and TRST*. The state transition diagram of the FSM is shown in Fig. 8. Its state diagram has sixteen (16) states. It has a single control input,

TMS, to determine the transitions between states which only occur on a *rising edge of the TCK* [5].

The inputs of the TAPC decoder are driven by four signals from the FSM and other three signals from the IR (operation field). All outputs of this decoder are latched into *D*-type flip-flops on the falling edge of TCK, except for two control signals, *IR_Update* and *BSR_Update*. These two control signals are latched on the rising edge of TCK because the UPD register in the BSR and shadow register in the IR are updated on the falling edge of TCK. The control bus, generated by TAPC and its latch flip-flops, is classified according to the selected register connected between TDI and TDO as shown in Fig. 9.

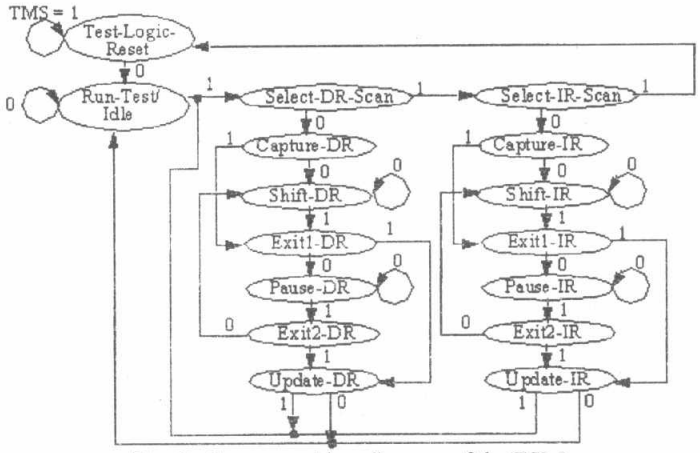

Fig. 8. State transition diagram of the FSM.

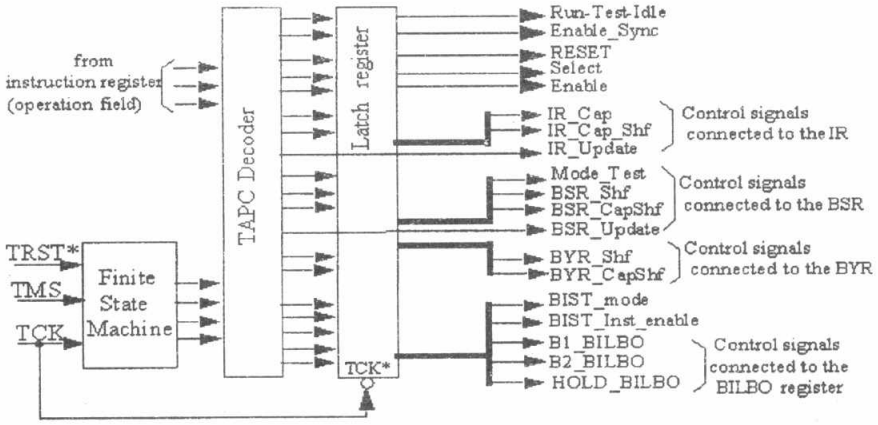

Fig. 9. Block diagram of the TAPC.

The control signals, which feed the *IR, BSR, and BYR*, have been discussed in [1]. Control signals *B1_BILBO*, *B2_BILBO*, and *HOLD_BILBO*, connected to B1, B2, and HOLD of the distributed decoder of the BILBO register (based on BILBO cell1), and control signals *BIST_mode*, and *BIST_Inst_enable* will be discussed here. The control signal *Enable_Sync* is used to switch between the clock of the chip and the clock of the boundary scan circuitry (TCK).

This paper uses seven instructions. The list of these instructions with their binary codes is presented in Table 1.

Table 1. Instruction codes

Instruction	Op-code	Operation field	Address field
SAMPLE/PRELOAD	000XX00	000	XX00
EXTEST	001XX00	001	XX00
BIST-BSR	010XX00	010	XX00
BIST-BILBO, first test session (BFT)	011XX01	011	XX01
BIST-BILBO, second test session (BST)	100XX01	100	XX01
INTEST	110XX00	110	XX00
BYPASS	111XX11	111	XX11

SAMPLE/PRELOAD instruction, EXTEST instruction, INTEST instruction, and BYPASS instruction are public instructions, discussed in [1]; the others are user-defined instructions which support the BIST process. The purpose of these instructions follows.

BIST-BSR: The instruction BIST-BSR is a pin-permission operational instruction. It places the BSR between TDI and TDO, and puts the system (core) logic inputs under the control of the UPD flip-flops of BIST BSR input cells. During shifting, the signature generated in the BIST BSR cells after finishing BIST process can be shifted out and a new initial seed can be shifted in. While this is happening, the logic values, driven to the output pins, are controlled by the BIST BSR output cells so that known safe output values are held during the BIST process. At the UPDATE-DR state, an initial seed can be applied to the UPD flip-flops. At the RUN-TEST/IDLE state, the control signal *BIST_mode* goes high on the falling edge of TCK and this signal feeds the *BIST_mode_O* input of the BIST BSR output cells. Control signal *BIST_mode* is shifted a half clock cycle to feed the *BIST_mode_I* input of the BIST BSR input cells. The shifted version of the control signal *BIST_mode* goes high on the next rising edge of TCK. The FSM stays in the RUN-TEST/ IDLE state for a specific number of clock cycles. When the FSM leaves this state, the signature generated in the BIST BSR cells needs to be shifted out at the SHIFT_DR state. The FSM goes through the CAPTURE_DR state before going to the SHIFT_DR state. The BSR does nothing in the CAPTURE_DR state to avoid the corruption of the signature (control signal *BSR_CapShf* remains low at the CAPTURE_DR state).

BIST-BILBO, first test session, referred to as BFT: The BFT instruction places the user-defined register (BILBO register) between TDI and TDO. During shifting, the signature can be shifted out and a new initial seed can be shifted in. At the UPDATE-DR state and the CAPTURE_DR state, the register is in the hold state.

The BILBO registers have been divided into two groups for two test sessions. The control signals *B1_BILBO* and *B2_BILBO* feed B1 and B2 of the BILBO register of the first group, respectively. The control signals *B1_BILBO* and *B2_BILBO* are permuted to feed B2 and B1 of the BILBO register of the second group, respectively.

In the BIST mode, the first group operates as a TPG in the first test session and as an MISR in the second test session, and the second group operates as an MISR in the first test session and as a TPG in the second test session. These groups are configured as a single shift register in the shift mode. The storage elements are assumed to be the edge-triggered D-type flip-flops.

In the BFT instruction, and at the RUN-TEST/IDLE state, the control signal *B1_BILBO* goes high and *B2_BILBO* goes low on the falling edge of TCK. The FSM stays in the RUN-TEST/IDLE state for a specific number of clock cycles. When the FSM leaves this state, the signature generated needs to be shifted out at the SHIFT_DR state.

BIST-BILBO, second test session, referred to as BST: The BST instruction places the user-defined register (BILBO register) between TDI and TDO. During shifting, the signature can be shifted out and a new initial seed can be shifted in. At the UPDATE-DR state and the CAPTURE_DR state, the register is in the hold state. In this instruction, and at the RUN-TEST/IDLE state, control signal *B1_BILBO* goes low and *B2_BILBO* goes high on the falling edge of TCK. The FSM stays in the RUN-TEST/IDLE state for a specific number of clock cycles. When the FSM leaves this state, the signature generated needs to be shifted out at the SHIFT_DR state.

The truth table of the TAPC decoder outputs is shown in Table 2. The sequence of the control signals in each column of the table is as follows: RESET, Enable, Select, Enable_Sync, IR_Cap, IR_Cap_Shf, IR_Update, BSR_CapShf, BSR_Shf, BSR_Update, BYP_Shf, BYP_CapShf, Mode_Test, BIST_mode, BIST_Inst_enable, Hold_BILBO, B1_BILBO, B2_BILBO, Run-Test-Idle.

Table 2. Truth table of the TAPC decoder outputs

State assignment of FSM	Instruction BIST-BSR	Instruction BIST-BILBO (BFT)	Instruction BIST-BILBO (BST)
0000 (Exit2-DR)	1100000000000101X000	1100000000001011110	1100000000001011110
0001 (Exit1-DR)	1100000000000101X000	1100000000001011110	1100000000001011110
0010 (Shift-DR)	1000000110000101X000	1000000000001010110	1000000000001010110
0011(Pause-DR)	1100000000000101X000	1100000000001011110	1100000000001011110
0100 (Select-IR-Scan)	1110000000000101X000	1110000000001011110	1110000000001011110
0101 (Update-DR)	1100000001001101X000	1100000000001011110	1100000000001011110
0110 (Capture-DR)	1100000000000101X000	1100000000001011110	1100000000001011110
0111 (Select-DR-Scan)	1100000000000101X000	1100000000001011110	1100000000001011110
1000 (Exit2-IR)	1110000000000101X000	1110000000001011110	1110000000001011110
1001 (Exit1-IR)	1110000000000101X000	1110000000001011110	1110000000001011110
1010 (Shift-IR)	1010010000000101X000	1010010000001011110	1010010000001011110
1011 (Pause-IR)	1110000000000101X000	1110000000001011110	1110000000001011110
1100 (Run-Test/Idle)	1110000110001111X001	1110000110001110101	1110000110001110011
1101 (Update-IR)	1110001000000101X000	1110001000001011110	1110001000001011110
1110 (Capture-IR)	1110111000000101X000	1110110000001011110	1110110000001011110
1111 (Test-Logic-Reset)	0110000000000000X000	0110000000000000X000	0110000000000000X000

2. Test sequence of the BIST boundary scan

In this section, the steps of the BIST process are described through the simple structure shown in Fig. 10. This structure has three combinational blocks C1, C2, and C3. C1 is fed from the BIST BSR input cells and feeds the BILBO register G2, C2 is fed from the BILBO register G2 and feeds the BILBO register G1. C3 is fed from the BILBO register G1 and feeds the BIST BSR output cells. This structure will be tested in two test sessions.
C1 and C3 are tested in the first test session. In order to test C1, the BIST BSR input cells are configured as a TPG and G2 is configured as an MISR. To test C3, G1 is configured as a TPG and the BIST BSR output cells are configured as an MISR. C2 is tested in the second test session with G2 configured as a TPG and G1 configured as an MISR.

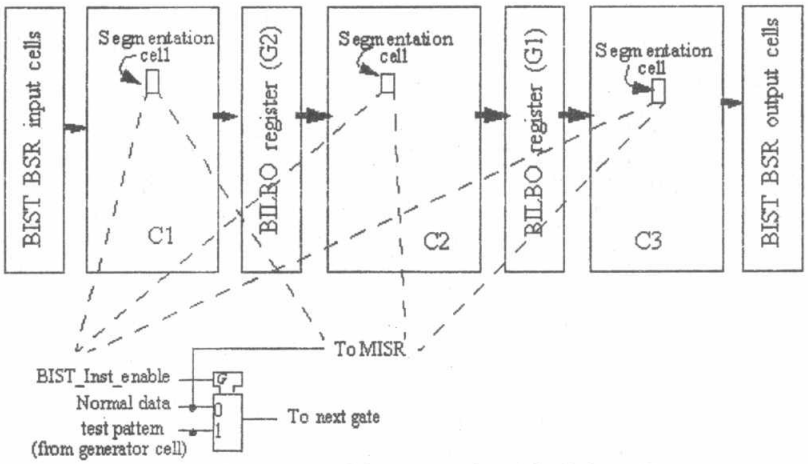

Fig. 10. Block diagram of the system (core) logic in BIST.

The control signal *BIST_Inst_enable* is active high when the instructions BIST-BSR, BFT, and BST are active. The control signal *BIST_Inst_enable* is responsible to cause the cut in the case of the pseudo-exhaustive testing [7-8] as shown in Fig. 10.

The TAPC controls the sequence of these two test sessions. The following steps will summarize the BIST process:

In the first test session:

1. Initialize the FSM of the TAPC to the TEST-LOGIC-RESET state.
2. Load the IR with the PRELOAD instruction. This connects the BSR between TDI and TDO, but does not grant pin-permission.
3. Shift the initial seed into the BIST BSR cells. While this is happening, the initial seed, driven to the BIST BSR primary output cells, controls the output pins so that known safe output values are held during the BIST process.
4. Load the IR with the BFT instruction. This connects G1 and G2 between TDI and TDO, and grants pin-permission.
5. Shift the initial seeds into G1 and G2 at the SHIFT_DR state.

6. Go to the RUN-TEST/IDLE state. In this state, control signal *B1_BILBO* goes high and *B2_BILBO* goes low to configure G2 as an MISR and G1 as a PETPG. The control signal *BIST_mode* goes high on the falling edge of TCK and this signal feeds the *BIST_mode_O* input of the BIST BSR output cells. Control signal *BIST_mode* is shifted a half clock cycle to feed the *BIST_mode_I* input of the BIST BSR input cells on the following rising edge of TCK. Stay in this state for the required number of clock cycles.
7. When the FSM leaves this state, the signature generated in G2 needs to be shifted out at the SHIFT_DR state.
8. Load the IR with the BIST-BSR instruction. This connects the BSR between TDI and TDO, and grants pin-permission.
9. The signature, generated in the BIST BSR output cells, needs to be shifted out at the SHIFT_DR state. While this is happening, the logic values, driven to the output pins, are controlled by the BIST BSR primary output cells so that known safe values are held during the BIST process.

In the second test session:

10. Load the IR with BST instruction. This connects G1 and G2 between TDI and TDO, and grants pin-permission.
11. Shift the initial seeds into G1 and G2 at the SHIFT_DR state.
12. Go to the RUN-TEST/IDLE state. In this state, control signal *B1_BILBO* goes low and *B2_BILBO* goes high to configure G1 as an MISR and G2 as a PETPG. Stay in this state for the required number of clock cycles.
13. When the FSM leaves this state, the signature generated in G1 needs to be shifted out at the SHIFT_DR state.
14. Go to the TEST-LOGIC-RESET state and halt the test.

A complete design of a boundary scan testable circuit was presented. The system (core) logic implements 8-bit parallel multiplier. The multiplier requires 1908.5 GE (gate equivalent). The BIST BSR input/output cells requires 1314 GE. The remaining BS logic requires 761 GE. Verilog simulation indicates that all cells reported in this paper operate correctly under the control of the TAPC.

The overhead of the BIST BSR input/output cells is high compared to the hardware required by the rest of the BS logic. By placing the BIST BSR input/output cells in the area between the I/O pad area and core area, the area overhead in the core area can be reduced. The full custom design for the BIST BSR input/output cells has been carried out using Cadence and the ES2 0.7μ CMOS process. In each case, the width of the cell has been restricted to fit the width of the associated I/O pad, which was taken from the ES2 library. The minimum space between the I/O pads and the core logic is approximately 185 μm. This space can be reduced in the core-limited version. More details are given in [1].

3. Conclusion

The incorporation of BIST capabilities into the boundary scan architecture, defined in the IEEE-1149.1 standard has been investigated. This paper presented new boundary scan architecture for BIST. This structure has the following capabilities:

- The UPD flip-flops of the BIST BSR input cells have been configured to form a TPG in the BIST mode. The CAP flip-flops of the BISR BSR input and output cells have been configured to form an MISR in the BIST mode. This configuration supports BIST for either the cascaded or non-cascaded BSR input and output cells.
- The TAPC controls the BIST process; in addition to the standard boundary scan process. Three instructions for BIST process have been presented to support the BIST operation for the BSR and the BILBOs (user defined registers).

The structure given in the paper proposes that the chip operates in two test sessions. In the case where the chip needs more than two test sessions, it is easy to design other user-defined instructions to deal with other test sessions.

References

[1] Mohamed H. El-Mahlawy, *Pseudo-Exhaustive Built-In Self-Test for Boundary Scan*, Ph.D. thesis, Kent University, U.K., 2000.

[2] Paul H. Bardell, Willian H. McAnney, Jacob Savir, "Built-In test for VLSI: pseudorandom techniques," John Wiley and Sons, 1987.

[3] S. W. Golomb, *Shift Register Sequences*, Laguna Hills, CA: Aegean Park Press, 1982.

[4] Mohamed H. El-Mahlawy, *Automatic Measurement of Digital Circuits*, M.Sc. thesis, Military Technical College, Egypt, 1995.

[5] Kenneth P. Parker, *The Boundary-Scan Handbook Second Edition Analog and Digital*, Kluwer Academic Publishers, 1998.

[6] TEXAS INSTRUMENTS, *IEEE Std. 1149.1 (JTAG) Testability*, 1997.

[7] Mohamed H. El-Mahlawy, Winston Waller, "An efficient algorithm to partition the combinational circuits for pseudoexhaustive testing." *IEEE 17th National Radio Science Conference*, Minufiya, Egypt, pp. C24 (1-11), Feb. 2000.

[8] Mohamed H. El-Mahlawy, Winston Waller, "An efficient algorithm to design convolved LFSR/SR." *IEEE 17th National Radio Science Conference*, Minufiya, Egypt, pp. C23 (1-10), Feb. 2000.