Embedded memories are an indispensable part of any deep submicron System on a Chip (SoC). The requirement arises not only to validate the digital logic against manufacturing defects but also do robust testing of large memory blocks post-manufacturing. MBIST (Memory built-in self-test) provides an effective solution for testing of such large memories. Verification of functioning MBIST is an essential part in any SoC design cycle, as it enables the designer to detect beforehand any issues related to MBIST. The main focus of this paper is to discuss the general issues faced, and best practices to be followed, during MBIST Verification.

MBIST is a self test logic that generates effective set of March Algorithms through inbuilt clock, data and address generator and read/write controller to detect possibly all faults that could be present inside a typical RAM cell whether it is stuck at 0/1 or slow to rise, slow to fall transition faults or coupling faults. Figure 1 shows the typical interface of memories with the collars and MBIST controller unit. A total of 68 such March algorithms are used for at speed testing of a particular type of memory.

Example of such an algorithm is March C- algorithm which does following operation to possibly detect stuck at 0/1 faults and transition faults from 0 to 1 or 1 to 0 in one or another write or read
operation.

\{ \downarrow(w0); \uparrow(r0,w1); \uparrow(r1,w0); \downarrow(r0,w1); \downarrow(r1,w0); \downarrow(r0) \}

Where r0/1, w0/1 in each March algorithm (M0-M5) describes the read 0/1 or write 0/1 operation in increasing/decreasing order in the corresponding memory array.

**Advantages of using MBIST**

- Simplification of test program.
- Design complexity is reduced as no need for managing direct access of memories from top level.
- Possible to run different algorithms on memories.
- Reduction in test costs due to test time reduction and tester resources reduction.
- At speed testing of memories and user defined algorithms could be run on any particular memory which is a challenge with ATE (Automatic test equipment).
- Repair data calculation automatically performed by BIST.
- Can be used for burn-in testing of memories.

MBIST has established itself as a successful solution for testing of memories at SoC level. But it is extremely important that it should be cautiously implemented for its reliable operation. **Figure 2** shows how MBIST is functionally implemented as a wrapper over the memories for efficient functional testing the memories.

**ISSUES FACED DURING MBIST VERIFICATION**

It is strongly recommended to use remarked best practices during verification of MBIST and keep a preciseness verification setup during Gate Level Simulation(GLS) of MBIST.

**ANNOTATION WARNINGS**

- After every compilation of the SoC design, SDF annotation warning for both WCS and BCS must be verified as it would confirm the delays information synchronization with the used netlist during the GLS of the MBIST of a particular SoC design. Otherwise, it might lead to faulty simulation and
erratic results causing the BIST to fail.

**FREQUENCY CHECKS**

- Executing MBIST Algorithm on a particular memory at a higher frequency than its maximum specified frequency may lead to X propagation in the design and will cause BIST to fail. Programming bits of control register for PLL locking and different dividers must be vigilantly implemented.

**PROPAGATION OF CLOCKS IN THE MBIST COLLAR AND MEMORY**

- MBIST collar (Wrapper around the memory) receives 2 set of clocks. One set of clock is the at speed memory clocks which equals the number of memories in the collar and other clock is the slow speed BIST TCK (BIST Controller Clock) which is controllable in Test mode. Prohibition of switching of any of these clocks will obstruct the start of BIST algorithm on the respective memories.

**TAP SELECTION**

- In order to program the internal registers of the MBIST Controller, MBIST TAP (TEST ACCESS POINT) is accessed from the MASTER TAP which are used for configuring different modes and selection of memories on which BIST Algorithms are to be run. JTAG TAP is accessed in order to program the control unit registers which are used for configuring rest of the SoC design for enabling entry into the MBIST mode. Both MBIST TAP as well as JTAG TAP can be only accessed through MASTER TAP. Proper selection ordering of different TAPs plays vital role in bringing the device into a particular configuration.

**MBIST COMPLETION PERIOD**

Time period for testing a particular set of memories is dependent upon

- Particular algorithm needed to be run on particular memory which determines the number of cycles desired for completion of MBIST.
- Size of biggest memory from the set of memories on which BIST is to be RUN.

Providing larger number of cycles for BIST completion would add to the test cost and vice versa providing lesser number of cycles for BIST completion will lead to Failure of the BIST as BEND would never be achieved.

**CONTROL UNIT RESET DEASSERTION**

- Control unit RESET should also be in deasserted state when the registers are being programmed as if the control unit is in reset it may lead to failure of programming of the control registers.
STATUS MONITORING AND EFFICIENT DEBUGGING

- Global status of the BIST depends upon individual status from the collars and each collar generates its own BREPAIRABLE and BBAD status. Whole of the BIST would fail if even a single memory cut is faulty when multiple memories are running simultaneously during the test case. Determination of that particular failing memory is achieved via inspection of the individual status of the memories at collar level or from the March end signal at the BIST controller level which indicates completion of a BIST algorithm on a particular memory.

FLASH REPAIR

- Fault Injection in the repairable memories is done through behavioral model of the particular type of SPREG or SPHD memory. Figure 3 represents the commonly adopted column redundancy approach used for repairing repairable memories. Flash repair in MBIST is the repair flow in which flash loading is done in LC scanning phase of flash through a hex code loaded in the hex file during GLS with the calculated repair data (during soft repair). Endianess of flash determines manner in which is loaded. Particular addresses and data loaded in the hex file should be correct. Otherwise, erratic data would be transferred to the memories rendering them irreparable causing the BIST to fail.

ROM TESTING

- Testing of ROMs is done entirely separately as only read operation is to take place in such type of memories. A MISR (Multiple input signature register) is generated through a triple read algorithm. This particular MISR is compared with the already generated MISR for ROM thereby concluding
the result of BIST run. ROM must be initialized accurately for its proper operation otherwise X would propagate in the SoC design during BIST run causing the BIST to fail.

CLOCK SWITCHING AND SELECTION

- Order of clock switching and selection is quite important. Clock selection should be done after switching of clock is done. This particular practice would enable the verification to avoid any particular timing issue in WCS and BCS GLS thereby preventing X propagation in the design which could be avoided with this best practice.

CLOCKING VERIFICATION

- Clocking verification must be done preceding MBIST Verification. As controllable clocks should get the same enable which has been programmed through the test cases otherwise it would lead to propagation of unwanted clocks or prohibition of required clocks in the design which might cause erratic results.

PROPAGATION OF CLOCKS

- Clock propagation should be only done when the clock is actually required in the functionality of BIST circuitry otherwise it would cause power as well as timing issues in cases where some enable signal is getting switched and due to propagation of clock it would lead propagation of X in the design.

POWER SUPPLY SEQUENCING

- GLS Ramp up of device is done through Vdd power supply switching from 0 to 1 after certain number of clock cycles (keeping Vss to 0). Failure to replicate the device behavior in order to ramp up the design will cause BIST to fail.

POWER DISSIPATION CONSIDERATION

- MBIST performs functional testing of memories so power dissipation due to memories should be identical in both functional mode as well as MBIST test mode but due to simultaneous testing of large number of memories in MBIST creates IR drop issues and also can exceed the specified power limits therefore power dissipation issues is more prominent in MBIST test mode as compared to normal functional mode. So, it is must to enable only a few numbers of memories at a time in MBIST test case after proper consultation with the Power management team.

The above mentioned challenges enumerate almost all the issues that would be faced by a DFT engineer during the verification of MBIST, and would enable them to maintain a good quality standard of verification through prior debugging of any issue related to MBIST.