Journal of Engineering and Applied Sciences 14 (8): 2695-2700, 2019
ISSN: 1816-949X

An Efficient Use of Memory Grouping Algorithm for Implementation of BIST in Self Test

[1]Sunil Kumar Ojha, [1]O.P. Singh, [1]G.R. Mishra and [2]P.R. Vaya
[1]Department of Electronics and Communication Engineering,
Amity School of Engineering and Technology (ASET),
Amity University, Lucknow Campus, Uttar Pradesh, India
[2]Indian Institute of Technology Madras (IITM), Department of Electrical Engineering,
Chennai, India

Abstract: Design for test engineers has to group memories for BIST implementation. But there are various problems involved during memory grouping such as number of memories are increasing day by day and DFT engineers spends lots of time on grouping memories for BIST structure. Also, the memories may have different frequency of operation and the physical placement may also be different and scattered. This study presents an algorithm to group the memory in an efficient way such that it reduces the effort of memory grouping for BIST structure and BIST can support the grouping. Also, it take care the power during memory BIST operation.

Key words: Design for test, memory grouping, built in self test, test time, testing memories, BIST

INTRODUCTION

As the number of memories in SoCs is increasing very rapidly, it becomes very difficult to group them for BIST structure. This grouping also takes huge amount of time if the memories are operating on different frequency and if their physical location is scattered. The power domain and memory type is also an important aspect while grouping, since, both impacts the memory grouping. Currently for an SoC on an average 40-60% area are used by the memories, so, in such case it becomes important to group them very effectively such that the area is utilized very efficiently and while operating the power dissipation is as minimum as possible. A memory grouping algorithm with various parameters involved is discussed.

MATERIALS AND METHODS

Memory grouping: The memory grouping is a critical step involved during BIST implementation, this grouping is responsible for determining number of BIST required for a particular SoC or chip and also how the BIST will operate to test memory. The general consideration for DFT engineer while performing memory grouping is that which memories cannot be grouped within a BIST: considering power domain, synchronous, frequency domain, asynchronous memory type and retention test. Some of the memories are defined as hard rules and hence cannot be grouped together but the major consideration for memory grouping is the frequency and the physical location of the memories. The physical location is important, since, it affects the routing length. The other consideration includes test time and power dissipation during memory BIST test.

Adham and Nadeau-Dostie (2004) proposed a work on a new BIST algorithm to detect defects in the bit and group write enable circuitry. Proposed algorithm is based on serially shifting data into the memory using an internally generated write enable mask which enables the detection of the failures. Ko and Huang (2017) proposed a two-stage approach to synthesize the memory BIST controllers of a 3D IC under both pre-bond testing and post-bond testing power constraints: the first stage performs memory grouping and the second stage performs test scheduling. Tehranipour *et al.* (2004) proposed a low cost BIST architecture. This architecture is an efficient method for testing of all processor and embedded SRAMs in system-on-chip. Miyazaki *et al.* (2006) formulated a memory grouping problem and algorithm developed to solve the problem. Their experiment results showed that the proposed method reduced the area of memory BIST wrappers. Zorian and Bederr (1997), proposed a new dual BIST scheme for multi-chip and single-chip BIST execution. With almost no

Corresponding Author: Sunil Kumar Ojha, Department of Electronics and Communication Engineering, Amity School of Engineering and Technology (ASET), Amity University, Lucknow Campus, Uttar Pradesh, India

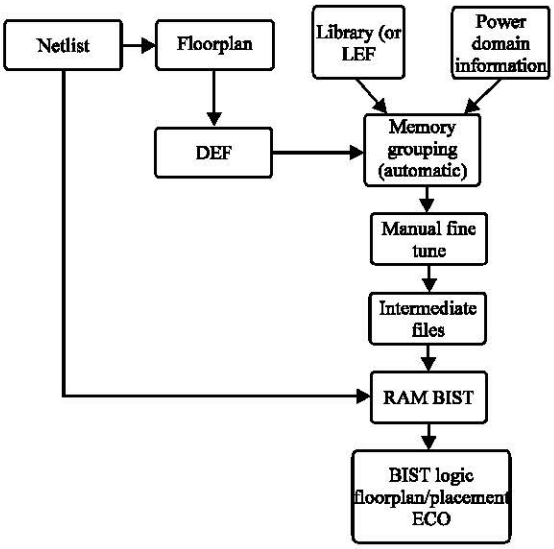

Fig. 1: Steps invoved in memory testing

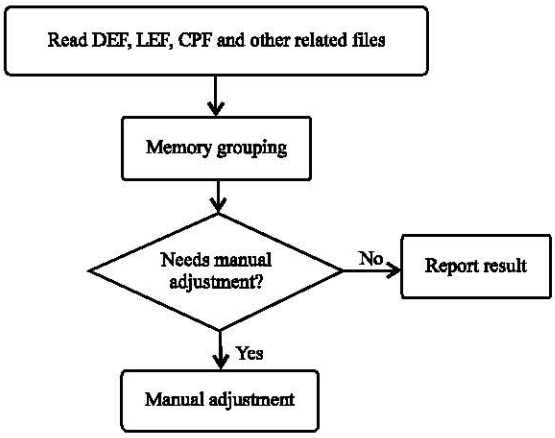

Fig. 2 : Memory grouping cycle

additional hardware cost, this research shows that the problem of performance testing of MCM can be effectively reduced. Harutyunyan *et. al.* (2011) in this research a method of symmetry measurement for memory test algorithms is introduced. The method can be applied both for March and non-March test algorithms.

By Ivaniuk (2008) here an efficient microcode based programmable memory BIST architecture is introduced. The main goal of investigation was to minimize the binary microcode of march tests. Yeh *et al.* (2016) propose an ILP approach to optimize the memory BIST design with both physical design (grouping and placement of memory BIST controllers) and test application time considered.

Memory testing methodology: Figure 1 shows the complete cycle for the memory testing and BIST

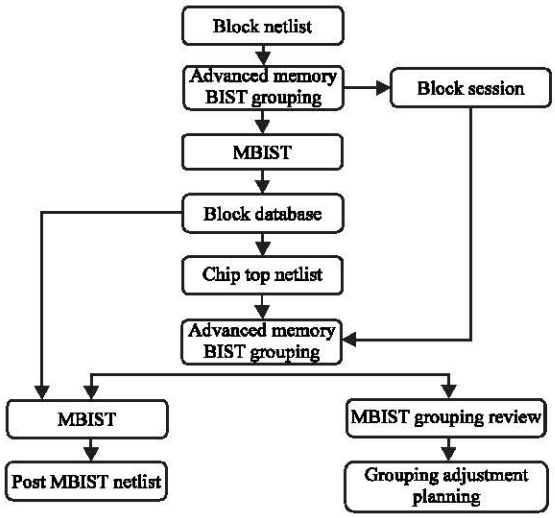

Fig. 3: Complete MBIST cycle

implementation. Te Netlist for floorplan can be different with the one for RAMBIST. But the memory parts (instance number, configuration, location) cannot be different. This method does not propose a new floorplan. floorplan should be the overall concern, not just RAMBIST part. This method try to find the optimization group based on the floorplan. Whenever memory parts changes, the RAMBIST group may need to optimize again. Figure 2 shows the fundamental of memory grouping. The following details are required for thememory grouping and BIST implementation:

- Needs memory library information.
- Power trend table: show memory power during BIST test
- Test time information
- Hierarchical: user can read BIST's test time and power in blocks
- Critical test interval and memory with maximum power consumption report
- Report memory with the largest power of each BIST
- Report under tested memories has highest power consumption in every BIST
- Report switching activitysetting on memories for toggle mode)
- User can use UPF file with power domain information

Figure 3 shows the complete test cycle for memory BIST for chip top and block level, it also shows the integration steps for block and top level.

RESULTS AND DISCUSSION

Memory grouping algorithm: For the algorithm to implement few environment setting files needs to be updated such as user specified constraints which includes setting memory for retention test and setting memories with redundancy but not to do memory repair. Classification for chip select and port isolation is another important factor to be taken care. Memory repair classification algorithm includes separate memories with redundancy and without redundancy. Retention test classification algorithm includes improve test time based on current grouping result. Apart from these two major bugs needs to be fixed if they occurred one of them is related to potential core dump problem and the other one is user specified hard group when running multiple times grouping. Below is the major requirement for the memory grouping:

- List of memories and BIST
- Total number of memories and BIST
- Frequency of operation
- Physical placement of memories
- Memory cell details

Algorithm 1: Memory grouping algorithm:

```
######## MEMORY_GROUPING ########
# Feature: Memory_Grouping
# File_Name: <mem_rgp_env>
# Author: <xyz>
# Version: <xxxx.xx>
# Date: <DD-MM-YYYY>
start:
## INPUT_Files ##
set df "*.def" --> input def files
set lf "*.lef" --> input lef files
set cf "*.cpf" --> input cpf files
set mif "*.info" --> input info files
set lfl "*.list" --> input lib files
## PARAMETER_SETTINGS ##
set rt "t|f" --> set retention test (true or false)
set mr "t|f" --> set memory repair (true or false)
## USER_SPECIFIED_CONSTRAINTS ##
load_session b|t --> for block or for top
set rt b|t
set rt -i --> instance
{
i_0
i_1
.
.
i_n
}
set mr -i
{
i_0
i_1
.
i_n
}
set gh -i --> hard_group
{
i_0
i_1
.
i_n
}
set gs -i --> soft_group
{
i_0
i_1
.
i_n
}
set cl -i --> class
{
i_0
i_1
.
i_n
}
cg = l --> create group according to location
cg = f --> create group according to frequency
cg = t --> create group according to type
cg = p --> create group according to power domain
end
```

Database requirements: To efficiently use the memory algorithm below are the major database requirements:

Mem_info file: Mem_info file should contain clock period to define the frequency domain specifications. If considering clock source use "clock_period @ clock_source "ex:10 @fclk_1 Power domain specification is also defined in mem_info file.

If a memory does not want to be grouped into BIST, specify the BIST name "NO_BIST ". A variable "exclude" of memory used to check "NO_BIST "or not. If a memory is not grouped into BIST, it will be specified "Ungroup ".

Environment setting files or information:

- Input file setting
- Parameter settings
- User specified constrains

Additional information

- Load_session: load session of other blocks to review block 's test time and power
- Create_group_hard: create a group not allowed other memories grouped together
- Create_group_soft: create a group allowed other memories grouped together
- Create_class: create a class not mixed with other memories
- Set_retention_test: specify memory for retention test.
- Disable_memory_repair: Specify memories with redundancy but not to do memory repair

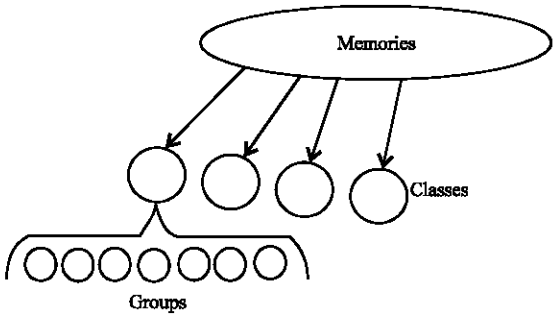

Fig. 4: Division of memories to classes and groups

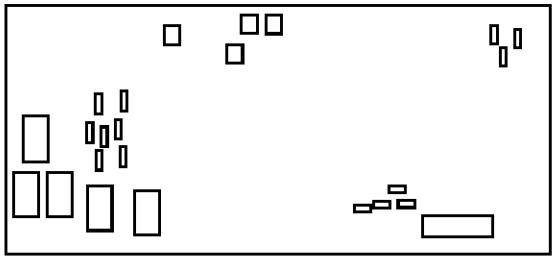

Fig. 5: Physical location of memory without application of grouping algorithm

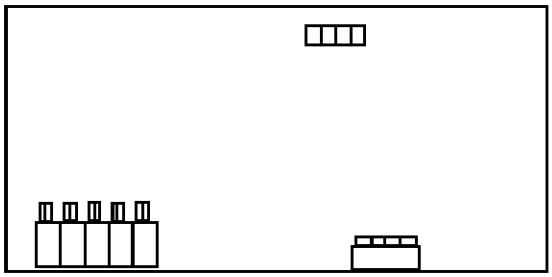

Fig. 6: Physical location of memory after application of grouping algorithm

Figure 1 show how memories are divided into classes and then a particular class is divided into groups. Here, the number of memories are divided into four class and each class is contains a group of similar memory w.r.t. frequency and location.

Figure 4 and 5 shows the physical location of memories without the application of algorithm, here it can be easily seen that the memories are scattered and placed randomly which makes it very complex and difficult to decide the BIST architecture.

Now once we apply the grouping algorithm the physical location can be made systematic and the BIST structure can easy be implemented with proper grouping of memories. Figure 6 shows the scenarios when the grouping algorithm is performed on the above case.

Table 1: Memory BIST

Memory	BIST
Mem_1	MBIST_1
Mem_2	MBIST_2
...	...
...	...
Mem_n	MBIST_n

Table 2: Test time optimization

Module (Top)	Test time (sec)	Power (W)
MBIST_1	2.2282e+00	1.0629e-01
MBIST_2	4.1635e-01	1.9071e-03
MBIST_3	5.4472e+00	2.6528e-02
MBIST_4	3.6922e-01	2.7611e-03
MBIST_5	3.4158e-01	3.7721e-03
MBIST_6	2.7225e-01	3.9537e-03
MBIST_7	8.2548e-01	4.0281e-02

Data preparation

DEF file:

- Contains design-specific information at any point of layout process
- Uncompressed DEF file only with memory placement information (routing can be excluded)

Memory LEF file list: Contains library information specify memory LEF file path in a list file.

CPF [optional]: UPF is also allowed.

Library file list: Memory library contains port function and memory repair information.

Mem_info file: Contains existed BIST grouping and test strategy.

More info: Memory leakage power and internal power consumption.

Determine grouping result: Determine the number of BIST groups based on the BIST_Size_Number. BIST_Size_Number means memory number in a group of reference. Actual grouping result may have more number or fewer memories in a group. If memories in the area are large, the report will be BISTs with less memory. If memories in the area are small, the report will be BISTs with more memory. The test time optimization is only for memory in sequential. The grouping is only for memories in sequential (Table 1-3).

Algorithm 2: Test time optimization:

BIST: MBIST_1
Port: single_Port|Double_Port
Sync: Synchronous
PD domain: xxx
Number of Memories: <xx>
Test Time: <yy.yyyy>

Table 3: Power consumption (W) during BIST test (No. retention)

Module/ BIST	1.390e-01~ 1.393e-01	1.393e-01~ 2.790e-01	2.790e-01~ 4.170e-01
Whole chip			
Total	0.2126	0.2126	0.2126
Top			
MBIST_1	0.0019	0.0033	0.0019
MBIST_2	0.0027	0.0072	0.0062
MBIST_3	0.0031	0.0019	0.0043
MBIST_4	0.0047	0.0022	0.0027
MBIST_5	0.0300	0.0085	0.0016
MBIST_6	0.0310	0.0299	0.0194
MBIST_7	0.0402	0.0317	0.0328
Block_1			
MBIST_1	0.0016	0.0013	0.0025
MBIST_2	0.0024	0.0045	0.0017
MBIST_3	0.0029	0.0071	0.0031
MBIST_4	0.0036	0.0028	0.0052
Block_2			
MBIST_1	0.0207	0.0182	0.0231
MBIST_2	0.0297	0.0321	0.0274
MBIST_3	0.0488	0.0269	0.0316

Constraints settings:

- Power domain
- Frequency domain
- Sync/Async type
- RAM/ROM type
- Port isolation
- Chip delect
- Redundancy
- Retention 5

Power table: Power table listed power consumption in two ways:

- Maximum memory power consumption
- Memory power consumption during BIST test time line

Simulated BIST time line may not be accurate because:

- Few cycle delay from previous memory test to the next memory.
- BISTs in Block and BISTs in top may not be activated at the same time. The delay time depends on TCK period
- Power value in power table only includes memory power
- Power consumption on BIST and function circuit is not included
- Use PTPX to get real power consumption

Algorithm 3: Cycle delay algorithm:

**Instance Report: (Memory Detail Parameter Information)
<Name>2048x35_instance_0
Design: hd_dpsram_2048×35
PD domain: <xxxx>
Clock: 100MHz @ CLK_1
XMIN: 3167.2900
XMAX: 3461.5700
YMIN: 4361.1700
YMAX: 4899.2761
Address: 2048
Width: 35
Read/Write power: 1.5307e-03
Standby power: 8.5081e-04
Deselect power: 8.5206e-04
Port: dual port
Chip enable: true
Redundancy: false
Retention: true
**(Memory port information)
Unknown type: WTSEL
Clock: CLK_A
Rise: 5.283e-02
Fall: 7.519e-02
Chip_Enable: CENA
Write_Enable: WENA
DATA_IN: DYA
DATA_OUT: QA
Address: AA
Errors: Cell numbers have different power domain within the group
**Toggle unit means transition/cycle. For PTPX toggle rate setting user need to use value (Toggl unit)/(Clock period)
**Clock's toggle rate is from constraint file's clock definition
** Static Probability is specified for cases if the signal is low/high active. Ex: TSMC memory chip enable is low active
For memory BIST grouping
Provide power information for reference
Provide hierarchical flow to allow user review
BIST plan easier
Provide new classification method, memory
Repair (To have better area overhead) and
Retention test (To have better test time)
Provide PTPX memory toggle rate setting file for
PTPX toggle mode

CONCLUSION

In the area of memory built in self test to optimize the memory placement and at the same time lowering the power requirements is a challenging task. This paper presented the way by which the memories can be grouped together with respect to many parameters in such a way that the area and power requirements can be met as per the requirements. The results show the low power requirements as per the grouping of memories with respect to particular grouping and the area is also optimized.

REFERENCES

Adham, S. and B. Nadeau-Dostie, 2004. A BIST algorithm for bit/group write enable faults in SRAMs. Proceedings of the 2004 International Workshop on Memory Technology, Design and Testing, August 10-10, 2004, IEEE, San Jose, California, USA., pp: 98-101.

Harutyunyan, G., A. Hakhumyan, S. Shoukourian, V.A. Vardanian and Y. Zorian, 2011. Symmetry measure for memory test and its application in bist optimization. J. Electron. Test., 27: 753-766.

Ivaniuk, A.A., 2008. Optimal memory tests coding for programmable BIST architecture. J. Electron. Test., 1: 32-37.

Ko, Y.C. and S.H. Huang, 2017. 3D IC memory BIST controller allocation for test time minimization under power constraints. Proceedings of the 2017 IEEE 26th International Symposium on Asian Test (ATS), Novomber 27-30, 2017, IEEE, Taipei, Taiwan, ISBN:978-1-5386-3516-2, pp: 260-265.

Miyazaki, M., T. Yoneda and H. Fujiwara, 2006. A memory grouping method for sharing memory BIST logic. Proceedings of the 2006 International Conference on Asia and South Pacific Design Automation, January 24-27, 2006, IEEE, Piscataway, New Jersey, USA., ISBN: 0-7803-9451-8, pp: 671-676.

Tehranipour, M.H., S.M. Fakhraie, Z. Navabi and M.R. Movahedin, 2004. A low-cost at-speed bist architecture for embedded processor and sram cores. J. Electron. Test., 20: 155-168.

Yeh, C.H., C.H. Cheng and S.H. Huang, 2016. Grouping and placement of memory BIST controllers for test application time minimization. Proceedings of the 2016 5th International Symposium on Next-Generation Electronics (ISNE), May 4-6, 2016, IEEE, Hsinchu, Taiwan, pp: 1-2.

Zorian, Y. and H. Bederr, 1997. An effective multi-chip BIST scheme. J. Electron. Test., 10: 87-95.