A Novel Architecture Design of Address Generators for BIST Algorithms

Pabba Pavani, Gaddam Anitha, Jetti Bhavana, J. Praneet Raj

Abstract— Memory Built In Self-Test has become a standard Industrial practise. It is used to determine the faults in an Embedded System. Memory BIST Address generators are used to implement addresses with low significant area, less power and high speed based on a set of Multiplexers and Counters. It shows the contribution of the area overhead with low power when compared to BIST's. This paper analyses and proposed a new architecture with more advantageous properties and implementation aspects of several Address Generators by using a single design. The design is implemented using Xilinx ISE and simulated using Xilinx simulation tool.

Index Terms— Memory Built In Self Test, Address Generators.

1 INTRODUCTION

Memory is one of the basic components in a system on chip. Memory Built In Self Test (MBIST) has become a standard industrial practice[1],[2],[5],[6]. The system on chip contains different types of memory like RAM, ROM, SRAM, and DRAM. In such chips RAM represents 2/3rd of the total number of transistors in use and 1/3rd of the chip area [3]. Hence in an embedded system 90% of the area is occupied by this memory itself. The dominant use of Embedded memory cores along with emerging new architecture designs provide low cost test with the decrease in area of memory in the system.
Therefore Memory Built In Self-Test is one of the memory testing mechanism and is cost effective for the reasons such as no external test required, Volume of test data, Realistic test time with high controllability and observability using test pattern generator on a chip.

2 ADDRESS GENERATORS

Address generator is a MBIST component. In order to detect speed related faults address generators are used to generate the address sequences in the test vectors. The main issue is its complexity that it requires large area and limits the Built In Self Test speed. For memory testing, the address generator has to generate different Counting Methods (CMs) since each counting method has its own detection.

The main purpose of the address generator is to generate the test patterns. Such address generators are Up Down linear, Up only Linear, Up only Address Complement, Binary Up/down Counter and Gray Code counter. The redesign of Memory Built In Self Test where 75% of the area is saved using a set of Multiplexers, Counters, Flipflops and Address Generators.

This results in significant savings in area and power, and allows for a higher speed Memory Built In Self Test engine and a very systematic implementation.

This paper contributes to the area of MBIST implementation by emphasizing its most critical part: Adress Generator.The main contribution consists of an implementation analysis of address generators to support a variety of address sequences such as Linear, Address Complement, Gray code etc.

ADDRESS COUNTING METHODS:

TABLE 1
TABLE SHOWING THE ADDRESS SEQUENCES

step	Li	Ac	Gc
0	0000	0000	0000
1	0001	1111	0001
2	0010	0001	0011
3	0011	1110	0010
4	0100	0010	0110
5	0101	1101	0111
6	0110	0011	0101
7	0111	1100	0100
8	1000	0100	1100
9	1001	1011	1101
10	1010	0101	1111
11	1011	1010	1110
12	1100	0110	1010
13	1101	1001	1011
14	1110	0111	1001
15	1111	1000	1000

Note: Li: linear, Ac: Address Complement, Gc: Gray code

GRAY CODE counting method has a one bit difference between the successive values (i.e., they have a hamming distance of 1). It is mainly used to reduce the noise, power and to minimize the delay [4].

- *Pabba Pavani, Gaddam Anita, Jetti Bhavana are currently pursuing Undergraduation in Electronics and Communication Engineering in Mahatma Gandhi Institute of Technology, Hyderabad, India.*
- *J. Praneet Raj is currently working as Assistant Professor in Electronics and Communication Engineering in Mahatma Gandhi Institute of Technology, Hyderabad, India.*

ADDRESS COMPLEMENT counting method represents the address sequence as: 0000, 1111, 0001, 1110, …..etc. Here, the even steps form a linear sequence whereas the odd steps are obtained by taking the complement of the preceding even steps. This counting method causes extreme delay, increase the noise and large power rise because all n or n-1 address bits change upon an address transition. It is generally used for speed related faults [4].

LINEAR counting method represents the address sequence as: 0000, 0001, 0010, 0011, ……. etc. when used as an up counter and 2^N -1, ….., 0011, 0010, 0001, 0000 when used as a down counter. It is used for the detecting the coupling and single cell faults [4].

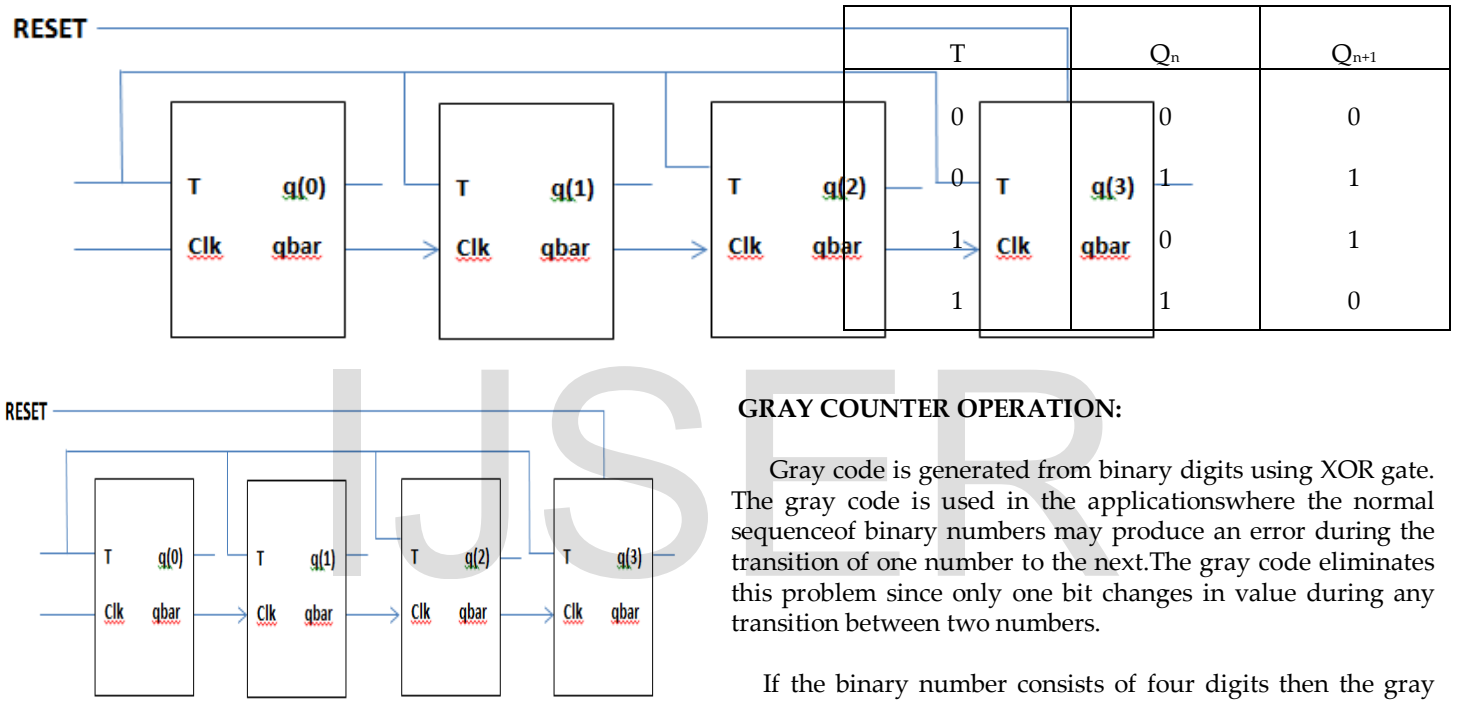


Fig1.**ASYNCHRONOUS UP COUNTER**

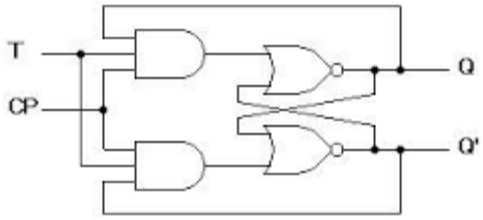
T-FLIPFLOP:

T or Toggle flipflop is a single input device and two outputs. According to the truth table,when T=0 the present state Q_n and the next state Q_{n+1} remains same.When T=1,the next state output will not remain the same as present state output such that the input T will change the next state to the inverse of present state.

This is a much simpler version of the J-K flip flop. Both the J and K inputs are connected together and thus are also called a single input J-K flip flop. When clock pulse is given to the flip flop, the output begins to toggle.

It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J-K flip-flop by tying both of its inputs high.

CIRCUIT DIAGRAM OF T-FLIPFLOP :



T- flipflop truth table:

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

GRAY COUNTER OPERATION:

Gray code is generated from binary digits using XOR gate. The gray code is used in the applicationswhere the normal sequenceof binary numbers may produce an error during the transition of one number to the next.The gray code eliminates this problem since only one bit changes in value during any transition between two numbers.

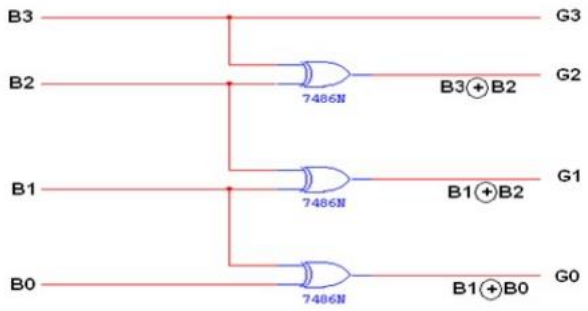
If the binary number consists of four digits then the gray code is generated as the first most significant digit is remained constant.The second binary digit is undergone XOR operation with MSB bit and stored as the second digit of the gray code.Similarly the third digit is undergone with XOR operation with the preceding digit.

XOR GATE:

XOR gate or Exclusive OR gate is a logic gate where if any one of the input is one or high then gate shows the output as high or 1.If both the inputs are 1's or 0's then the output of XOR gate is 0.

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Binary to gray code conversion
G0 = B0 ⊕ B1
G1 = B1 ⊕ B2
G2 = B2 ⊕ B3
G3 = B3



Logic diagram for binary to Gray code converter

4 PROPOSED ARCHITECTURE

UP-DOWN LINEAR ADDRESS GENERATOR:

To form an up/down counter the control input (up/down) is used to control whether the normal flip-flop outputs or the inverted flip-flop outputs are fed to the j and k inputs of the following flip-flops. Up/down counter will count from 0000 up to do

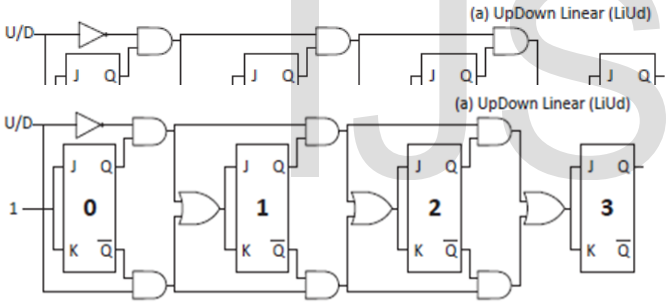


Fig2.Up down linear address generator

UP-ONLY LINEAR ADDRESS GENERATOR:

Up-only address generator is designed with the help of linear up-down address generator and multiplexers. The U/D signal acts as the control input to the all the four multiplexers. Based on this control input, the mux selects either Q or Qbar as the output.

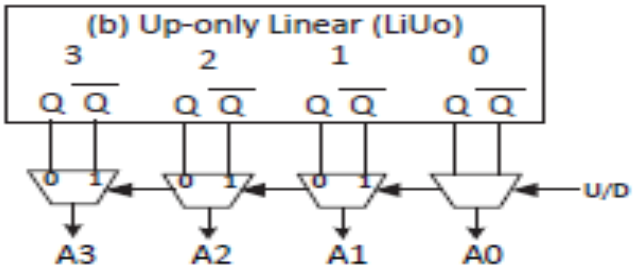


Fig3.Up only linear address generator

UP-ONLY ADDRESS COMPLEMENT ADDRESS GENERATOR:

Address Complement address generator is implemented from the up-down linear address generator as follows: The control signal U/D controls the MSB bit A3.The least significant bit of the linear address generator is given as control input to the mux of the A0, A1 and A2 i.e.,A0 = Q1 XOR Q0, A1 = Q2 XOR Q0 and A2 = Q3 XOR Q0.

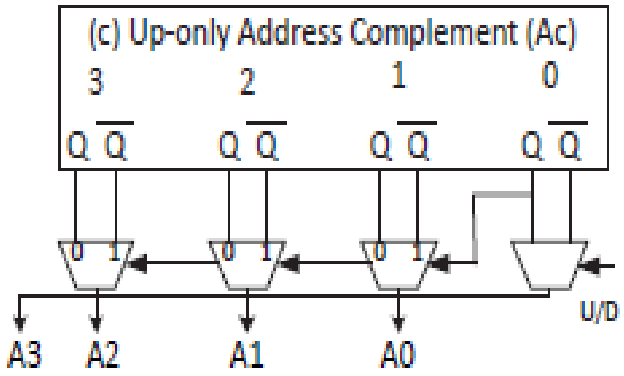


Fig4.Up only address complement address generator

GRAY CODE ADDRESS GENERATOR:

Gray code address generator is derived from the up-down linear address generator as follows: the mux of A0 is controlled by Q1 i.e., A0<= Q0 XOR Q1. In the same way A2<= Q1 XOR Q2 and A2<= Q2 XOR Q3. The mux of A3 is controlled by U/D signal i.e., A3<=Q3 which indicates that 0 input will select Q3 so as to produce the output A3.

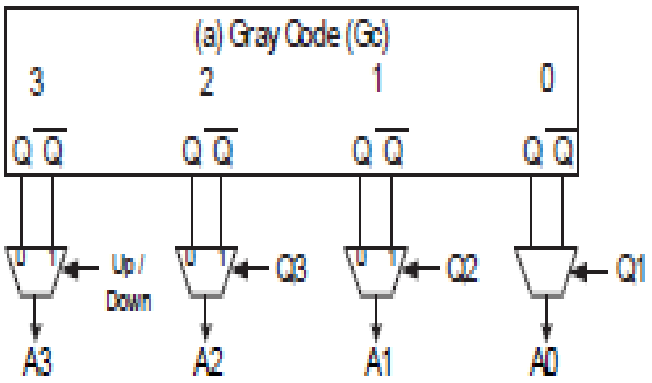


Fig5.Gray code address generator

IMPLEMENTATION OF DIFFERENT ADDRESS GENERA-

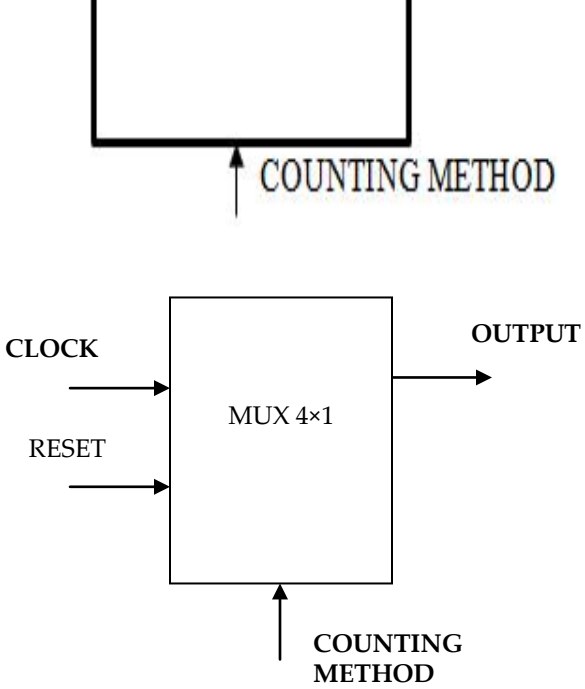


TABLE 2
COUNTING METHODS SELECTION

COUNTING METHODS	ADDRESS GENERATORS
00	LINEAR UP/DOWN COUNTER
01	LINEAR UPONLY COUNTER
10	UP ONLY ADDRESS COMPLEMENT
11	GRAY CODE

In the above circuit based on the value of control signal Counting method, the output of the mux network gives any of the four address sequences such as address complement, linear up-only, linear up-down and gray code. If counting method = 00, then the output of the mux network is linear up-down sequence. If counting method=01, then the output is linear up-only address sequence. If counting method=10, then it is address complement counting method. If counting method=11, then output is gray code sequence.

5 SIMULATED RESULTS

Implementing the Proposed architecture using Hardware description language in VHDL using Xilinx ISE and simulating it using the Xilinx Simulation tool resulted in the following waveforms. Each of these simulated results is explained below.

In all the below figures, when reset=0 the output is displayed as 0. In the fig 6, when reset=1 and input =0, the output is shown as 15,14,13,12,11,…0 (down counter) if input =1 after 100 ns (based on the time period mentioned in the test bench)) it starts to act as the up counter. In the fig 7, it gives the output as up only linear counter (0,1,2,3,…15) when reset=1, input=1. In fig 8,if reset=1 the output is displayed as 0,15,1,14,13,3,……,8 (address complement).In fig 9, the output is presented as 0,1,3,2,6,7,….(gray code counter) when reset=1. In fig10 [a] and fig10 [b], when reset=1 it gives the output based on the value of the counting method i.e., when cm=00 it displays linear up down counter, cm=01 output is linear up only counter, cm=10 gives the output as address complement counter and cm=11 provides gray code counter as output.

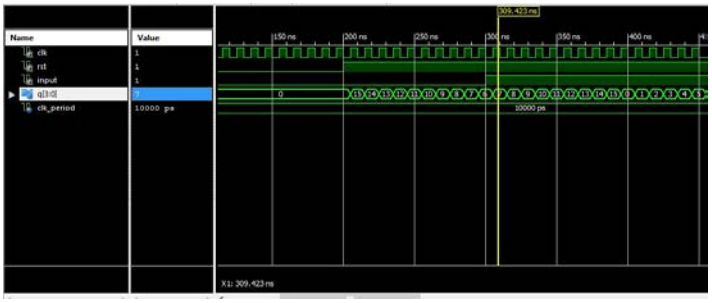


Fig6. Simulated waveform showing Asynchronous up/down counter output

Fig7. Simulated waveform showing Asynchronous up only counter output

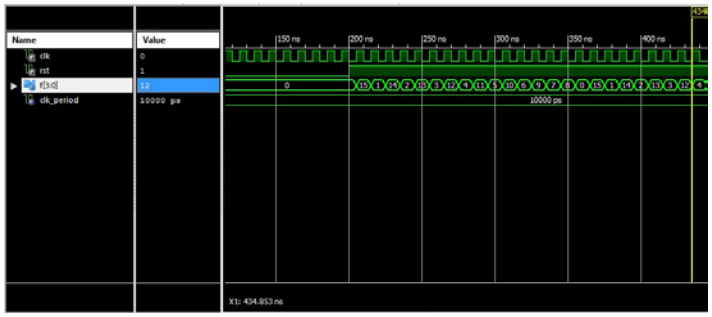


Fig8. Simulated waveform showing Address complement output

Fig9. Simulated waveform showing Gray counter output

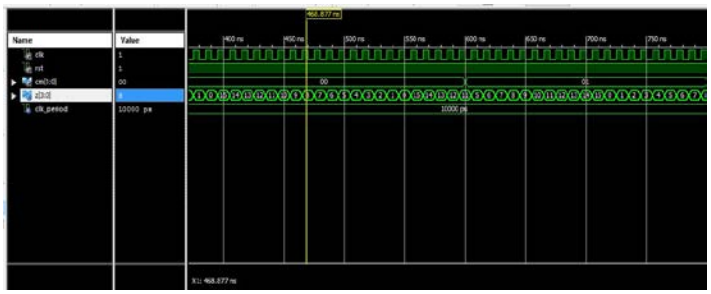


Fig10 [a]. Simulated waveform showing different address generators for counting methods 00 and 01

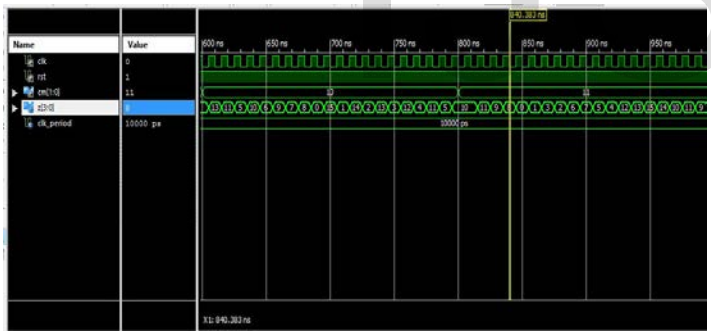


Fig10 [b]. Simulated waveform showing different address generators for counting methods 10 and 11.

By observing the below table, we can say that the parameters like Registers, Look up tables,etc. which are required in the new design are comparatively less than the old design.

TABLE 3
COMPARISION OF OLD DESIGN WITH NEW DESIGN

DESIGN SUMMARY	OLD DESIGN	NEW DESIGN
LOGIC UTILIZATION	USED	USED
Number of Slice registers	16	9
Number of Slice LUTs	28	10
Number of fully used LUT-FF pairs	0	8
Number of bonded IOBs	27	8
Number of BUFG/BUFGCTRLs	4	1

6 CON-CLUSION

This paper analyzes the implementation of address generator alternative for Memory Built In Self Test. This is due to the fact that the address generator occupies about 30% of the Memory Built In Self Test (MBIST) area. The different counting methods that are used for detecting faults are linear, address complement and gray code. The results show that the implementation of these counting methods in one address generator which is done with the help of multiplexer occupies less area overhead which in turn is more efficient in terms of Area of the MBIST and also consumes less power. These designs can be enhanced in future to still improve the results.

7 REFERENCES

[1]. R.Aitken. et .al. "A Modular Wrapper Enabling High Speed BIST and Repair for Small Wide Memories", Proc. Of Int. Test Conference,pp. 997-1005, 2004.
[2]. H. Kukner, "Generic and Orthogonal March Element based Memory BIST Engine", Master Thesis, CE-MS-2010-01, Delft University of Technology, September 2010.
[3]. Allen C. Cheng,"Comprehensive Study on Designing Memory BIST Algorithms, Implementations and Trade-offs", MI 48109-2122, The University of Michigan,Ann Arbor, December 2002.
[4]. Ad J. van de Goor, Halil Kukner, Said Hamdioui,"Optimizing memory Built In Self Test Address Generator Implementation", 2011.
[5]. X. Du, N. Mukherjee, W.T Cheng, S.M. Reddy,"A Field Programmable Memory BIST Architecture Supporting Algorithms ans Multiple Nested Loop", proc. Of the Asian Test Symposium, paper 45.3, 2006.
[6]. X. Du, N. Mukherjee, W.T Cheng, S.M. Reddy, "Full-Speed Field Programmable Memory BIST Architecture", proc. Of Int. Test Conference, pp. 1173-1182, 2005.