ISSN 2229-5518

ANALYSIS OF ADDER USING BIST

Vishwas Taneja
Surendera Group of Institutions, Sri Ganganagar
Email id: vishwastaneja@rediffmail.com

***Abstract*—Embedded memories consume an increasing portion of the die area in deep sub- micron Systems On Chip (SOCs). Manufacturing test of embedded memories is an essential step in the SOC production that screens out the defective chips and accelerates the transition from the yield learning phase to the volume production phase of a new manufacturing technology. Built In Self Test (BIST) is establishing itself as an enabling technology that can effectively tackle the SOC test problem. However, unless consciously implemented, its main limitations lie in elevated power dissipation and area overhead, and potential performance penalty and increased testing time, all of which directly influence the cost and quality of manufacturing test. This thesis introduces study of different adder and their implementation in BIST and also includes two new embedded memory BIST architectures, whose objective is to reduce the cost of test and increase the test quality to improve product reliability and yield..**

A distributed memory BIST approach with a serial interconnect scheme is first developed. This solution can concurrently support multiple memory test algorithms for heterogeneous memories with low power dissipation during test and with relatively low gate and routing area overhead, in addition to facilitating self-diagnosis. The distribute d BIST approach is then extended to a hardware/software co-testing memory BIST architecture for complex SOCs . By reusing the existing on-chip re- sources (e.g., processor cores and busses), further savings in area overhead can be achieved and performance penalty for bus-connected memories can be eliminated. This is accomplished using a design space exploration framework based on a new test scheduling algorithm that balances the usage of the existing on-chip resources and dedicated design for test (DFT) hardware such that the functional power constraints are not exceeded during test, while trading-off the testing time against the DFT area.

Then implementation of different adder in BIST architecture. Then after using synthesis tool of Synopsis, impact on critical path timing and slack have been studied so basically trying to make Memory BIST faster and efficient.*)*

Keywords:* *BIST , SOC , DFT and Embeded etc .

I. INTRODUCTION

Due to the rapid progress in the very large scale integrated (VLSI) technology, an increasing number of transistors can be fabricated onto a single silicon die. For ex- ample, a state-of-the-art 130 nm complementary metal-oxide semiconductor (CMOS) process technology can have up to eight metal layers, poly gate lengths as small as

80nm and silicon densities of 200K-300K gates/mm2. However, although million- gates integrated circuits (ICs) can be manufactured, the increased chip complexity requires robust and sophisticatedtest methods. Hence, manufacturing test is becom- ing an enabling technology that can improve the declining manufacturing yield, as well as control the production cost, which is on the rise due to the escalating volume of test data and testing times. Therefore reducing the cost of manufacturing test, while improving the test quality required to achieve higher product reliability and manufacturing yield, has already been established as a key task in VLSI design. With the ever increasing complexity and gate counts of modern devices, a number of testability problems have been encountered. One of the fundamental issues is the complexity and size of the test program required to test these devices. The test program is a necessary way to ensure that the high quality standards demanded by the market are met. In brief, two factors are in favour of smaller test programs. Firstly the smaller the program the fewer the number of test vectors and therefore the faster it can be run. Thus, test time can be reduced.

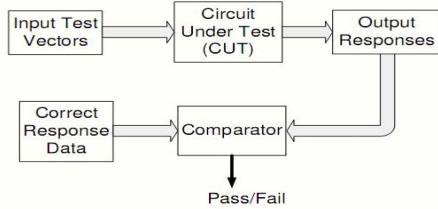
Secondly, if the program is small, problems with the available memory on testing machines can be avoided. One of the main factors which greatly increases the number of test vectors required to test a device is the use of large embedded memories. Hence, if these memories can test themselves a great reduction in test program size can be achieved.

A BIST block is an off-line verification of the Circuit Under Test (it implements the algorithm to test it), as opposed to an on-line verification or concurrent test. To be sure of shipping good quality products (memories) to the customers, it is key to meet ST corporate test standards. The solution providing this testing capability should be the most time and test efficient, capable of running the algorithms AT- SPEED and tuning them to the memory type on a given technology while impacting least on area and memory performance. Looking at the challenges the current tech- nologies are offering, the solution should also provide debug, diagnostic and repair feature along with providing simpli?ed ways of communicating with ATE (by being less demanding in terms of pins and test program).

Memory BIST system explained in the current product document satisties many more of such requirements.

II. ATE VS BIST

Circuit under test (CUT) can be the entire chip or only a part of the chip (e.g., a memory core or a logic block). Input test vectors are binary patterns applied to the inputs of the CUT and the associated output responses are the values observed on the outputs of the CUT. Using a comparator output responses are checked against the expected correct response data, which is obtained through simulation prior to design tape-out.



If all the output responses match the correct response data, the CUT has passed the test and it is labeled as fault-free. Based on the techniques how the test vectors are applied to the CUT and how the output responses are compared, there are two main directions to test electronic circuits: external testing using automatic test equipment (ATE) and internal testing using built-in self-test (BIST). When external testing is employed, the input test vectors and correct response data are stored in the ATE memory. Input test vectors are generated using ATPG tools, while correct response data is obtained through circuit simulation. For external testing, the comparison is carried out on the tester. Although the ATE-based test methodology has been dominant in the past, as transistor to pin ratio and circuit operating frequencies continue to increase, there is a growing gap between the ATE capabilities and circuit test requirements (especially in terms of speed and volume of test data). ATE limitations make BIST technology an attractive alternative to external test for complex chips. BIST is a design-for-test (DFT) method where part of the circuit is used to test the circuit itself (i.e., test vectors are generated and test responses are analyzed on-chip). BIST needs only an inexpensive tester to initialize BIST circuitry and inspect the final results (pass/fail and status bits).

A. Advantages of using memory BIST

Reduces the design complexity for managing a direct access to the memories from the design top level, with many test modes (as opposed to not using the BIST)

a. Test cost reduction due to test time reduction and tester resources reduction.

b. Simpli_cation of the test program.

c. Possibility of at-speed test.

d. Possibility of testing using user de_ned algorithms with the help of the Pro-

grammability feature.

e. Repair Data Calculation is managed automatically by the BIST.

f. Advanced Diagnostic Features with the help of the BITMAP feature.

g. Provides IP reuse for memory testing with implemented self test.

h. Could be used for burn-in tests.

B. Limitations of using memory BIST

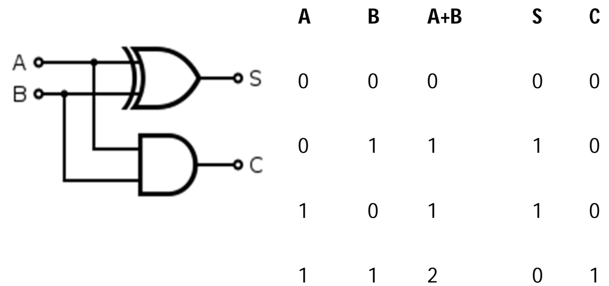
Since BIST is integrated on the silicon as a part of design, it has the following limi- tations:

a. Increases the memory area

But these disadvantages are minor, as compared to the brilliant bene?cial features it offers

III. Adder Circuits

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s),but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar.

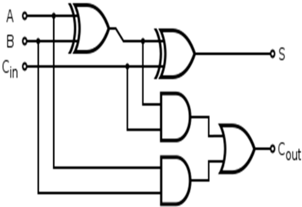


A	B	A+B	S	C
0	0	0	0	0
0	1	1	1	0
1	0	1	1	0
1	1	2	0	1

ISSN 2229-5518

A. Half Adder

A half-adder is a circuit which adds two bits together and outputs the sum of those two bits. The half-adder has two outputs: **sum** and **carry**. Sum represents the remainder of the integer division A+B/2, while carry is the result.



B. Full Adder

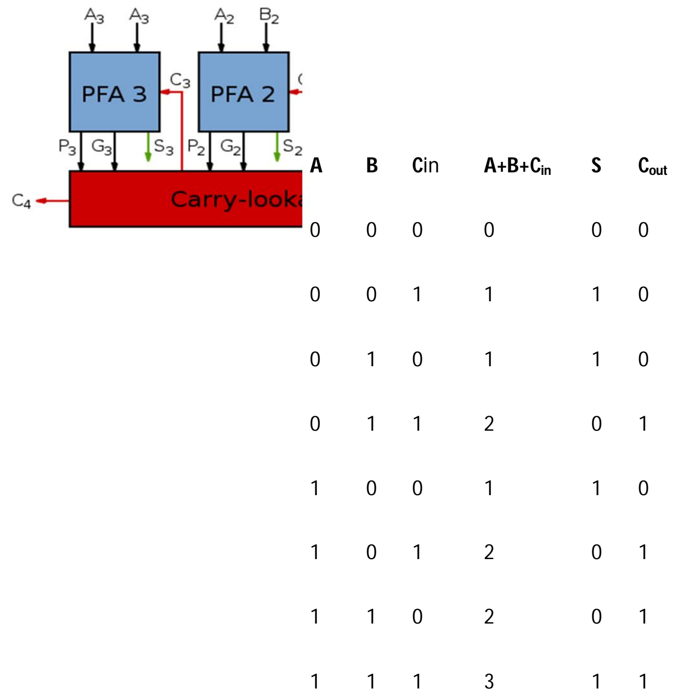
The full-adder can accept three bits as an input. Commonly, one bit is referred to as the carry-in bit. Full adders can be cascaded to produce adders of any number of bits by daisy-chaining the carry of one output to the input of the next.

C. Ripple-Carry Adder

A ripple carry adder is simple several full adders connected in a series so that the carry must propagate through every full adder before the addition is complete. Ripple carry adders require the least amount of hardware of all adders, but they are the slowest. The following diagram shows a four-bit adder, which adds the numbers A[4:0] and B[4:0], as well as a carry input, together to produce S[4:0] and the carry output.

D. Carry-lookahead Adder

A fast method of adding numbers is called carry-lookahead. This method doesn't require the carry signal to propagate stage by stage, causing a bottleneck. Instead it uses additional logic to expedite the propagation and generation of carry information, allowing fast addition at the expense of more hardware requirements.



A	B	Cin	A+B+C_{in}	S	C_{out}
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	1	1	0
0	1	1	2	0	1
1	0	0	1	1	0
1	0	1	2	0	1
1	1	0	2	0	1
1	1	1	3	1	1

A_i	B_i	C_i	G_i	P_i	C_{i+1}
0	0	0	0	0	0
0	0	1	0	0	0

IJSER © 2013

ISSN 2229-5518

0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	1	1

IV. PREVIOUS WORK ON MEMORY BIST AND MOTIVATION

There are two main approaches for testing embedded memories: external test by direct access using ATE and internal test using BIST. On the one hand, direct access to the embedded memory cores from the limited number of I/O pins needs a highperformance ATE, as well as very long testing time since tester channels are timeshared by different memories under test. Thus, external test becomes infeasible, in particular for large SOC devices where transistor to pin ratio is high. On the other hand, BIST provides at-speed and high-bandwidth access to the embedded memory cores,and it only needs a low cost ATE to initialize the test sessions and to inspect the final results. However, although BIST is state-of-the-art technology for embedded memory testing, unless carefully designed, it may induce excessive power, in addition to performance and area overhead. Since embedded memories account for more than 60% of the silicon area in modern SOCs (up to 95% by 2016) this chapter describes the relevant approaches to embedded memory BIST, summarizes their strengths and limitations and motivates the research presented in this thesis.

A. *Memory BIST Challenges*

A typical embedded memory BIST (MBIST) approach comprises an MBIST wrap- per, an MBIST controller and the interconnect between them. The MBIST wrapper further includes an address generator to provide complete memory address sequences(i.e.,for n address lines all the 2n locations are visited in a complete sequence);a background pattern generator to produce data patterns when testing word oriented memories (as described in the preceding chapter);a comparator to check the memory output against the expected correct data pattern; a finite state machine(FSM) to generate proper test control signals based on the commands received from the MBIST controller. The MBIST controller pre-processes the commands re- ceived from upper-level controller (either on-chip microprocessor or off-chip ATE) and then sends them to the MBIST wrapper.The interconnect between the wrapper and the controller could be either serial (i.e., a single command line is shared by all the wrappers) or parallel (i.e., dedicated multiple command lines are linking different wrappers to the controller). Note, the previously described partition of the MBIST architecture and the terms MBIST wrapper and MBIST controller are not universal, and only applicable in this thesis.

BIST addresses most of the challenges faced by testing embedded memories in an SOC. However, the increasing size and number of embedded memory cores and the rapid development in VLSI process technologies lead to unique requirements for embedded memory BIST. Various tests are as follows:

a. *Support multiple test algorithms*

b. *Diagnosis and repair support*

c. *Test heterogeneous memories*

d. *Power dissipation constraints*

e. *Reuse the available on-chip processing/communication resources*

f. *Design reuse*

V. WORK ON ADDER ARCHITECTURE

Part-1

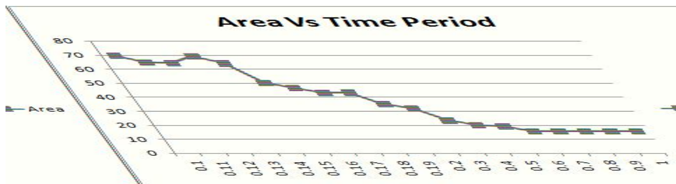
Ob jective: To perform synthesis at various frequencies using specified target library at worst case operating condition to find the critical frequency and area requirement for the different kind of adders. Also verifying that the adder type is not changed after synthesis and check which one is the fastest adder.

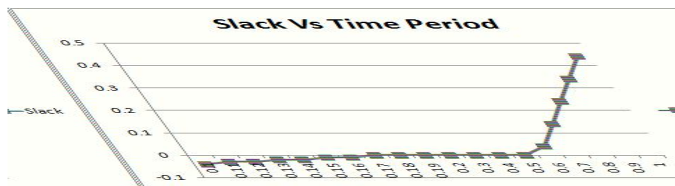
Types of Adder Used: Ripple carry adder, Carry look-ahead adder, Carry skip adder, Carry select adder

Target library: Using standard cell library in 32nm technology of ST

A. *Analysis of Ripple Carry Adder*

Time Period (ns)	Area (nm)	Slack (ns)	Critical Path Timing (ns)
0.1	69.52	-0.04	0.14
0.11	64.84	-0.03	0.14
0.12	64.3	-0.03	0.15
0.13	69.2	-0.02	0.15
0.14	64.08	-0.02	0.16
0.15	50.156	-0.01	0.16
0.16	46.67	-0.01	0.17
0.17	43.08	0	0.17
0.18	43.3	0	0.18
0.19	35.25	0	0.19
0.2	31.98	0	0.2
0.3	23.5	0	0.3
0.4	20.23	0	0.39
0.5	19.25	0	0.5
0.6	15.77	0.04	0.56
0.7	15.77	0.14	0.56
0.8	15.77	0.24	0.56
0.9	15.77	0.34	0.56
1	15.77	0.44	0





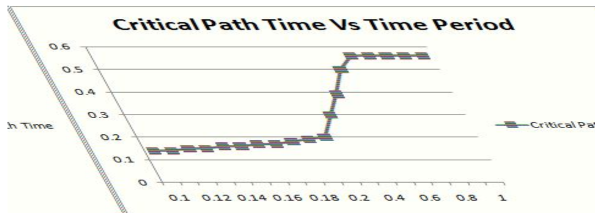
B. *Analysis of Carry Skip Adder*

Table:

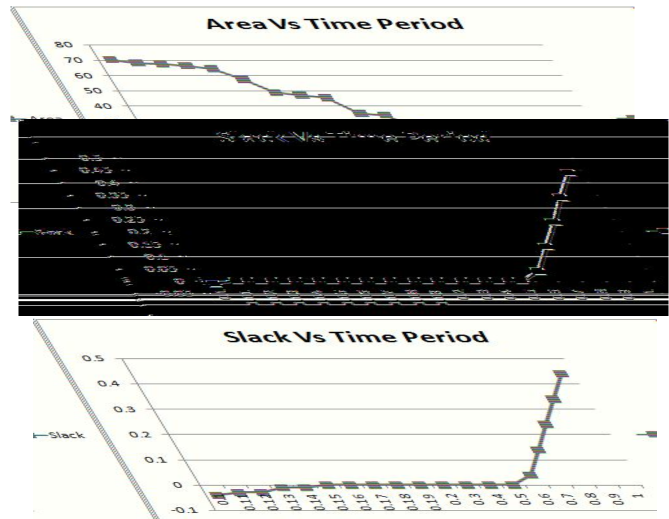
Time Period (ns)	Area (nm)	Slack (ns)	Critical Path Timing (ns)
0.1	70.16	-0.04	0.14
0.11	68.14	-0.03	0.14
0.12	67.35	-0.03	0.15
0.13	66.24	-0.01	0.14
0.14	64.32	-0.01	0.15
0.15	57.15	0	0.15
0.16	48.7	0	0.16
0.17	47.08	0	0.17
0.18	45.31	0	0.18
0.19	35.2	0	0.19
0.2	33.98	0	0.2
0.3	25.5	0	0.3
0.4	22.23	0	0.39
0.5	21.25	0	0.5
0.6	17.82	0.04	0.56
0.7	17.82	0.14	0.56
0.8	17.82	0.24	0.56
0.9	17.82	0.34	0.56
1	17.82	0.44	0

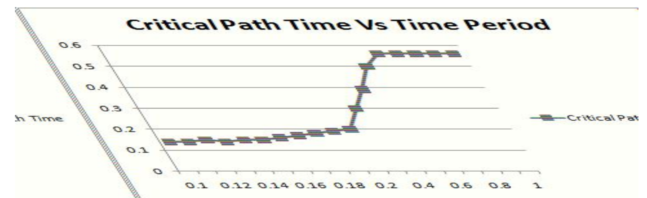
C. *Analysis of Carry-Look ahead Adder*

Table:

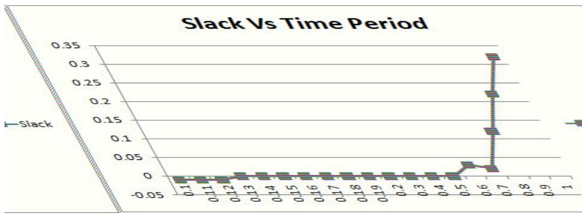


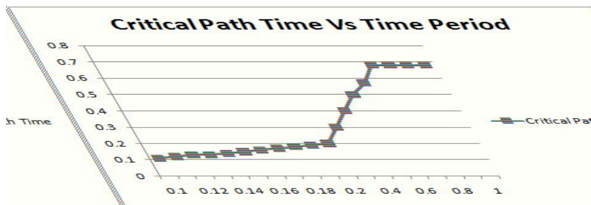
Time Period (ns)	Area (nm)	Slack (ns)	Critical Path Timing (ns)
0.1	65.82	-0.01	0.11
0.11	54.07	-0.01	0.12
0.12	46.24	-0.01	0.13
0.13	37.32	0	0.13
0.14	52.33	0	0.14
0.15	39.92	0	0.15
0.16	38.73	0	0.16
0.17	40.15	0	0.17
0.18	26.438	0	0.18
0.19	31.11	0	0.19
0.2	31.11	0	0.2
0.3	20.345	0	0.3
0.4	16.32	0	0.4
0.5	16.1	0	0.5
0.6	15.01	0.03	0.57
0.7	14.25	0.02	0.68
0.8	14.25	0.12	0.68
0.9	14.25	0.22	0.68
1	14.25	0.32	0.68





ISSN 2229-5518

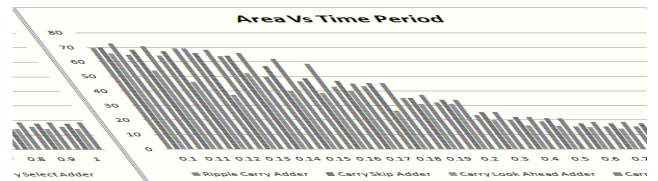


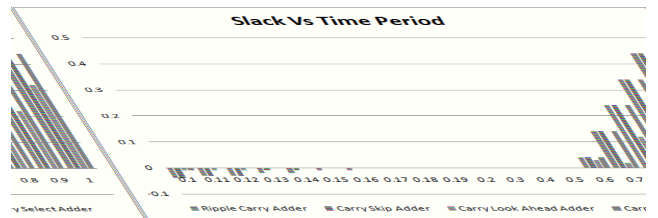


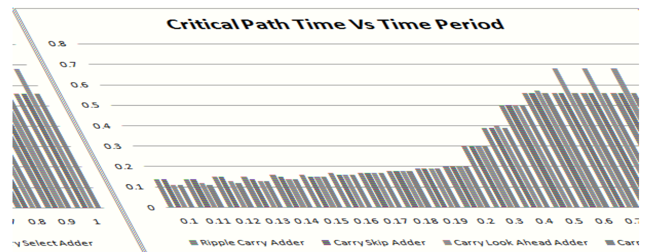
D. *Analysis of Carry Select Adder*

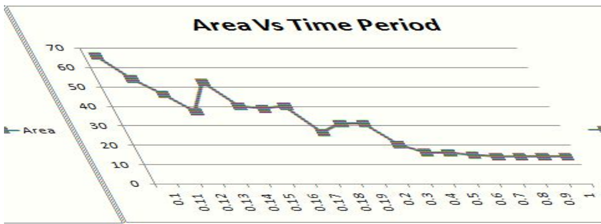
Table:

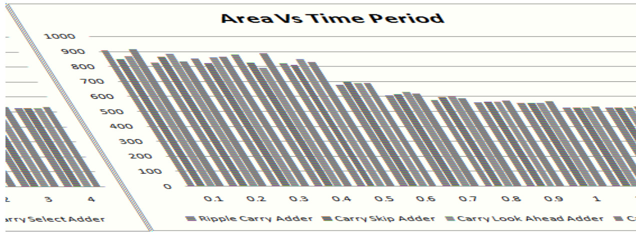
Time Period (ns)	Area (nm)	Slack (ns)	Critical Path Timing (ns)
0.1	72.52	-0.01	0.11
0.11	70.81	0	0.11
0.12	69.33	0	0.12
0.13	68.19	0	0.13
0.14	66.08	0	0.14
0.15	62.15	0	0.15
0.16	58.67	0	0.16
0.17	45.08	0	0.17
0.18	45.3	0	0.18
0.19	37.25	0	0.19
0.2	33.98	0	0.2
0.3	25.5	0	0.3
0.4	22.23	0	0.39
0.5	21.25	0	0.5
0.6	18.78	0.04	0.56
0.7	18.78	0.14	0.56
0.8	18.78	0.24	0.56
0.9	18.78	0.34	0.56
1	18.78	0.44	0.56











E. Comparison of Adders

i. Comparison of Area for different Adders

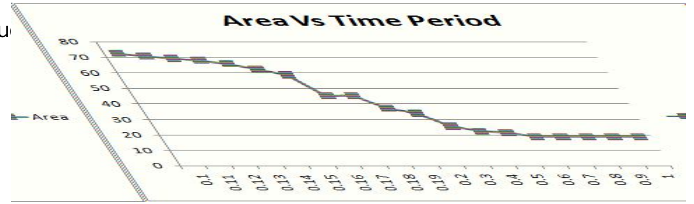
ii. Comparison of Slack for different Adders

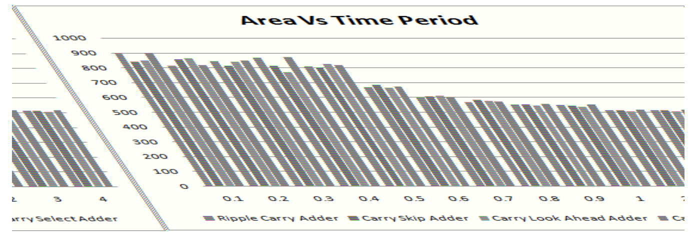
iii. Comparison of Critical Path for different Adders

Part-2

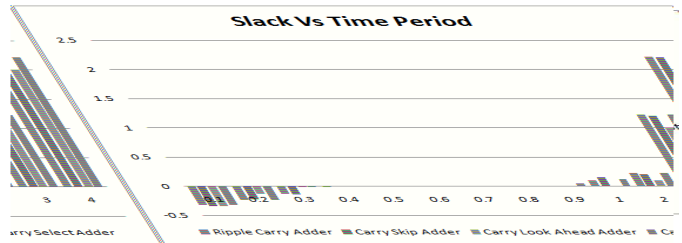
Ob jective:Analyze the area and timing requirement of BIST and adder. Analyze that the same result of area and timing are met for the adder after synthesis at different range of frequencies or not as per the Part-1.

Types of Adder Used: Ripple carry adder, Carry look-ahead adder, Carry skip adder, Carry select adder .**Target library:** Using standard cell library in 32nm technology of ST





A.Comparison of Adders of Part-2



i. Comparison of area after implementation of adder in BIST

ii. Comparison of Slack after implementation of adder in BIST

iii. Comparison of critical path after implementation of adder in BIST

Part-3

Ob jective:Analyze the area and timing requirement of BIST using boundary opti- mization. Analyze that the same result of critical path timing of BIST is met or not for different range of frequencies as per the Part-2.

Types of Adder Used: Ripple carry adder, Carry look-ahead adder, Carry skip adder, Carry select adder.

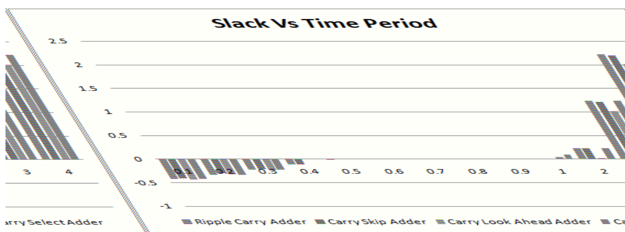
Target library: Using standard cell library in 32nm technology of ST

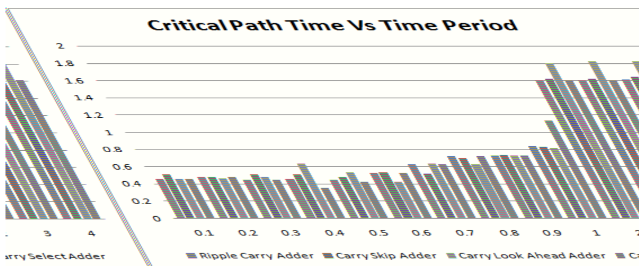
A. Comparison of Adders of Part-3

i. Comparison of area after implementation of adder in BIST with boundary optimization

ii. Comparison of Slack after implementation of adder in BIST with boundary optimization

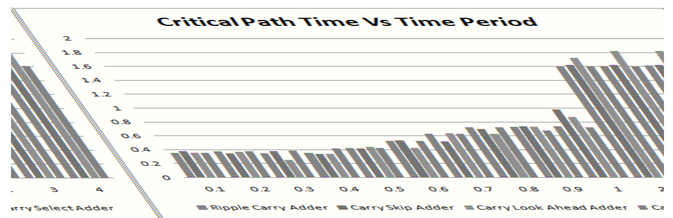
iii. Comparison of critical path after implementation of adder in BISt with boundary optimization





VI. Conclusion

Comparing the performance metrics for the 8-bit adders implemented in BIST, using Synopsys synthesis tools, the trade offs becomes apparent. As can be seen there exist an inverse relationship between time delays, operating speed, and circuit area, in this case the number of CLBs (measure of the area). The ripple carry adder, the most basic of flavours, is at the one extreme of this spectrum with the least amount of CLBs but the highest delay. The carry select adder on the other hand, is at



the opposite corner since it has the lowest delay (half that of the ripple carry.s) but with a larger area required to compensate for this time gain. Finally, the carry look-ahead is middle ground. Power dissipation, for this case study, is in direct proportion to the number of CLBs.

VII. References

[1] .Abramovici, M. Breuer, and A. Friedman. Digital Systems Testing and

Testable Design. IEEE Press IEEE,1995

[2] .M.Al-Harbi and S. K. Gupta. An Efficient Methodology for Generating Optimal and Uniform March Tests. In Proc. IEEE VLSI Test Sympo- sium,pages 231-237,2001.

[3] . Appello, F. Corno, M. Giovinetto, M. Rebaudengo, and M. S. Reorda.

AP1500 Compliant BIST-Based Approach to Embedded RAM Diagno- sis. In Proc. IEEE Asian Test Symposium, pages 97-102, 2001.

[4] . H. Bardell, W. H. McAnney, and J. Savir. Built-In Test for VLSI: Pseudorandom Techniques. John Wiley & Sons, Inc., New York, 1987.

[5] . J. Bergfeld, D. Niggemeyer, and E. M. Rudnick. Diagnostic Testing of Embedded Memories Using BIST . In Proc. of the Design, Automation and Test in Europe Conference, pages 305-309, 200

[6] . L. Bodoni, A. Benso, S. Chiusano, S. D. Carlo, G. D. Natale, and P.

Prinetto.An Elective Distributed BIST Architecture for RAMS. In Proc. IEEE European Test Workshop, pages 119-124, 2000.

[7] . T. Chen, J. Rajski, J. Khare, O. Kebichi, and W. Maly. Enabling Embedded Memory Diagnosis via Test Response Compression. In Proc. IEEE VLSI Test Symposium, pages 292-298, 2001.

[8] . Cheung and S. K. Gupta. A BIST Methodology for Comprehensive Testing of RAM with Reduced Heat Dissipation. In Proc. IEEE Inter- national Test Conference, pages 386-395, 1996.

[9] . M. Chou, K. K. Saluja, and V. D. Agrawal. Scheduling Tests for VLSI Systems Under Power Constraints. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 5(2):175-184, June 1997.

[10] . T. Gonciari, B. M. Al-Hashimi, and N. Nicolici. Test Data Compression: the System Integrators Perspective. In Proc. of the Design, Automation and Test in Europe Conference, pages 726-731, 2003

[11] . L. Craig, C. R. Kime, and K. K. Saluja. Test Scheduling and Control for

VLSI Built-In Self-Test. IEEE Transactions on Computers, 37(9):1099-

1109, September-1988.

[12] . Dreibelbis, J. Barth, H. Kalter, and R. Kho. Processor-Based Built-In Self-Test for Embedded DRAM. Solid-State Circuits, 33(11):1731-1740, November-1998.

.