

# Improved Fault Tolerant Sparse KOGGE Stone ADDER

Mangesh B Kondalkar<sup>1</sup>Arunkumar P Chavan<sup>2</sup>P Narashimaraja<sup>3</sup><sup>1, 2, 3</sup> Department of Electronics and Communication, R V college of Engineering, Bangalore 560059, India.

## ABSTRACT

A fault tolerant adder implemented using Kogge-stone configuration can correct the error due to inherent redundancy in the carry tree but no error detection is possible. This proposed design is based on fault tolerant adder [1] that uses Sparse kogge-stone adder that is capable of both fault detection and correction. Fault tolerance is achieved by using two additional ripple carry adders that form the basis of triple mode redundancy adder. Triple mode redundancy is one of the most common methods used to create fault tolerant designs in both ASIC and FPGA implementations. The latency will be increased because of the voter in the circuit's critical path. More advanced fault tolerant methods exist including roving and graceful degradation approaches. Allowing fault tolerance to operate at different levels of abstraction might facilitate a more cost-effective design. Several enhancements are introduced in the design; the error recovery time is reduced by using a 16-bit register, error correction due to fault in multiple ripple carry adders is included which improves the reliability of the circuit. The power analysis and the timing analysis for the estimation of setup time and hold time is also performed.

## General Terms

Sparse Kogge-Stone adder, Triple Mode Redundancy, Fault tolerant.

## Keywords

RCA, GP block, Gray Cell, Black Cell.

## 1. INTRODUCTION

Fault tolerance plays a very important role in modern systems where immediate human intervention is not possible and system failure can have disastrous consequences. An extreme temperature change is one of the reasons in which fault tolerance is necessary for devices operating in harsh operating environments, as found, for example, in space and military applications. Fault tolerance will also be necessary in nano-electronic systems, as small device dimensions make the system more susceptible to outside interference, such as space radiation.

A fault tolerant system has the ability to detect and then correct the occurrence of a hardware malfunction. In order to detect the fault, the system must be able to sense any deviations from its normal operation. A fully fault tolerant system also has the ability to correct the fault in order to return the system to its normal functionality. In many digital systems the adder circuit decides the critical path delay such as digital signal processor and hence this paper will focus on optimization of an adder circuit. The adder circuit should detect and correct error with minimum recovery time and

should minimize the amount of extra logic and hardware required.

The complete work is explained in the subsequent sections. Section 2 provides the background information and research review in the area of fault tolerant circuits. The design and implementation of the fault tolerant adder based upon Sparse kogge-stone (SKS) adder is explained in section 3. Conclusion and discussion of ongoing future work in this area will be provided in section 4.

## 2. BACKGROUND

Basic fault tolerance can be achieved by N-module redundancy (NMR) where N refers to the degree of redundancy used in the design. This approach is easy to apply but results in high area overhead. Triple Modular Redundancy (TMR) is a fault tolerant method where the hardware is essentially replicated in triplicate with a voter circuit used to pass the majority rule signals to the output. Three copies of the same circuit are connected to a majority voter which is used to obtain the fault free output. This method works as long as all the faults are confined to one of the redundant blocks. The latency will be increased because of the voter in the circuit's critical path. The triple modular redundant ripple carry adder (TMR-RCA) is used as the reference design as shown in Figure 1.

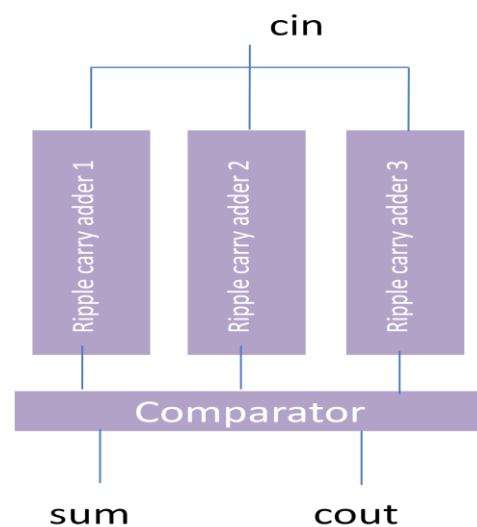


Fig. 1: Triple mode redundancy RC adders

Triple Mode redundancy Ripple carry (TMR-RC) is simple and effective method of fault detection and correction but it increases the area overhead and almost results in tripling of the associated power dissipation. Compared to the basic TMR-RCA, more advanced fault tolerant methods exist

including roving and graceful degradation approaches. Allowing fault tolerance to operate at different levels of abstraction might facilitate a more cost-effective design [2].

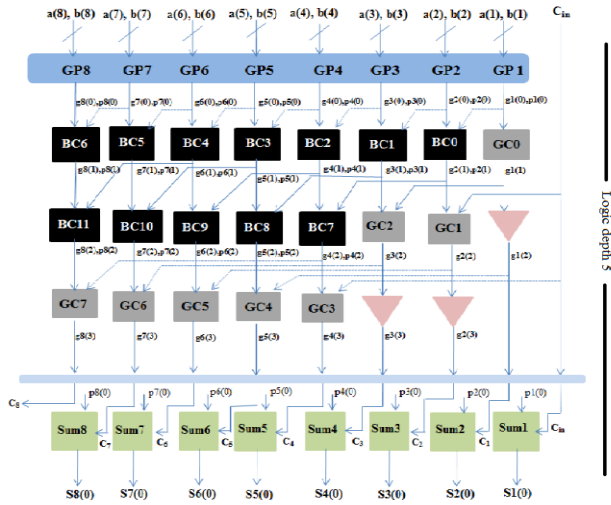


Fig. 2: 8 bit Kogge-Stone Adder (Logic depth 5)

The Kogge-Stone adder (KSA) [3] is classified as a parallel prefix adder since the generate and the propagate signals are pre computed. In a tree-based adder, carries are generated in tree and fast computation is obtained at the expense of increased area and power. The main advantage of this design is that the carry tree reduces the logic depth of the adder by essentially generating the carries in parallel. The parallel-prefix adder becomes more favorable in terms of speed due to the  $O(\log 2n)$  delay through the carry path compared to  $O(n)$  for the Ripple Carry Adder (RCA). The differences in terms of logic depth and number of logic blocks can be seen by comparing Figure 2, which illustrates a 8 bit Kogge-stone Adder, with Figure 3, which depicts a 8 bit RCA. The carry block in Figure 2 consists of the generate-propagate (GP) blocks, which are categorized as Black Cells (BC) and Gray Cells (GC) as defined in [4].

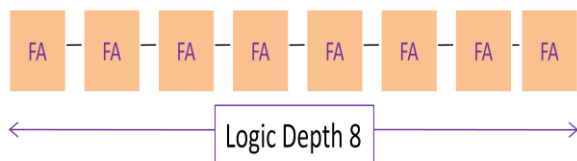


Fig. 3: 8 bit Ripple Carry Adder (Logic depth 8)

The number of GC and BC increases for adder with larger bit width. Table 1 summarizes the number of different cells required for adder of different bit widths.

Table 1. Kogge-Stone adders of different bit widths

Bit width of KSA	No. of GP cells	No. of Black cells	No. of Gray cells	No. of Sum blocks
16-bit	16	37	16	16
64-bit	64	257	64	64
128-bit	128	641	128	128
256-bit	256	1537	256	256

The another important advantage of using KSA is its mutual exclusive nature. if a defect is present in the one-half of the carry tree, the other half can be utilized to compute the carries for both the even and odd carries. The timing diagram for this design is illustrated in Figure 4. In the scenario depicted, three instructions are scheduled on three different adders present in the execution unit. The second adder is defective and is evaluated in two clock cycles whereas the fault free adders are evaluated in a single-clock cycle, assuming that the defect is in the odd bits. However this results in increased in time required for execution which can be reduced with the help of proper architectural design.

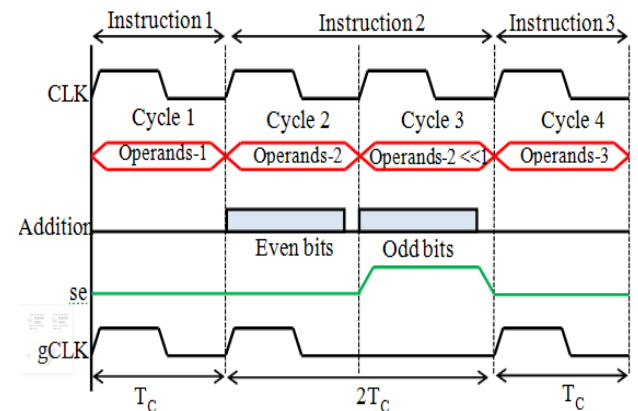
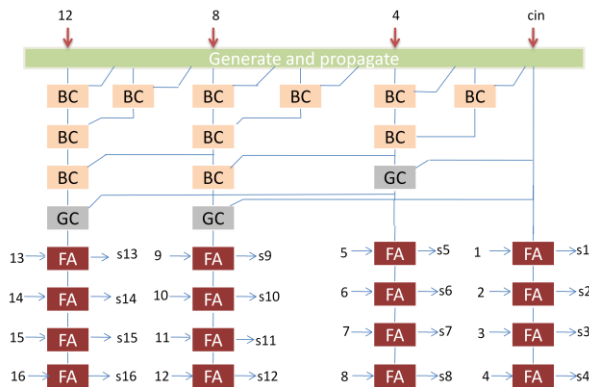


Fig. 4: Timing diagram for three adders in execution unit.

### 3. PROPOSED FAULT TOLERANT ADDER DESIGN

The novel design uses Sparse Kogge-Stone (SKS) adder consists of several smaller ripple carry adders (RCA) on its lower half and a carry tree on its upper half. Thus, the SKS adder terminates with RCAs. The number of generated carries is less in a SKS adder compared to the regular Kogge-Stone adder. The functionality of the GP block, black cell and the gray cell remains exactly the same as in the regular Kogge-Stone adder. The schematic for a 16-bit SKS adder with a factor of four reduction in the carry tree is shown in Figure 5. The advantage of combining SKS adder along with RCA is that latter takes less time to compute carries by generating them in parallel but requires more area on the other hand RCA

requires less hardware but generates carries serially and thus both the timing and area overhead are balanced.

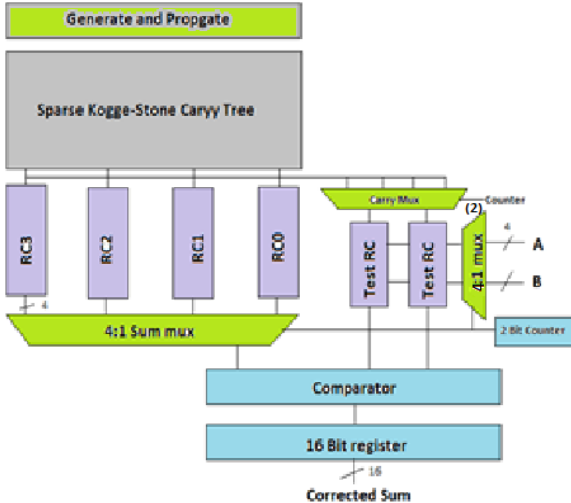


**Fig. 5: A 16 bit Sparse Kogge-Stone (SKS) Adder**

As shown in Figure 5 only three carries are generated using SKS tree structure and four ripple carry adders are used to complete summation process. The additions required to make this basic adder circuit fault tolerant are described in two steps. First lower half of the adder which includes RCA's and second the upper half which consists of SKS tree structure.

### 3.1 Lower Half SKS Fault Tolerant design

The addition of RCA in the lower half makes it possible to use similar testing methodology as used in TMR-RC adder. In order to detect error in any of the RCA two Test RCA are included as shown in the Figure 6.

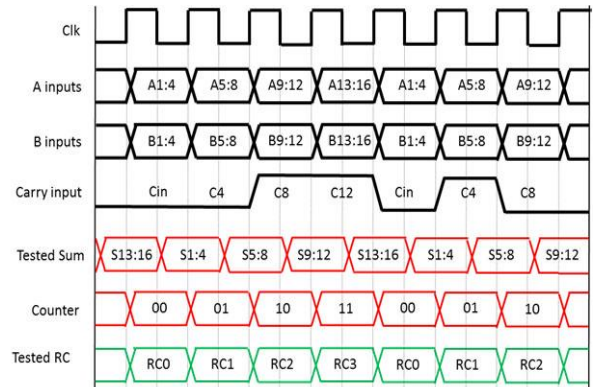


**Fig. 6: Fault Tolerant Sparse Kogge Stone Adder**

Some multiplexers and a bit counter are also added for fault detection and correction. During each input clock cycle one of the four RCA (RC0 – RC3) is selected for testing. The corresponding carry input and the operand A and B are routed to both Test RCA by a multiplexer. The selection of a RCA is controlled by a 2-bit counter. The Sum outputs from Test RCA and the selected RCA is given to a comparator which selects majority output. Due this even if one of the selected

RCA becomes faulty, the correct output is available at the comparator output.

A timing diagram illustrating this process is shown in Figure 7.



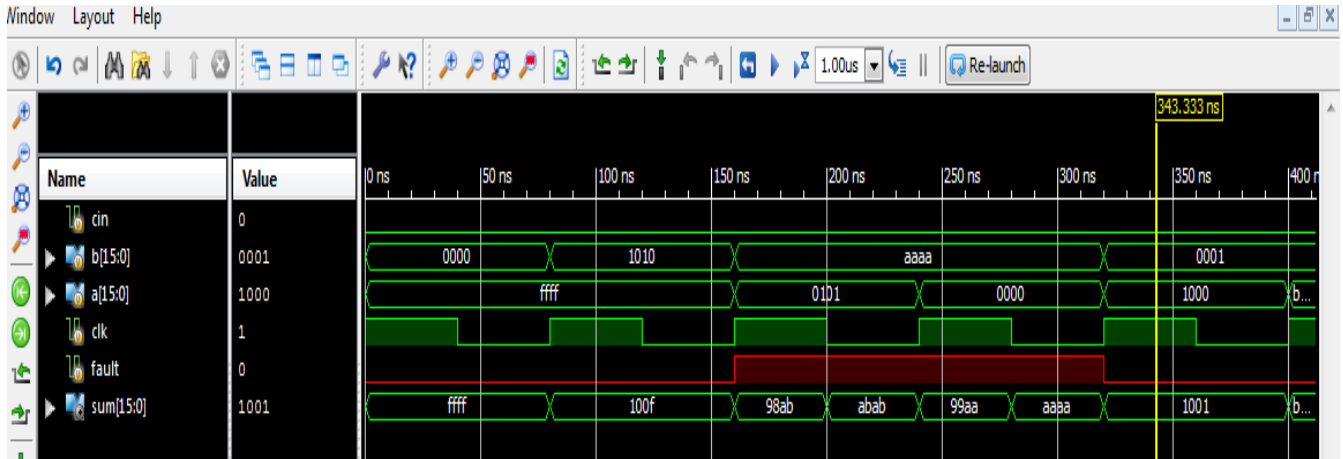
**Fig.7: Timing diagram for lower half SKS adder**

The design proposed in [1] can detect and correct error only if any one of the RCA becomes faulty as the clock signal to the bit counter is stopped if the fault is found in any of the RCA. In order to correct the errors in more than one RCA a 16-bit register is added to the design which holds the sum output from the comparator. For each count value starting from 00 to 03 of a bit counter a corrected sum from the comparator is stored in the register. Thus allowing error correction in case more than one RCA becomes faulty. The improvement introduced in the proposed design is the reduction in the error recovery time which is very critical in any fault tolerant circuit. As the corrected sum output is available in the register after all the RCA are tested the final sum output can be sampled from the register and no extra clock cycles are required to correct error after fault detection as required in the design proposed in [1].

The final design is coded in verilog and simulated using ModelSim. The simulation results shown in Figure 8 illustrate the successful detection and correction of error in one of the RCA chains. For illustration the fault is introduced in the RC2. Figure 9 shows the error correction and detection when more than one RCA becomes faulty. The input signal Fault is used to inject fault in any of the RCA.

As seen from the graph during positive cycle of clock input the addition is performed and simultaneously each RCA is tested for fault and the corrected sum is stored in the register. On the negative edge of the clock input the corrected sum is available. Thus the correction delay is equal to the duration of the half cycle of the clock input. Similarly when the fault is injected into two RCA the circuit gives the corrected sum on the falling edge of the clock input. It should be noted that for each clock input the bit counter goes from count 00 to 03 twice for each half cycle and thus clock for the bit counter should be derived from the system clock.

The synthesis results for a TMR-RCA, regular Kogge-Stone fault tolerant adder, proposed fault tolerant sparse Kogge-Stone for both lower half and upper half, and graceful degradation are obtained for a Vertex5 FPGA. Design statistics are obtained by synthesizing the adders using Xilinx ISE software in two ways. First, the number of resources in terms of look up tables (LUTs) is observed. The results



obtained are shown in Figure 10 which gives an estimation of the number of the look up tables (LUTs) used by each design.

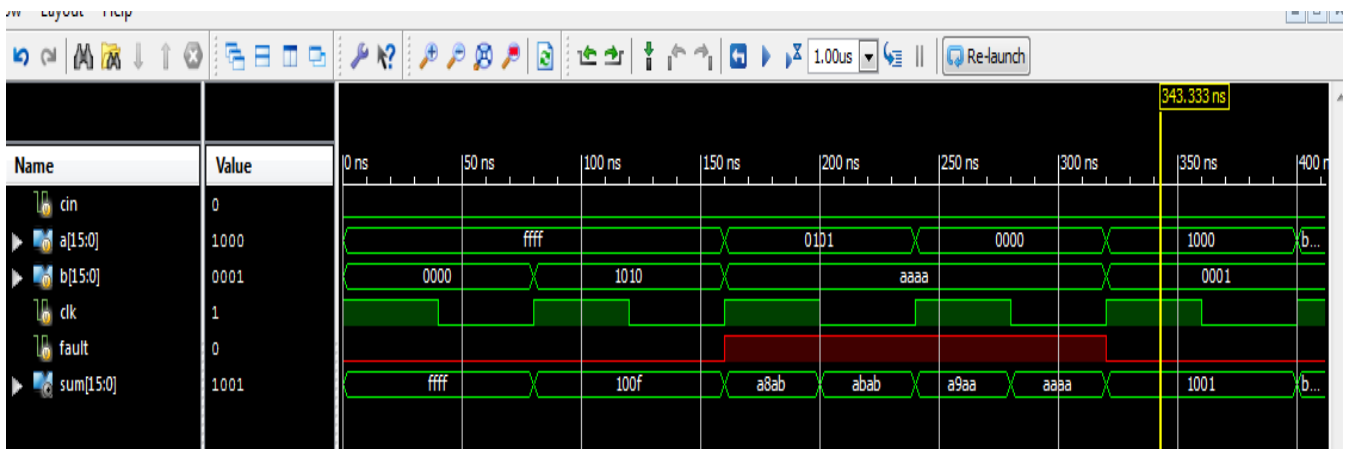


Fig.8: Simulation Result for Fault Tolerant SKS adder

Fig.9: Simulation Result for Fault Tolerant SKS adder

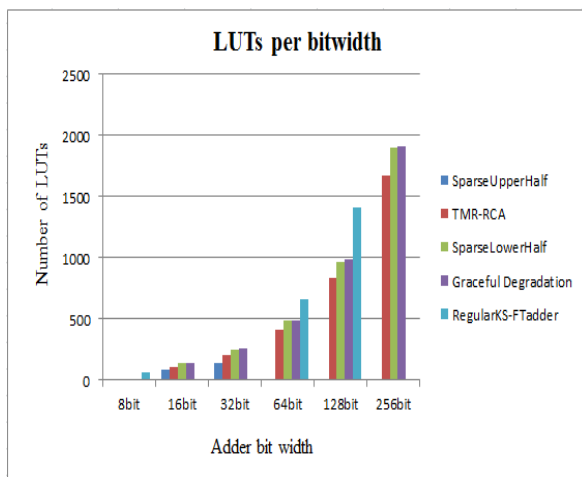


Fig.10: Resource estimation from FPGA synthesis

The power analysis is performed to estimate the total static and dynamic power dissipation. As shown in

Figure 11 the total Dynamic power dissipation is only 2mW.

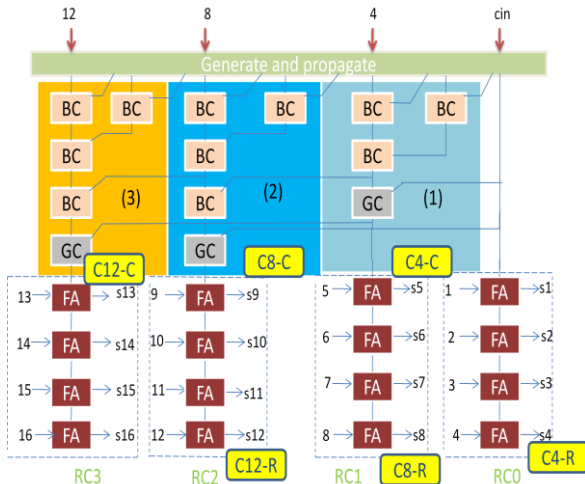
A	B	C	D	E	F	G	H	I
Device		On-Chip	Power (W)	Used	Available	Utilization (%)		
Family	Virtex5	Clocks	0.002	2	---	---		
Part	xc5vx110t	Logic	0.000	101	63120	0		
Package	#1136	Signals	0.000	139	---	---		
Grade	Commercial	I/Os	0.000	59	640	9		
Process	Typical	Leakage	1.188					
Speed Grade	-2	Total	1.190					

Fig.11: Total Power dissipation

### 3.2 Upper half SKS Fault Tolerant design

In order to have a complete fault free adder circuit the carry tree should be made fault free. This section focuses on making carry tree of SKS adder fault tolerant. For a 16 bit SKS adder there are two sets of carry generated for C4, C8

and C12. There are two approaches to make tree structure fault tolerant. The first approach by making the use of generated by the RCA after each RCA is corrected by comparing it with the test RC. For example if the mismatch is found in C4-C and C4-R as shown in Figure 12 the carry out produced by the first RCA, C4-R is routed to the Test RCs instead of C4-C to produce completely fault free value of the next carry i.e C8-R. In this approach it is assumed that the RC0 is fault free. The second approach is to divide the tree into three sections and detecting the faulty section. Sparse for each section can be made available to replace a faulty section by using multiplexer to reroute the carry tree from faulty branches to the spare section.



**Fig.12: Error detection scheme for upper half**

if carry C4-C does not match carry C4-R, the error must be located in the first section assuming the first ripple carry adder (RC0) is fault free. If the result obtained is fault free, a fault free carry C4 will enter the second ripple carry adder (RC1). Next, if a mismatch is found in the C8-R and C8-C pair, there exists a fault in the second section. Finally, if the carries C8 and C4 are fault free, the error can be in the third section if a mismatch is found in the C12 pair. The fault detection mechanism is summarized in Table 2.

**Table 2. Truth table to detect faulty section**

C12	C8	C4	Faulty Section
X	X	1	Green
X	1	0	Purple
1	0	0	Blue
0	0	0	No Fault

X = Don't care, 1 = Carry mismatch, 0 = Carry match

## 4. CONCLUSION

A Sparse Kogge-Stone adder which is fully fault tolerant in its lower half (i.e., in the ripple carry adders) was proposed.

Simulation results demonstrate that this design is able to detect and correct error in its RCA chains. A fault tolerant Sparse Kogge-Stone adder is designed by taking advantage of the existing ripple carry adders in the architecture and adopting a similar approach to the TMR-RCA by inserting two additional ripple carry adders into the design. The addition of register reduces the error recovery time and also makes it possible to detect and correct the error in multiple RCA.

## 5. FUTURE WORK

Two main areas for extending the present work are briefly considered. First, the development of methods and tools to make the proposed fault tolerant methods easier to implement can be undertaken. A method for easily scaling to larger bit widths for the upper half fault tolerant Sparse Kogge-Stone adder should be investigated. Automated techniques for implementing the fully fault tolerant Sparse Kogge-Stone adder should be developed. Second, a largely unexplored area of research is the application of error correcting codes to fault tolerant adder designs. In digital communications, an additional number of bits is added to a message to allow the detection and correction of corrupted bits during transmission. A similar method might be feasible with arithmetic circuits. An optimal error correcting code would take into account the logic structure of the adder and would enable fully fault tolerant implementations while adding a minimum amount of overhead.

## 6. ACKNOWLEDGEMENT

We sincerely like to thank all the people who have directly and indirectly encouraged us and helped us in working out our research. Thank to R.V College from where we got complete support. Our thanks to the experts who have contributed towards implementation of the circuit.

## 7. REFERENCES

- [1] Chris D. Martinez, L. P. Deepthi Bollepalli, and David H. K. Hoe, "A Fault Tolerant Parallel-Prefix Adder for VLSI and FPGA Design", 44th IEEE Southeastern Symposium on System Theory, 2012.
- [2] R. Iris, D. Hammerstrom, J. Harlow, W. H. Joyner Jr., C. Lau, D. Marculescu, A. Orailoglu, M. Pedram, "Architectures for Silicon Nanoelectronics and Beyond," Computer, vol. 40, no. 1, pp. 25-33, Jan. 2007.
- [3] P. M. Stone and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence equations," *IEEE Trans. on Computers*, vol. C-22, no. 8, pp. 786-793, Aug. 1973 [5] S. Ghosh, P. Nda, and K. Roy, "A Novel Low Overhead Fault Tolerant Kogge-Stone Adder using Adaptive Clocking," *Design, Automation and Test in Europe*, pp. 366- 371, 2008.
- [4] N. H. E. Weste and D. Harris, *CMOS VLSI Design*, 4<sup>th</sup> edition, Pearson-Addison-Wesley, 2011.
- [5] D. H. K. Hoe, C. Martinez, and J. Vundavalli, "Design and Characterization of Parallel Prefix Adders using FPGAs," *IEEE 43rd Southeastern Symposium on System Theory*, pp. 170-174, March 2011.
- [6] S. Ghosh, P. Nda and K. Roy, "A Novel Low Overhead Fault Tolerant Kogge-Stone Adder using Adaptive

Clocking,” Design, Automation and Test, pp. 366-371, 2008.

- [7] T. Lynch and E. E. Swartzlander, “A Spanning Tree Carry Lookahead Adder,” IEEE Transactions on Computers, vol. 41, no. 8, pp. 931-939, Aug. 1992.
- [8] J. Vundavalli, “Design and Analysis of wide bit adders for FPGA Implementation,” MSEE Thesis, University of Texas at Tyler, May 2010.
- [9] N. Banerjee, C. Augustine, K. Roy, “Fault-Tolerance with Graceful Degradation in. Quality: A Design Methodology and its Application to Digital Signal Processing Systems,” IEEE International Symposium on, Defect and Fault Tolerance of VLSI Systems, pp. 323-331, 1-3 Oct. 2008.
- [10] M. Abramovici, J. M. Emmert, and C. Stroud, “Roving STARS: An Integrated Approach to On-Line Testing, Diagnosis, and Fault Tolerance for FPGAs,” NASA/DoD Workshop on Evolvable Hardware, pp. 73-92, 2001.

## **AUTHORS PROFILE**

**A.Mangesh B Kondalkar** born on August 8<sup>th</sup>, 1988 in Mumbai, India, obtained his B.E degree in Electronics Engineering from Mumbai University, Mumbai, India. Currently pursuing M. Tech in VLSI Design and Embedded Systems. His areas of interest are VLSI design, and Digital electronics and Design.

**B. Arunkumar. P Chavan** born on July, 4<sup>th</sup>, 1987 in Karnataka, India, obtained his B.E degree in Electronics and Communication Engineering from Visvesvaraya Technological University (VTU), Belgaum, India. Currently pursuing M.Tech in VLSI Design and Embedded Systems. His areas of interest are VLSI design, Analog circuit design and digital electronics.

**C. Narashimaraja. P** born on January, 31<sup>st</sup>, 1982 in Karnataka, India, obtained his BE degree in Electronics from Madurai Kamaraj university, in 2004 and ME degree in VLSI Design from Bharath institute of higher education & research, in 2006. Currently, Is an Assistant professor at RV College of Engineering. His research interest are in the field of VLSI architectures,