Design of Booth Multiplier using Efficient Carry Speculative Adder

R.MALLIKARJUNA REDDY
Associate Professor, Department of ECE, VEMU Institute of Technology, P Kothakota, Andhra Pradesh, India.

Abstract: **Arithmetic logic units and digital signal processors widely use adders. It is the most complicated arithmetic circuits in digital electronics. The existing adders suffer from critical path delay, area overhead and power consumption. Speculative adders are designed with variable latency that combines speculation technique along with correction methodology to attain high performance in terms of low area overhead over the existing adders. In speculative adders the sum and carry generation part is separated to reduce the area overhead. Carry Speculative Adder (CSPA) uses carry predictor circuit to reduce power consumption and to reduce the computational time and it uses error recognition and error correction circuit to find the fault occurred in the partial sum generator and to recover it to get accurate results.**

Index Terms **– Speculative Adder, Variable Latency, Error Detection, Error Correction**

1. INTRODUCTION

In electronics, addition of the binary numbers in various computers and other types of processors are performed by the adders. Adder circuits are used in various processors for calculating increment or decrement operations, table indices, addresses etc. The different formats like XS-3, Binary Coded Decimal (BCD) and gray code can be added using the adder circuits. Adder found wide range of applications in many fields and for many operations such as decoding, calculation etc. The critical path is not often activated in traditional adders, based on this observation speculative adders have been designed. Traditional adders depend on its previous values for its each output [1]. Particularly, the MSB of the sum depends on all the n bit previous outputs, where n is the block adder width. As the width of block adder increases, there will be an error growth. The error grows linearly with n. There will be a large area and large fanout at the primary inputs due to this error. Speculative adders can overcome the area problem but it has high error rate [2], [3]. For this error tolerant variable latency adder is design upon the speculative adder [4]. This variable latency adder consist of error recogmtIon and correction circuit, which can overcome the high error rate and this design helps the speculative adder to use in many applications such as image and signal processing etc [5].

In any digital system, adder is the most crucial arithmetic circuit. For traditional adders, the critical path delay and area over head are very high Ω (log n), Ω (n). Increasingly huge data sets and the need for instant response require the adder to be large and fast. Traditional ripple carry adder (RCA) is not suitable for large adders because of its low speed performance. To reduce the critical path delay and power consumption, approximate designs are used by sacrificing accuracy. Approximation can increase performance or reduce power consumption with a simplified or inaccurate circuit where strict requirements are relaxed. In an accuracy-configurable approximate (ACA) adder for which the accuracy of results is configurable during runtime. Because of its configurability, the ACA adder can adaptively operate in both approximate (inaccurate) mode and accurate mode. The ACA adder achieves approximately power reduction than conventional pipelined adder.

Speculative technique is an optimization technique based on a prediction mechanism for improving the delay of arithmetic circuits. Speculative adders are built upon the observation that the critical path is rarely activated in traditional adders. In speculative adders, each output depends only on the previous k bits rather than all previous bits. Static window addition (SWA), a novel function speculation technique for the design of low area overhead, high performance variable latency adders is proposed.

A block, called a window, includes several consecutive input bits. Grouping input bits into blocks, the carry chain length can be made comparable to the block size with high probability. Variable Latency addition using SWA based speculative adders is faster than the fastest Design Ware adder with area requirements for different adder widths. In ,a correlation aware speculative addition (CASA), which is a generic lightweight extension to existing speculative adders which intelligently exploits the correlation between the most significant bit of the input operands and the carry in values to improve the correctness of speculative adders. It shows that CASA achieves a significant reduction in error rate with small overhead in timing and area.

Carry Speculative Adder (CSPA) is used to reduce the critical path delay of the circuit which is based on carry speculation. The block diagram of the carry speculative adder (CSPA) is shown below in figure. The n-bit CSPA is divided into several small blocks adders that are operated independently and Carry Predictor Circuits. The size of each block adder is x-bit, except the leftmost block adder.

This paper is organized as section II describes Litarature review and in section III discussed about Proposed System and Section IV describes shows the simulation results and section V concludes the paper followed by references.

2. LITARATURE REVIEW

[1]. D. Mohapatra, A. Raghunathan, V. Gupta and K. Roy, "Low-power Digital signal processing using approximate adders, "IEEE Transaction Computer- Aided Design Integration Circuits Systems. vol. 32, no. I, pp. 124-137, Jan. 2013.

Digital signal processing (DSP) blocks form the back bone of various multimedia applications used in portable devices. Most of these DSP blocks implement image and video processing algorithms, where the ultimate output is either an image or a video for human consumption. Human beings have limited perceptual abilities when interpreting an image or a video. This allows the outputs of these algorithms to be numerically approximate rather than accurate. This relaxation on numerical exactness provides some freedom to carry out imprecise or approximate computation. We can use this freedom to come up with low-power designs at different levels of design abstraction, namely, logic, architecture, and algorithm.

[2]. N. Zhu, K. S. Yeo, W. L. Goh, and Z. H. Kong, "Design of low-power High-speed truncation-error-tolerant adder and its application in digital Signal processing," IEEE Transaction Very Large Scale Integration (VLSI) Syst., vol. 18, no. 8, pp. 1225-1229, Aug. 2010.

In conventional digital VLSI design, one usually assumes that a usable circuit/system should always provide definite and accurate results. But in fact, such perfect operations are seldom needed in our non digital worldly experiences. The world accepts "analog computation," which generates "good enough" results rather than totally accurate results. The data processed by many digital systems may already contain errors. In many applications, such as a communication system, the analog signal coming from the outside world must first be sampled before being converted to digital data. The digital data are then processed and transmitted in a noisy channel before converting back to an analog signal. During this process, errors may occur anywhere. Furthermore, due to the advances in transistor size scaling, factors such as noise and process variations which are previously insignificant are becoming important in today's digital IC design.

The concept of error tolerance (ET) and the PCMOS technology are two of them. According to the definition, a circuit is error tolerant if: 1) it contains defects that cause internal and may cause external errors and 2) The system that incorporates this circuit produces acceptable results the "imperfect" attribute seems to be not appealing.

[3]. P. Varman, K. Du and K. Mohanram, "High performance reliable variable latency carry select addition," in Proceedings, Design Automation Test Conference Exhibition (DATE), Mar. 2012, pp. 1257- 1262.

Addition, one of the most frequently used arithmetic operations, is employed to build advanced operations such as multiplication and division. Theoretical research has found that the lower bound on the critical path delay of the adder has complexity O (log n), where n is the adder width. The design of high performance adders has been extensively studied, and several adders have achieved logarithmic delays. Whereas theoretical bounds indicate that no traditional adder can achieve sub-logarithmic delay, it has been shown that speculative adders can achieve sub-logarithmic delays by neglecting rare input patterns that exercise the critical paths. Furthermore, by augmenting speculative adders with error detection and recovery, one can construct reliable variable-latency adders whose average performance is very close to speculative adders.

[4]. A. K. Verma, P. Brisk, and P. lenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proceedings, Design Automation Test Cof! Ference Exhibition (DATE), Mar. 2008, pp. 1250-1255.

Binary addition is one of the most frequently used arithmetic operations. It is a vital component in more complex arithmetic operations such as multiplication and division. Researchers have established lower bounds on the delay and area of the adder. In particular, an n-bit adder must have a delay Ω (log n) and area Ω (n). Certain adders having delay and area of the same complexity as the ones mentioned above have been presented in literature: A Ripple Carry Adder requires a linear number of gates, and fast adders such as Carry Look-Ahead Adders (CLA), Prefix Adders etc. have logarithmic delays. These bounds indicate that no reliable adder can be implemented with sub-logarithmic delay; however, unreliable adders can be implemented with sub-logarithmic delay. Unreliable adders could, for example, be used in the domain of cryptographic attacks; alternatively, reliable adders could be constructed from unreliable adders by augmenting them with additional circuitry for error detection and correction.

[5]. A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proceedings, Design Automation Conference. (DAC), 2012, pp. 820-825.

Guardbands for dynamic variations severely limit performance and energy efficiency of conventional IC designs. To overcome consequences of overdesign, several recent mechanisms+ for variation-resilient design allow timing errors and manage design reliability dynamically. Relaxing the requirement of correctness for designs may dramatically reduce costs of manufacturing, verification and test. In resilient designs, errors can be corrected with redundancy techniques (error-tolerance), or accepted in some applications relating to human senses such as hearing and sight (error acceptance). In the error-acceptance regime, approximation via a simplified or inaccurate circuit can increase performance and/or reduce power consumption.

[6]. M. Olivieri, "Design of synchronous and asynchronous variable-latency Pipelined multipliers," IEEE Transaction Very Large Scale Integration (VLSI) Systems, vol. 9, no. 2, pp. 365-376, Apr. 2001

For the high speed multipliers which is used in DSP processors and advanced microcontrollers, the CMOS architectures are used in order to obtain such an accurate arithmetic manipulations. In micro architectures with CMOS implementation 2ns has proved. In full custom CMOS multiplier design < 3ns is possible.

Pipelined multipliers are impact on various factors like registers, propagation delay and it is limited by certain times and leads to slow down the performance of multiplier process. In order to increase the throughput of pipelined arithmetic adaptive latency logic has been proposed. The interfacing is the basic problem between the synchronous and asynchronous unit because of signal handoff. High

speed VLSI adder has been proposed in previous work. But still there is an issue on adaptive latency between synchronous and asynchronous interfacing. Asynchronous (i.e., UN clocked) design but not adaptive latency, [to reduce power consumption was addressed earlier but not for high speed. Simple blocks of Array Multiplier The aim of the paper are to produce fastest multiplier design with accuracy. The proposed design deals with pipelined multiplier architecture, which combines various algorithm and design VLSI implementation for synchronous adaptive latency multiplier.

3. PROPOSED METHOD

A. *Speculative Carry Select Adder*

Carry chain in the addition process is observed for the design of speculative carry select addition (SCSA). The carry chain is observed because the long carry chain is rarely activated in the block adders. To overcome this problem, in SCSA the input bits are divided into two parts of equal sizes. A group of consecutive input bits are given as a input to a single block adder. A block adder is known as the window, the number of consecutive input bits is known as window size and it is denoted by K. number of windows to be used is found by M=N/K, where, n is the total number of input bits. The speculative sum bit is calculated by the (couti*) carry out bit of the window. By this prediction technique, the addition process is faster but the possibility of occurrence of error is increased. The proposed speculative adder can be able to complete the addition process quickly when compared to the existing technique.

B. *Carry Speculative Adder*

Speculative addition is widely used in asynchronous design. The speculative addition involves two cycles. In the first cycle, the addition process is done and the end result is assumed as accurate sum. Meanwhile, a parallel carry propagation circuit checks whether the operation uses the carry long path known as the critical path. If it uses the longest path the system requires the additional clock cycle to complete the addition process. If it didn't use the longest path, bypass logic is used to reduce the clock cycle required. The block diagram of CSPA is shown in Figure 1. Assume the a, b are the two inputs then the (P/G) signal known as (PROPAGATE/GENERATE) is defined as

Pj=a^bj

Gi=aibi

Then the sum and carry can be written as

$$Si = Pi-\hat{}C_{i-1}$$

$$Ci = Gi + PiC_{i-1}$$

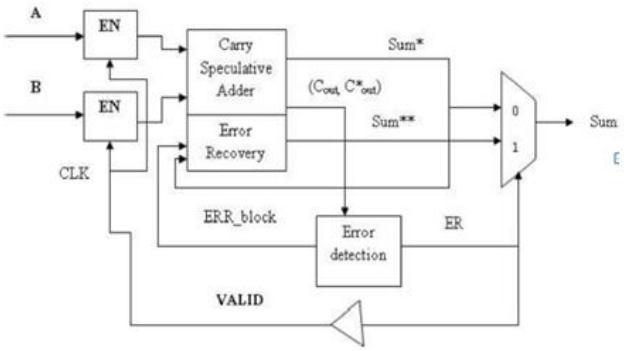


Fig 1: Block diagram of carry speculative adder

The carry speculative adder uses bits close to the MSB to predict the carry. By doing so, the time consumption for addition process is reduced.

C. *Error Detection And Recovery*

The carry prediction circuit may cause errors while predicting the carry. In the existing speculative adder there is no error detection and error correction circuit. In this carry speculative adder, it has an additional block called error detection and error recovery. In error detection circuit it uses EX-OR operation to find the error.

Error Analysis

There are two possible cases of errors.

• *Case (i)*

pix -l:x-k= I and Gi x-k-l:O = I,

• *Case (ii)*

Px-I:O= I and Gi -I

x-I:x-k= I,

In case (i), pix -I:x-k = 1 says that Gx-l:x-k=O. Cut of the ith block adder is recognized as incorrect. The carry out bit is corrected by using the following equation,

$$C^{i}_{out} = G^{i}_{x-1:x-k} + P^{i}_{x-1:x-k}G^{i}_{x-k-1:0} + P^{i}_{x-k-1:0}C^{i-1}_{out}$$

In case (ii), pix -k-I:O = I says that

Gi x-k-I:O = O.

This shows that the ith block adder output is O. The correct carry out bit is calculated using the following equation

$$C^{i}_{out} = G^{i}_{x-1:0} + P^{i}_{x-1:0}C^{i-1}_{out}$$

The major advantage is that the error detection circuit can find which block adder prediction is wrong. By this advantage the work of the recovery circuit is simplified. The recovery circuit rectifies the affected block adder and corrects the output so that the output sum is accurate. This circuit is designed using variable latency design, so that if an addition process is completed it send a valid signal to the input side to fetch another set of inputs to perform addition. This will greatly reduce the time consumption.

D. *Radix-4 Booth Multiplier*

To avoid the problems in Radix -2 algorithm, realization of high speed multipliers is needed. One of the solutions of realizing high speed multipliers is to enhance parallelism which helps to decrease the number of subsequent calculation stages. The original version of the Booth

algorithm (Radix-2) had two drawbacks. The number of add subtract operations and the number of shift operations are variable and become inconvenient in designing parallel multipliers. The algorithm becomes inefficient when there are isolated 1's. These are overcome by using modified Radix-4 Booth multiplication algorithm.

This algorithm scans strings of three bits at a time as follows:

1) Extend the sign bit 1 position if necessary to ensure that n is even.

2) Append a 0 to the right of the LSB of the multiplier.

3) According to the value of each vector, each Partial Product will be 0, +y, -y, +2y or -2y.

The multiplicand encoding process using Radix -4 Booth algorithm is based on the multiplier bits. It will compare 3 bits at a time with overlapping technique. Grouping starts from the LSB, and the first block uses only two bits of the multiplier and assumes a zero for the third bit. It consists of eight different types of states as we are comparing 3bits at a time and during these states we can obtain the outcomes, which are multiplication of multiplicand with 0,-1 and -2 consecutively. The state diagram presents various logics to perform the Radix-4 Booth multiplication in different states as per the adopting encoding technique.

Table 1: Radix-4 Booth Encoding Table

Block	Partial Product
000	0
001	1*multiplicand
010	1*multiplicand
011	2*multiplicand
011	2*multiplicand
100	-2*multiplicand
101	-1*multiplicand
110	-1*multiplicand
111	0

At last the addition of partial products is done by tree type hybrid adder. the simulation result of Radix-4 Booth multiplier in which they are two binary inputs as input, one is multiplicand and another one is multiplier. If both binary numbers are positive then it will perform booth encoding. Typically 3X is computed as the addition of 2X+X and −3X as its negation, increasing the critical path of the multiplier. we leverage the existence of slack cycles in the data path to compute these 3X multiples. In this way we can take advantage of the larger height reduction produced by the radix-8 recoding instead of the radix 4-based one.

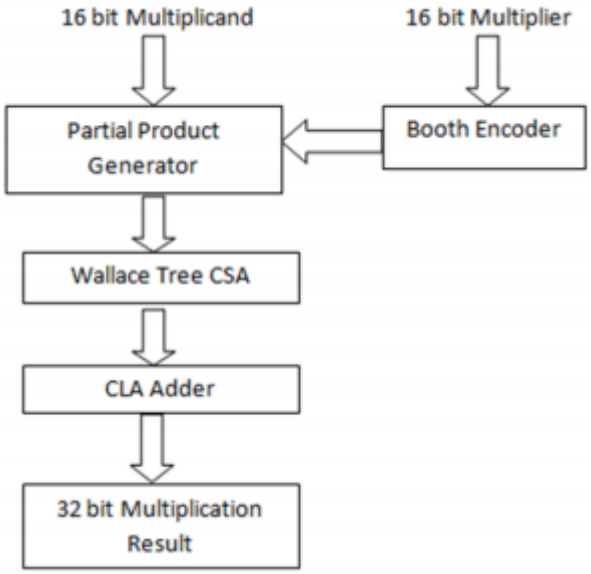
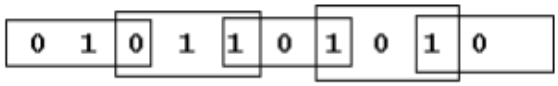


Fig 2: Block diagram for Booth Multiplier

Radix-4 Booth Recoding to Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit.

In place of carry look ahead adder we are using carry speculative adder.



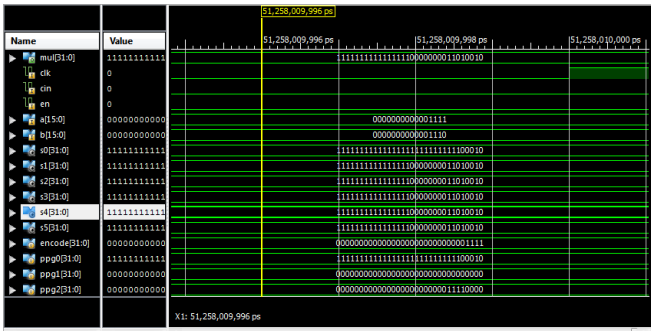
Grouping starts from the LSB, and the first block only uses two bits of the multiplier (since there is no previous block to overlap): Fig: Grouping of bits from the multiplier term, for use in Booth recoding.

The least significant block uses only two bits of the multiplier, and assumes a zero for the third bit. The overlap is necessary so that we know what happened in the last block, as the MSB of the block acts like a sign bit. We then consult the table 2-3 to decide what the encoding will be.

4. SIMULATION RESULTS

Every block in the carry speculative adder is designed and simulated using Xilinx ISE. Here the existing and the proposed method are compared in terms of area, power and delay.

Simulation results for Booth Multiplier



Power Consumption

Device	
Family	Virtex5
Part	xc5vlx20t
Package	ff323
Temp Grade	Commercial
Process	Typical
Speed Grade	-2

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA
Board Temperature (C)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.005	1	--	--
Logic	0.000	752	12480	6
Signals	0.000	767	--	--
IOs	0.000	66	172	38
Leakage	0.321			
Total	0.326			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	2.8	84.1	50.9

Supply Summary	Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.241	0.005	0.236
Vccaux	2.500	0.032	0.000	0.032
Vcco25	2.500	0.002	0.000	0.002

Supply Power (W)	Total	Dynamic	Quiescent
	0.326	0.005	0.321

Delay

```
Delay:               5.147ns (Levels of Logic = 5)
  Source:            b<0> (PAD)
  Destination:       c2<0> (PAD)

  Data Path: b<0> to c2<0>
                                Gate     Net
    Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             3    0.694    0.609  b_0_IBUF (b_0_IBUF)
     LUT3:I0->O            2    0.086    0.666  g4/Mxor_s_Result<1>11 (N11)
     LUT5:I1->O            2    0.086    0.491  g4/C_3_or00001 (g4/C<3>)
     LUT3:I1->O            1    0.086    0.286  g4/cout1 (c2_0_OBUF)
     OBUF:I->O                  2.144           c2_0_OBUF (c2<0>)
    ----------------------------------------
    Total                      5.147ns (3.096ns logic, 2.051ns route)
                                       (60.1% logic, 39.9% route)
```

Area Used

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice LUTs	17	12,480	1%	
Number used as logic	17	12,480	1%	
Number using O6 output only	6			
Number using O5 and O6	11			
Number of occupied Slices	13	3,120	1%	
Number of LUT Flip Flop pairs used	17			
Number with an unused Flip Flop	17	17	100%	
Number with an unused LUT	0	17	0%	
Number of fully used LUT-FF pairs	0	17	0%	
Number of slice register sites lost to control set restrictions	0	12,480	0%	
Number of bonded IOBs	56	172	32%	
Average Fanout of Non-Clock Nets	1.74			

5. CONCLUSION

A variable latency adder that combines the speculative adder with error recognition and correction for unsigned random inputs, called variable latency carry speculation adder. The sum and carry generator are separated in CSP A and thus the carry signal and partial sum bit can be calculated faster. Carry predictor circuit of the block adder only uses the input bits near the MSB to predict the carryout bit. The hardware cost of the prediction circuit is reduced and the CSP A has minimal error rate increase. The proposed error detection circuit indicates which block adder produced an incorrect carry-out bit, and the error recovery circuit only focuses on recovering the block adders with incorrect partial sum bits. On comparing CSP A and SCSA, CSPA reduces delay and also reduces computational complexity.

REFERENCES

[I] D. Mohapatra, A. Raghunathan, V. Gupta and K. Roy, "Low-power digital signal processing using approximate adders, " IEEE Transaction Computer- Aided Design Integration Circuits Systems., vol. 32, no. I, pp. 124-137, Jan. 2013.

[2] N. Zhu, K. S. Yeo, W. L. Goh, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," IEEE Transaction Very Large Scale Integration (VLSI) Syst., vol. 18, no. 8, pp. 1225-1229, Aug. 2010.

[3] P. Varman, K. Du and K. Mohanram, "High performance reliable variable latency carry select addition," in Proceedings, Design Automation Test Conference Exhibition (DATE), Mar. 2012, pp. 1257-1262.

[4] Y. Chen et aI, "Variable-latency adder (VL-adder) designs for low power and NBTI tolerance," IEEE Transaction Very Large Scale Integration (VLSI) System, vol. 18, no. II, pp. 1621-1624, Nov. 2010.

[5] Y.-H. Su, D.-C. Wang, S.-c. Chang, and M.-S. Malgorzata, "Performance optimization using variable-latency design style," IEEE Transaction Very Large Scale Integration (VLSI) Systems, vol. 19, no. 10, pp. 1874-1883, Oct. 2011.

[6] D. Baneres, 1. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in Proceedings, Design Automation Test Conference Exhibition (DATE), Apr. 2009, pp. 1704-1709.

[7] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proceedings, Design Automation Test Conference Exhibition. (DATE), Mar. 2010, pp. 957-960.

[8] N. Zhu, W. L. Goh, and K. S. Yeo, "Ultra low-power high-speed flexible probabilistic adder for error-tolerant applications," in Proceedings Integration SoC Design Conference (ISOCC), Nov. 2011, pp. 393-396.

[9] A. K. Verma, P. Brisk, and P. lenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proceedings, Design Automation Test Conference Exhibition (DATE), Mar. 2008, pp. 1250-1255.

[10] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proceedings, Design Automation Conference. (DAC), 2012, pp. 820-825.

[11] Y. Liu, Y. Sun, Y. Zhu, and H. Yang, "Design methodology of variable latency adders with multistage function speculation," in Proc. 11th International Symposium Qual Electron. Design (ISQED), Mar. 2010, pp. 824-830.

[12] M. Olivieri, "Design of synchronous and asynchronous variable-latency pipelined multipliers," IEEE Transaction Very Large Scale Integration (VLSf) Systems., vol. 9, no. 2, pp. 365-376, Apr. 2001.

[13] Y. Chen, H. Li, J. Li, and c.-K. Koh, "Variable-latency adder (VL adder): New arithmetic circuit design practice to overcome NBTI," in Proceedings International. Symposium Low Power Electronic Design (ISLPED), Aug. 2007, pp. 195-200.

[14] Synopsys Design Compiler. (2010) [Online]. Available: http://www.synopsys.com.

ABOUT AUTHORS

Mr.R.Mallikarjuna Reddy received B.Tech degree from Department of Electronics & Communication Engineering from AITS, Rajampeta, Kadapa (Dt), Andhra pradesh and M.E Degree from Sathya bama University, Chennai, Tamil Nadu.

Currently working as Associate Professor in Department of ECE in VEMU Institute of Technology, P.Kothakota, Chittoor Dist, A.P, and India. His intersted areas are ES, DIP and Communications etc.