IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p-ISSN: 2278-8735
PP 53-56
www.iosrjournals.org

Implementation of Power Efficient Approximate Multiplier

J.Anjana, A.Chitra, K.Shanthalakshmi
Department of ECE, Adhiyamaan College of Engineering, Hosur, India,
Department of ECE, Adhiyamaan College of Engineering, Hosur, India,
Asst Proffessor, Departmet of ECE, Adhiyamaan College of Engineering, Hosur, India,

Abstract: *Low power is the most important requirement for multimedia devices including various digital signal processing algorithms. In most multimedia signal processing, the final output is disturbed by human senses, which are not accurate. This is the reason for the need to produce exactly correct outputs. Previous study in this process utilizes error reduction through voltage over scaling, utilizing algorithmic and architectural techniques. In this paper, we propose logic complexity reduction by varying partial products to improve the accuracy. We reveal this concept by proposing various type approximate half , full adder and compressor with reduced number of the transistor, and then use in the process of designing approximate multipliers. In addition to this our process usually results in significantly low power dissipation and reduced actual time. We design architectures for both compressions of video and image algorithms using the proposed approximate multipliers. This simulations results a large power savings and area savings with a very little loss in output quality of image, when it is compared to existing system.*

Key Terms: *Approximate multiplier, partial products, Low-power.*

I. Introduction

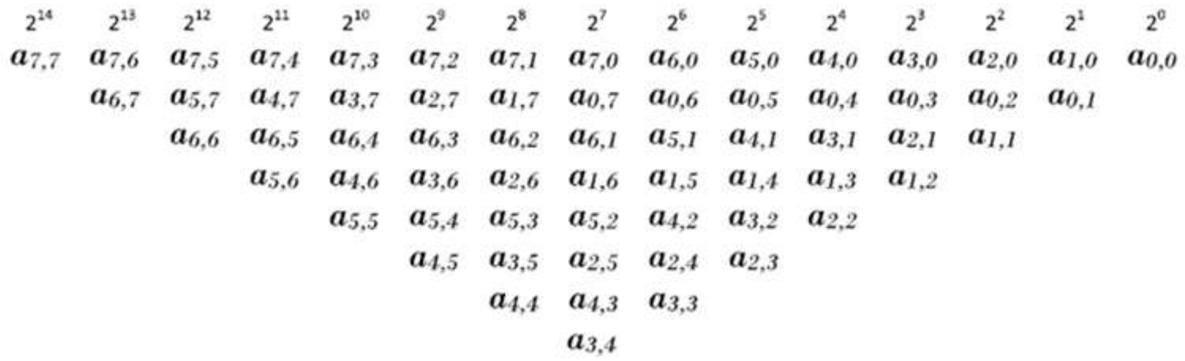
The idea for modern VLSI design and digital Embedded Systems is the ability of VLSI design and embedded systems to create an impact to the environment[1][2]. The purpose for creating power efficient VLSI architectures in the approximate multipliers is not only to reduce the power dissipation but also to improve the approximation value[3].

In order to improve power efficiency of such computing devices, many attempts have been recommended at various levels, from simulation to architecture, and all the way down to hardware circuit and technology[4]. Embedded system and all the mobile computing devices are habitually required to execute some key elements in the digital signal processing (DSP) applications. Further some improvement in the power efficiency for executing such applications can be made by integrated devices [5]. It has been observed that the use of these specialized processors can improve power efficiency compared with general processors at the same voltage and technology generation [6].Second, many DSP system and classification applications heavily depend on complex mathematical models and are designed to process information that typically contains noise[7]. Thus, for reducing the computational error, they exhibit only slight change in the overall DSP quality and classification precision instead of a severe failure in the multipliers. Such type of the error tolerance has been created by trading accuracy with power consumption [8]. Finally, these algorithms are initially designed and trained with partial products, but they are often converted to fixed point arithmetic due to the area, time and power cost of supporting units in embedded computing devices [9]. Even though this conversion process leads to some loss of computational precision, it usually does not affect the quality of DSP, images and the accuracy of classification applications due to reduced approximate difference [10].

The rest of this paper is organized as follows. Section II details the various related works. Section III gives a clear description about the proposed power efficient approximate multiplication technique. Section IV explains about the simulation and synthesis results. Finally, Section V details the conclusion.

II. Related Works

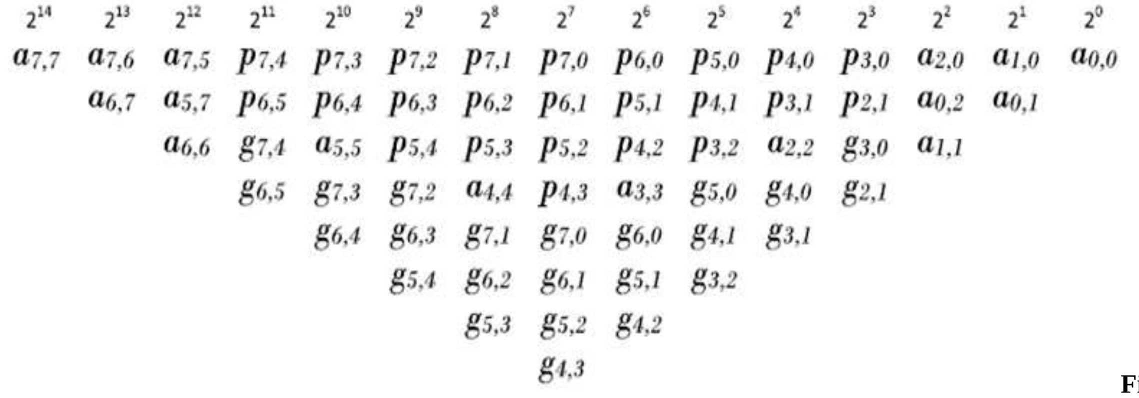
Approximate computing of the partial products is an emerging design paradigm that creates highly efficient hardware and software implementations by reducing the inherent resilience of applications to in accuracy in their computations[11][12]. Several efforts have been done for approximate computing in both the software and hardware with exact results. Software techniques always improves the performance by reducing the computations or the use of costly operations is avoided whereas some hardware techniques change the design at various levels of partial products to introduce change between output quality and power efficiency [13]. These efforts have been established for the significant approximate computing in the multipliers [14]. The figure 1 shows the generation of the partial products in the multipliers.



2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$a_{7,7}$	$a_{7,6}$	$a_{7,5}$	$a_{7,4}$	$a_{7,3}$	$a_{7,2}$	$a_{7,1}$	$a_{7,0}$	$a_{6,0}$	$a_{5,0}$	$a_{4,0}$	$a_{3,0}$	$a_{2,0}$	$a_{1,0}$	$a_{0,0}$
	$a_{6,7}$	$a_{5,7}$	$a_{4,7}$	$a_{3,7}$	$a_{2,7}$	$a_{1,7}$	$a_{0,7}$	$a_{0,6}$	$a_{0,5}$	$a_{0,4}$	$a_{0,3}$	$a_{0,2}$	$a_{0,1}$	
		$a_{6,6}$	$a_{6,5}$	$a_{6,4}$	$a_{6,3}$	$a_{6,2}$	$a_{6,1}$	$a_{5,1}$	$a_{4,1}$	$a_{3,1}$	$a_{2,1}$	$a_{1,1}$		
			$a_{5,6}$	$a_{4,6}$	$a_{3,6}$	$a_{2,6}$	$a_{1,6}$	$a_{1,5}$	$a_{1,4}$	$a_{1,3}$	$a_{1,2}$			
				$a_{5,5}$	$a_{5,4}$	$a_{5,3}$	$a_{5,2}$	$a_{4,2}$	$a_{3,2}$	$a_{2,2}$				
					$a_{4,5}$	$a_{3,5}$	$a_{2,5}$	$a_{2,4}$	$a_{2,3}$					
						$a_{4,4}$	$a_{4,3}$	$a_{3,3}$						
							$a_{3,4}$							

Fig 1: Generation of partial products

Approximate computing have been considered for reducing the error in all the applications that can resilience some loss of accuracy in the image and video with improved performance and power efficiency [14]. Applications including image processing, image recognition, multimedia signal processing and data mining can tolerant some error and does not require perfect output in computation[15]. For these types of applications, approximate multipliers may play an important role as a best alternative for reducing power, area and energy in digital systems that can tolerate some loss of accuracy in the output, so that it is possible to achieve best performance in power efficiency [16]. Such type of process exhibits the property of high inherent algorithmic tolerance to errors[17]. These algorithms are used to process large amounts of input data for multipliers that has significant discharge and may frequently contain imperfections in the output [18]. The fig 2 shows the conversion of partial products.

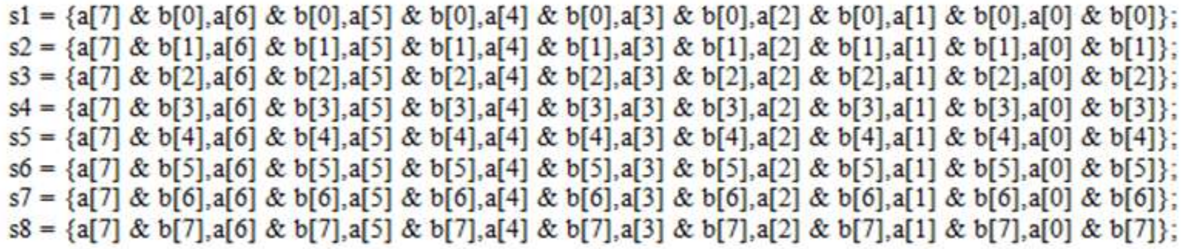


2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
$a_{7,7}$	$a_{7,6}$	$a_{7,5}$	$p_{7,4}$	$p_{7,3}$	$p_{7,2}$	$p_{7,1}$	$p_{7,0}$	$p_{6,0}$	$p_{5,0}$	$p_{4,0}$	$p_{3,0}$	$a_{2,0}$	$a_{1,0}$	$a_{0,0}$
	$a_{6,7}$	$a_{5,7}$	$p_{6,5}$	$p_{6,4}$	$p_{6,3}$	$p_{6,2}$	$p_{6,1}$	$p_{5,1}$	$p_{4,1}$	$p_{3,1}$	$p_{2,1}$	$a_{0,2}$	$a_{0,1}$	
		$a_{6,6}$	$g_{7,4}$	$a_{5,5}$	$p_{5,4}$	$p_{5,3}$	$p_{5,2}$	$p_{4,2}$	$p_{3,2}$	$a_{2,2}$	$g_{3,0}$	$a_{1,1}$		
			$g_{6,5}$	$g_{7,3}$	$g_{7,2}$	$a_{4,4}$	$p_{4,3}$	$a_{3,3}$	$g_{5,0}$	$g_{4,0}$	$g_{2,1}$			
				$g_{6,4}$	$g_{6,3}$	$g_{7,1}$	$g_{7,0}$	$g_{6,0}$	$g_{4,1}$	$g_{3,1}$				
					$g_{5,4}$	$g_{6,2}$	$g_{6,1}$	$g_{5,1}$	$g_{3,2}$					
						$g_{5,3}$	$g_{5,2}$	$g_{4,2}$						
							$g_{4,3}$							

Fig 2: Conversion of partial products.

III. Proposed Method

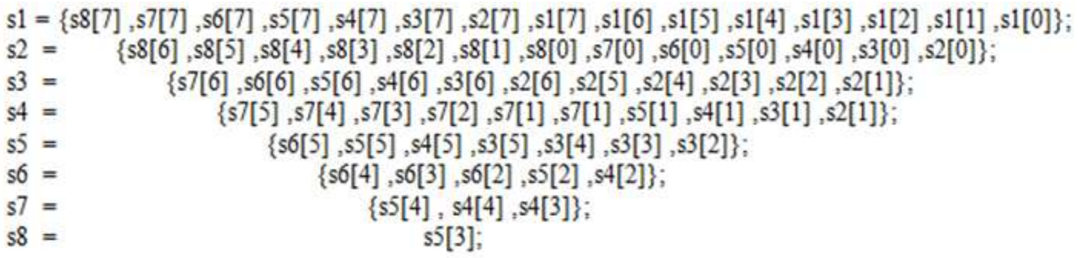
In the proposed multipliers we consider two 8-bit operands a7,a6.......*a2a1a0* and *b7,b6.….b2b1b0 for DADDA* multiplier, the partial products of two 8-bit numbers are *aibj* where *i,j* go from 0,1,..7. The partial products form a matrix of 0 rows and 7 columns as show in Fig.3.



```
s1 = {a[7] & b[0],a[6] & b[0],a[5] & b[0],a[4] & b[0],a[3] & b[0],a[2] & b[0],a[1] & b[0],a[0] & b[0]};
s2 = {a[7] & b[1],a[6] & b[1],a[5] & b[1],a[4] & b[1],a[3] & b[1],a[2] & b[1],a[1] & b[1],a[0] & b[1]};
s3 = {a[7] & b[2],a[6] & b[2],a[5] & b[2],a[4] & b[2],a[3] & b[2],a[2] & b[2],a[1] & b[2],a[0] & b[2]};
s4 = {a[7] & b[3],a[6] & b[3],a[5] & b[3],a[4] & b[3],a[3] & b[3],a[2] & b[3],a[1] & b[3],a[0] & b[3]};
s5 = {a[7] & b[4],a[6] & b[4],a[5] & b[4],a[4] & b[4],a[3] & b[4],a[2] & b[4],a[1] & b[4],a[0] & b[4]};
s6 = {a[7] & b[5],a[6] & b[5],a[5] & b[5],a[4] & b[5],a[3] & b[5],a[2] & b[5],a[1] & b[5],a[0] & b[5]};
s7 = {a[7] & b[6],a[6] & b[6],a[5] & b[6],a[4] & b[6],a[3] & b[6],a[2] & b[6],a[1] & b[6],a[0] & b[6]};
s8 = {a[7] & b[7],a[6] & b[7],a[5] & b[7],a[4] & b[7],a[3] & b[7],a[2] & b[7],a[1] & b[7],a[0] & b[7]};
```

Fig 3: Generation of partial products of the proposed multipliers.

The next is the partial products reduction in each part. The generated partial products is reduced using some basic rules of DADDA multipliers. The final two rows of each part in the reduction of partial products are added using a Carry save Adder. The fig 4 shows the reduction of partial products.



```
s1 = {s8[7] ,s7[7] ,s6[7] ,s5[7] ,s4[7] ,s3[7] ,s2[7] ,s1[7] ,s1[6] ,s1[5] ,s1[4] ,s1[3] ,s1[2] ,s1[1] ,s1[0]};
s2 =       {s8[6] ,s8[5] ,s8[4] ,s8[3] ,s8[2] ,s8[1] ,s8[0] ,s7[0] ,s6[0] ,s5[0] ,s4[0] ,s3[0] ,s2[0]};
s3 =             {s7[6] ,s6[6] ,s5[6] ,s4[6] ,s3[6] ,s2[6] ,s2[5] ,s2[4] ,s2[3] ,s2[2] ,s2[1]};
s4 =                   {s7[5] ,s7[4] ,s7[3] ,s7[2] ,s7[1] ,s7[1] ,s5[1] ,s4[1] ,s3[1] ,s2[1]};
s5 =                         {s6[5] ,s5[5] ,s4[5] ,s3[5] ,s3[4] ,s3[3] ,s3[2]};
s6 =                               {s6[4] ,s6[3] ,s6[2] ,s5[2] ,s4[2]};
s7 =                                     {s5[4] , s4[4] ,s4[3]};
s8 =                                            s5[3];
```

Fig 4: Reduction of partial products.

After the process of reduction of the partial products we use all the approximate half adders, full adders and 4-2 compressors for further reduction of the partial products. Then the output of the approximate circuits is given to the carry save adder which produces the final value of 8x8 multiplications. Carry save adder reduces delay when compared with the ripple carry adder so we make use of the carry save adder.

IV. Results And Discussion.

All the generated partial products usually make use of the approximate half adders, full adders, 4-2 compressors to produce the sum and carry row. The various generated sum and carry rows are given as the input to the carry save adder. The fig 5 shows the simulation results of the carry save adder.

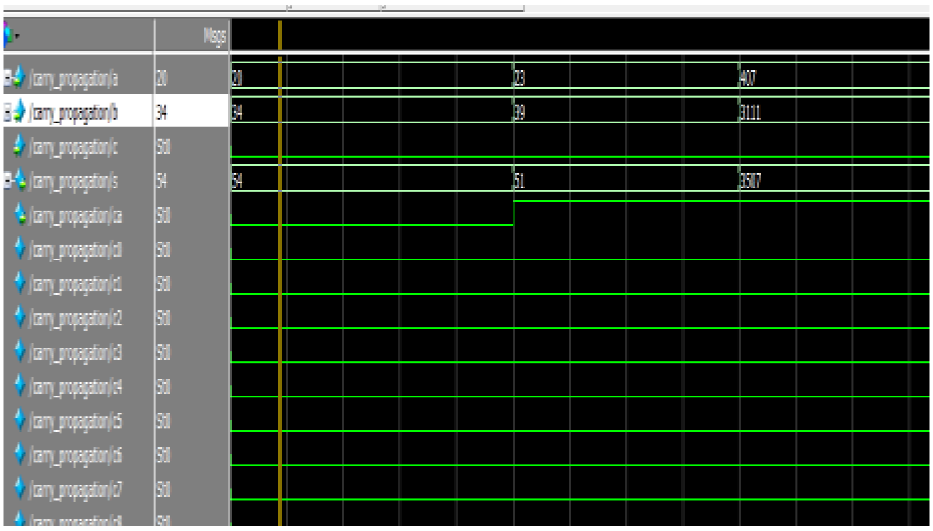
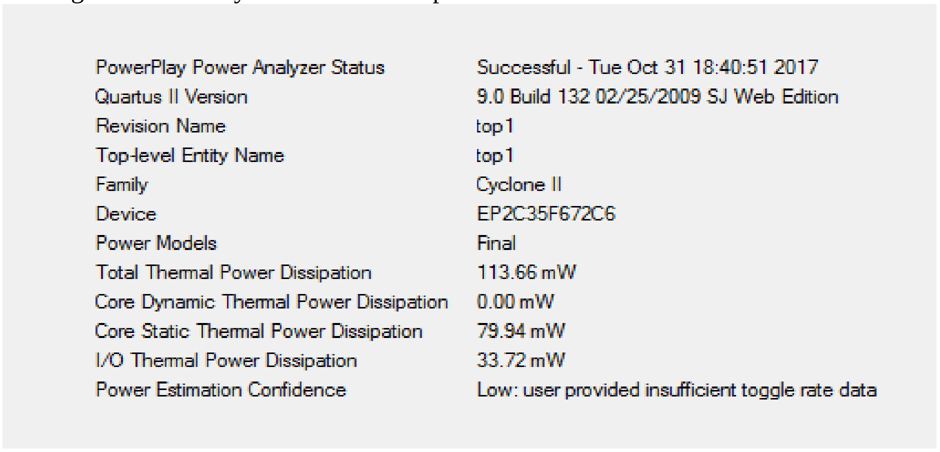


Fig 5: Simulation results of carry save adder.

So finally we can easily find the value of the approximate multiplication using the DADDA multipliers. Then we need to synthesis the final results in order to calculate the amount of the power dissipation. After the process of synthesis we found that the approximation reduces the power consumption from 135 mW to 113.66 mW. The fig 6 shows the synthesis results of power.



PowerPlay Power Analyzer Status	Successful - Tue Oct 31 18:40:51 2017
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	top1
Top-level Entity Name	top1
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	113.66 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	79.94 mW
I/O Thermal Power Dissipation	33.72 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Fig 6: Synthesis results of power

V. Conclusion

In this paper, to propose power efficient approximate multipliers, first the partial products of the multiplier are changed using generate and propagate signals in the initial stage. Approximation of the multipliers is applied using simple OR gate for changed generate partial products. Approximate half-adder, approximate full-adder, and approximate 4-2 compressor are designed to reduce remaining partial products. They are also found to have better precision when it is compared to existing approximate multiplier. The proposed approximate multiplier designs can be used in applications like image processing with little loss in output quality of the image while reducing power and area. The proposed approximate multiplier design technique can be used for any type of parallel multipliers to improve the speed and performance.

References

[1]. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digitalsignal processing using approximate adders," *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137,Jan. 2013.

[2]. E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncationscheme for parallel multipliers," in *Proc. 31st Asilomar Conf. Signals, Circuits Syst.*, Nov. 1998, pp. 1178–1182.

[3]. K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design oflow-error fixed-width modified booth multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 522–531,May 2004.

[4]. H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[5]. A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[6]. P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *Proc. 24th Annu. Conf.VLSI Design, Jan. 2011, pp. 346–351.*

[7]. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., Vol. 23, No. 6, pp. 1180–1184, Jun. 2015.*

[8]. C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: A tool for evaluating and synthesizing multimedia and communications systems," in *Proc. 13th Annu. IEEE/ACM Int. Symp. Microarchitecture, Dec. 1997, pp. 330–335.*

[9]. G. Zervakis, K. Tsoumanis, S. Xydis, N. Axelos, and K. Pekmestzi, "Approximate multiplier architectures through partial product perforation: Power-area tradeoffs analysis," in *Proc. 25th Great Lakes Symp.VLSI, 2015, pp. 229–232.*

[10]. D. Soudris, C. Piguet, and C. Goutis, *Designing CMOS Circuits forLow Power, ser. European low-power initiative for electronic systemdesign. Springer, 2002.*

[11]. R. Zimmermann and W. Fichtner, "Low-power logic styles: Cmos versus pass-transistor logic," *Solid-State Circuits, IEEE Journal of, vol. 32, no. 7, pp. 1079–1090, Jul 1997.*

[12]. Srinivasan Narayanamoorthy, Hadi Asghari Moghaddam, Zhenhong Liu, Taejoon Park, and Nam Sung Kim, July 2014, "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Early Access Articles, pp. 1 – 5.

[13]. Zdenek Vasicek and Lukas Sekanina, April 2014, "Evolutionary Design of Approximate Multipliers under Different Error Metrics", Design and Diagnostics of Electronic Circuits & Systems, 17th International IEEE Symposium, pp. 135 – 140.

[14]. K. S. Ganesh Kumar, J. Deva Prasannam, M. Anitha Christy, March 2014, "Analysis of Low Power, Area and High Performance Multipliers for DSP applications", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 3,pp.278-382.

[15]. Vinay K. Chippa, Srimat T. Chakradhar, Kaushik Roy and Anand Raghunathan, May- June 2013, "Analysis and Characterization of Inherent Application Resilience for Approximate Computing", IEEE Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE, pp. 1- 9.

[16]. Jie Han, Michael Orshansky, May 2013, "Approximate Computing: An emerging Paradigm for Energy-Efficient Design", Test Symposium (ETS), 2013 18th IEEE European, pp. 1-6.

[17]. A.Kishore Kumar, D. Somasundareswari, V. Duraisamy and T. Shunbaga Pradeepa, February 2013 , "Design of Low Power Multiplier with Energy Efficient Full Adder Using DPTAAL" , Hindawi Publishing Corporation,VLSI Design, Volume 2013, pp. 1-9.

[18]. V. K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S. T. Chakradhar, June 2010, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency," in Proc. 47th IEEE/ACM Design Autom. Conf., pp. 555–560.

[19]. Sjalander.M, Larsson-Edefors,P., Aug. 31 2008-Sept. 3 2008, "High-speed and low-power multipliers using the Baugh-Wooley algorithm and HPM reduction tree", Electronics, Circuits and Systems, 2008, ICECS-2008. 15th IEEE International Conference, pp. 33 – 36.