International Journal of VLSI System Design and Communication Systems

ISSN 2322-0929
Vol.02, Issue.09,
November-2014,
Pages:0824-0829

www.ijvdcs.org

Design and Simulation of Various Approximation Adders to Reduce the Power Consumption

J.PRAMEELA[1], M.S.SHAFEE UNNISA SAYYED[2], DR.S.P.V.SUBBA RAO[3]

[1]PG Scholar, Dept of DSCE, Sreenidhi Institute of Science & Technology, Hyderabad, India.
[2]Assoc Prof, Dept of ECE, Sreenidhi Institute of Science & Technology, Hyderabad, India.
[3]Assoc Prof , Dept of ECE, Sreenidhi Institute of Science & Technology, Hyderabad, India.

Abstract: Power dissipation has become a significant issue for integrated circuit design in nanometric CMOS technology. To reduce power consumption, approximate implementations of a circuit have been considered as a potential solution for applications in which strict exactness is not required. In inexact computing, power reduction is achieved through the relaxation of the often demanding requirement of accuracy. Earlier research exploits error resiliency primarily through voltage over scaling, using algorithmic and architectural techniques to mitigate the resulting errors. In our paper, we propose logic complexity reduction at the transistor level as the alternative approach to take advantage of the relaxation of numerical accuracy. We examined this concept by proposing several imprecise or approximate full adder cell with reduced complexity at the transistor level, and used them to design approximate full adders. Simulation outputs indicate up to 69% power savings using the proposed approximate adders, when compared to previous implementations using accurate adders.

Keywords: VLSI, CMOS Technology, Adders, Power Consumption.

I. INTRODUCTION

Today there are an escalating number of portable applications with limited amount of power available, requiring small area, low-power and high throughput circuitry. Therefore circuits which consume low power become the major concern factor for design of microprocessors and system components. The research effort in low power microelectronics has been intensified and low power VLSI systems have emerged as exceedingly in demand. In highly integrated nanoscale designs, reliability issues resulting from PVT (process, voltage and temperature) variations, aging effects and soft errors have become major impediments for leveraging the benefits of a lower device scaling; moreover, leakage and static power are significant concerns for the high power consumption encountered at such high density. A potential solution to lower power dissipation is to employ approximate circuit designs [1]. Commonly used multimedia applications have digital signal processing (DSP) blocks as core. Most of these DSP blocks implement algorithms, in which the ultimate output is either an image or a video for human presentation and analysis. For example, the limited perception of human vision allows the outputs of these algorithms to be numerically approximate rather than accurate [2].

The relaxation on numerical exactness provides at least some freedom to perform imprecise or approximate computation. The development of imprecise, but simplified arithmetic units can provide an extra layer of power saving over conventional low-power design techniques such as using a lower supply voltage. As basic building blocks in many digital circuits, adders have been investigated for approximate implementations. This paper proposes the new 9 transistor approximate adders a reduction in logic complexity is accomplished at transistor level by removing some of the transistors required in the accurate adder design. Additionally, the node capacitances and thus dynamic power are reduced to lower the power/energy consumption of the proposed circuits. In this paper, delay, energy consumption, area and power-delay product are measured for comparing the different designs with an accurate adder. An example version of our work appeared in [3]. We extend our paper in [3] by giving two more simplified versions of the MA. We also introduced a methodology that can be used to harness maximum power savings using approximate adders, subject to a specific quality constraint. Our contributions in this paper are summarized as follows. To simplify the logic complexity of a conventional MA cell by reducing the number of transistors and switched capacitances. Keeping this aim in mind, we propose five various simplified versions of the MA, ensuring minimum errors in the full adder (FA) truth table.

A. Conventional Full Adder:

Adder is one of the most vital components of a CPU (central processing unit), Arithmetic logic unit (ALU), and floating point unit and address generation like cache or memory access unit. On the other hand, increasing demand for portable equipments Such as cellular phones, personal digital assistant (PDA), and Notebook personal computer, arise the need of using area and Power efficient VLSI

circuits. Low-power and high-speed adder cells are used in battery-operation based devices

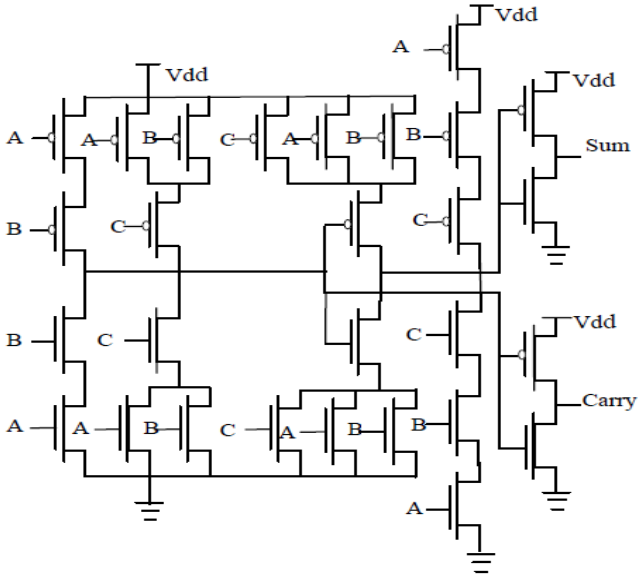

Fig1. Schematic of Conventional Full Adders

B. Approximate Full Adders

In several approximate implementations; multiple-bit adders are divided into two modules: the (accurate) upper part of more significant bits and the (approximate) lower part of less significant bits. For each lower bit, a single-bit approximate adder implements a modified, thus inexact function of the addition. This is often accomplished by simplifying a full adder design at the circuit level, equivalent to a process that alters some entries in the truth table of a full adder at the functional level.

C. Approximate mirror adders (AMAs)

A mirror adder (MA) is a common yet efficient adder design. Five approximate MAs (AMAs) have been obtained from a logic reduction at the transistor level, i.e., by removing some transistors to attain a lower power dissipation and circuit complexity. A faster charging/ discharging of the node capacitance in an AMA also incurs a shorter delay. Hence, the AMAs trade off accuracy for energy, area and performance.

D. Strategies for the Mirror Adder

The Strategies of Mirror Adders we describe step-by-step procedures for coming up with various approximate MA cells with fewer transistors. Cancellation of some series connected transistors will facilitate faster charging and discharging of node capacitances. Moreover, complexity reduction by removal of transistors also leads in reducing the αC term (switched capacitance) in the dynamic power expression $P_{dynamic} = \alpha CV^2DDf$, where α is a switching activity or average number of switching transitions per unit time and C is the load capacitance being charged/ discharged. This directly results in less power dissipation. Area reduction is also produced by this process. Now, let us focus the conventional MA implementation followed by the proposed approximations.

D. Conventional Mirror Adders

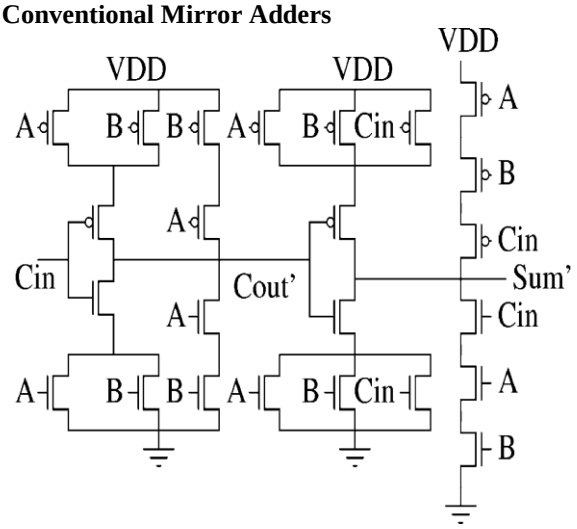

Fig2. Conventional Mirror Adders

Fig.1 shows the transistor-level schematic of a conventional MA, which is a famous way of implementing an FA. It contains a total of 24 transistors. Since this implementation is not based on complementary CMOS logic, it gives a good opportunity to design an approximate version with removal of selected transistors.

E. Mirror Approximation Adder 1

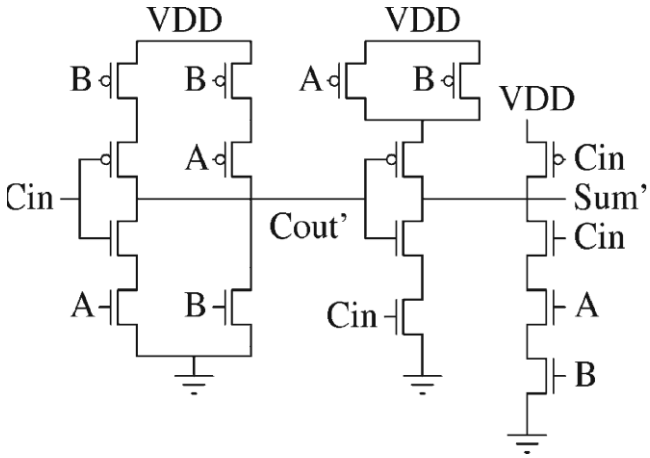

Fig3: Mirror Approximation adder

Fig3 shows the approximation Mirror adders. In order to get approximate MA with lesser transistors, we start to remove transistors from the conventional schematic one by one. However, we should not do this in an arbitrary fashion. We have to make sure that any input combination of A, B and Cin will not result in short circuits or open circuits in the simplified schematic. Another main criterion is that the resulting simplification should introduce minimal errors in the FA truth table.

F. Approximation Mirror Adder-2

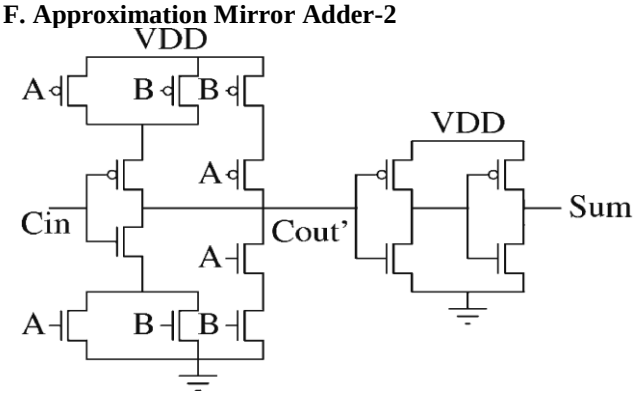

Fig4:Approximation Mirror Adder-2

The truth table of an FA indicates that Sum= Cout 1 for six out of eight cases, except for the input combinations A = 0, B = 0, Cin = 0 and A = 1, B = 1, Cin = 1. Now, in the conventional MA, $Cout$ is calculated in the first stage. Thus, an simple way to get a simplified schematic is to set Sum=$Cout$. However, we introduce a buffer stage after $Cout$ Fig. 4 to produce the same functionality.

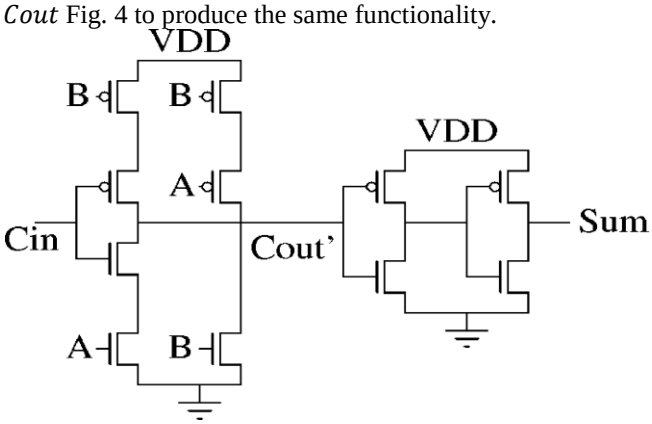

Fig5:Approximation Mirror Adder-3

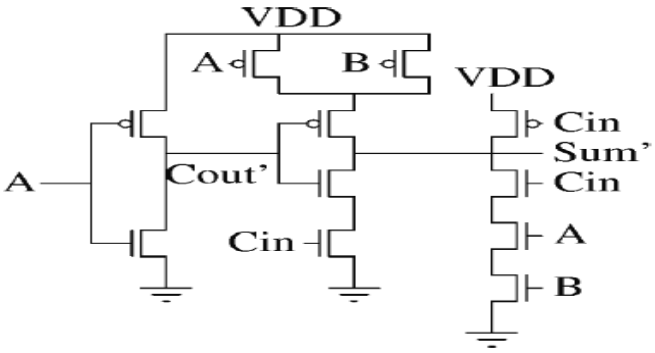

Fig6:Approximation Mirror Adder-4.

The schematic of 9T full adder cell is shown in Figure 7 and its truth table is given in Table 2. The principle of current circuit is differed from traditional circuits. The full adder operation can be given as follows. Given the three 1-bit inputs *A*, *B*, and Cin, it is desired to compute the two 1-bit outputs Sum and Cout, given by

Sum = *A XOR B XOR* Cin.

Cout = $A \cdot B$ + Cin (*A XOR B*).

For generating the Sum output in the proposed design, the truth table has been segmented into two parts, one for input *A* = "0" and another for *A* = "1" rather than implementing the conventional Sum module. From the truth table shown in Table 1 it is clear that when *A* = "0", Sum can be produced by XORing inputs *B* and Cin. Similarly, when *A* = "1", Sum focusing the XNORing between inputs *B* and Cin. Therefore, the operation of Sum module depends on implementing XOR operation and XNOR operation between inputs *B* and Cin.

Table I. Approximation Adders

The Truth Table of Approximation Adder is shown in the Table1

Inputs			Accurate Outputs		Approximate Outputs							
A	*B*	C_{in}	Sum	C_{out}	Sum_1	C_{out1}	Sum_2	C_{out2}	Sum_3	C_{out3}	Sum_4	C_{out4}
0	0	0	0	0	0✓	0✓	1×	0✓	1×	0✓	0✓	0✓
0	0	1	1	0	1✓	0✓	1✓	0✓	1✓	0✓	1✓	0✓
0	1	0	1	0	0×	1×	1✓	0✓	0×	1×	0×	0✓
0	1	1	0	1	0✓	1✓	0✓	1✓	0✓	1✓	1×	0×
1	0	0	1	0	0×	0✓	1✓	0✓	1✓	0✓	0×	1×
1	0	1	0	1	0✓	1✓	0✓	1✓	0✓	1✓	0✓	1✓
1	1	0	0	1	0✓	1✓	0✓	1✓	0✓	1✓	0✓	1✓
1	1	1	1	1	1✓	1✓	0×	1✓	0×	1✓	1✓	1✓

G. Proposed New 9 Transistor Full Adder

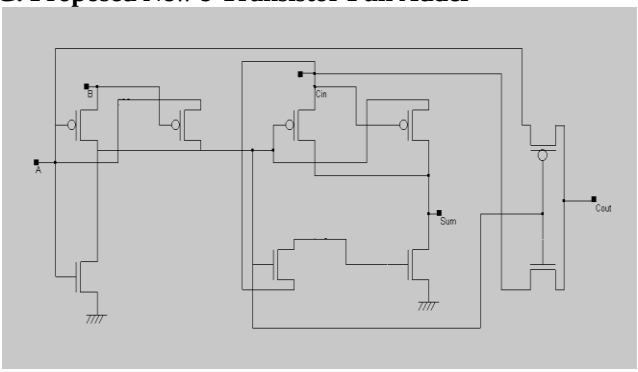

Fig7: 9 T Full Adder.

Table2. Truth Table of 9T Full Adder

Inputs			Outputs	
A	**B**	**C**	**Sum**	**Carry**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0**x**	1**x**
1	0	1	1**x**	1
1	1	0	0	1
1	1	1	1	1

H. Parallel Adders:

Parallel adders are digital circuits that compute the addition of variable binary strings of equivalent or different size in parallel. The ripple carry adder is constructed by

cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. A number of full adders may be added to the ripple carry adder or ripple carry adders of different sizes may be cascaded in order to accommodate binary vector strings of larger sizes. For an n-bit parallel adder, it requires n computational elements (FA). Figure 4 shows an example of a parallel adder: a 4-bit ripple-carry adder. It is composed of four full adders. The augend's bits of x are added to the addend bits of y respectfully of their binary position. Each bit addition creates a sum and a carry out. The carry out is then transmitted to the carry in of the next higher-order bit. The final result creates a sum of four bits plus a carry out (c_4).

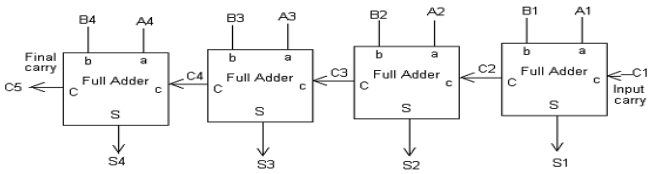

Fig8: Parallel Adders.

II. RESULTS

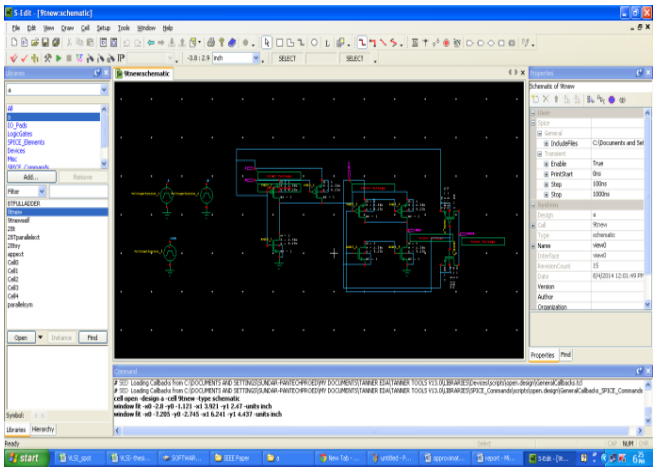

Fig8:9T Full adder.

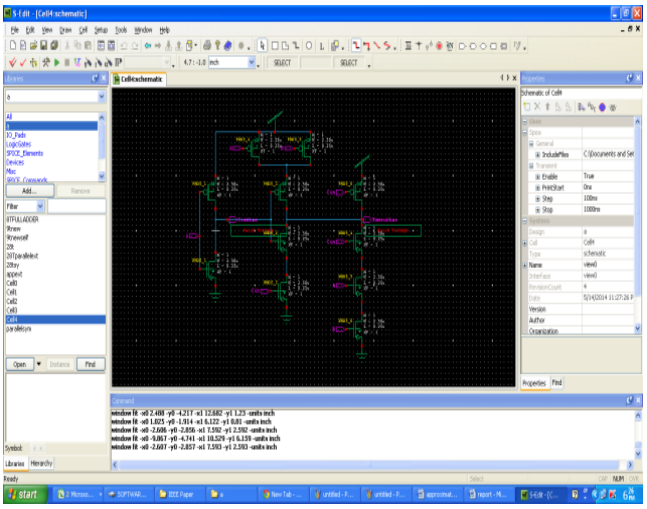

Fig9:Approximation Adder-4

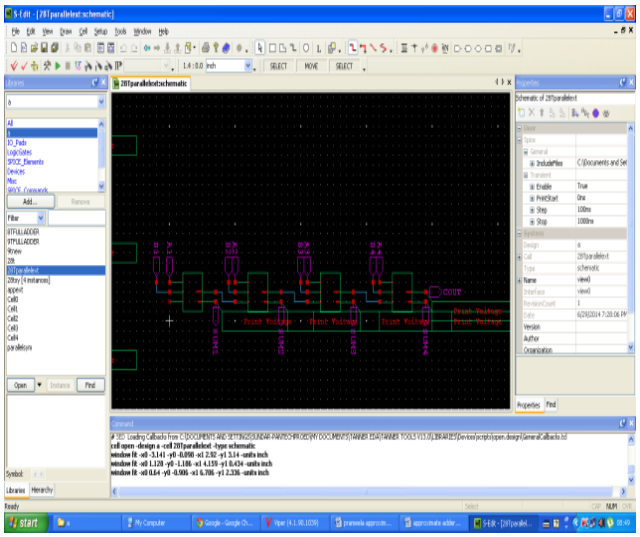

Fig10: Parallel adder with 28T fulladder

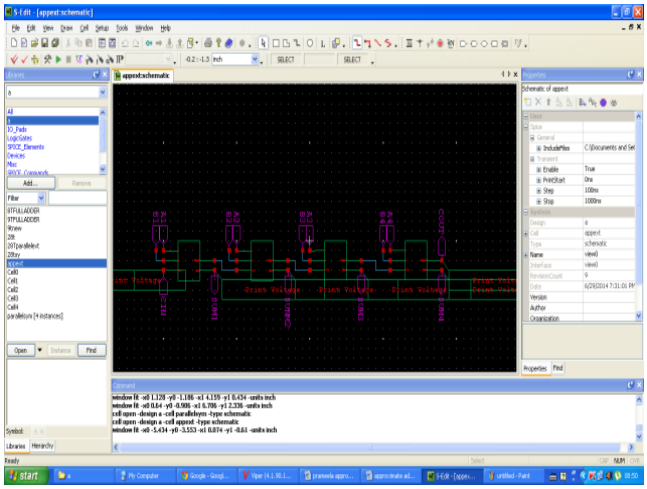

Fig11: Parallel Adder with 9T Full adders.

III. SIMULATION RESULTS

Table3: Simulation Results

Design	Average Power consumption
Conventional Mirror Adder	1.504689e-005 watts
Approximation Adder-1	1.028594e-005 watts
Approximation Adder-2	1.208219e-005 watts
Approximation Adder-3	1.124275e-005 watts
Approximation Adder-4	7.337461e-006 watts
New 9T Full Adder design	2.033670e-006 watts
Parallel Adder	9.717866e-003 watts
Parallel Adder with 9T Full Adder	1.214752e-011 watts

A. Image using Approximation adder Applications

When approximate FA cells are used to design multi-bit adders, the outputs of these adders will be erroneous. Multimedia DSP algorithms mostly consist of additions and multiplications. Multiplications can be treated as shifts and adds. Therefore, adders can be considered as basic building blocks for these algorithms. Interestingly, most DSP algorithms used in multimedia systems are characterized by inherent error tolerance. Hence, occasional errors in intermediate outputs might not manifest as a substantial reduction in the final output quality. We focus on two algorithms, namely, image and video compression, and present the results of using our approximate FA cells in these algorithms. We use approximate FA cells only in the LSBs, thus ensuring that the final output quality does not degrade too much.

B. Image Compression Method

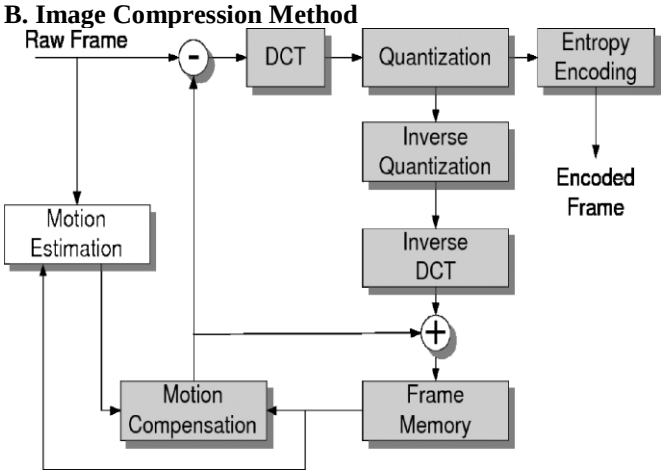

Fig12: DCT and IDCT using Adders.

The power consumption for DCT and IDCT blocks was determined using nanosim run with 12 288 vectors from the standard image in IBM 90-nm technology. The Fig12 shows the total power savings for DCT and IDCT blocks over the base case for different approximations and truncation. The New 9T full Adder provides maximum power savings among all approximations. It is interesting to note that it provides ≈ 60% power savings when 9 LSBs are approximated, with a PSNR of 25.46 dB. In the same scenario, truncation provides ≈ 61% power savings, but the output quality is severely degraded with a PSNR of 13.87 dB.

IV. CONCLUSION

In this paper, we proposed several imprecise or approximate adders that can be effectively utilized to trade off power and quality for error-resilient DSP systems. Our approach aimed to simplify the complexity of a conventional MA cell by reducing the number of transistors and also the load capacitances. When the errors introduced by these approximations were reflected at a high level in a typical DSP algorithm, the impact on output quality was very little. Note that our approach differed from previous approaches a decrease in the number of series connected transistors helped in reducing the effective switched capacitance and achieving voltage scaling. We also derived simplified mathematical models for error and power consumption of an approximate RCA using the approximate FA cells. Using these models, we discussed how to apply these approximations to achieve maximum power savings subject to a given quality constraint. Reducing the Transistor count to achieve the Approximation adders. In future enhancement is 6Transistor to design the adders

V. REFERENCES

[1] W. Dally, J. Balfour, D. Black-Shaffer, J. Chen, R. Harting, V. Parikh, J. Park, and D. Sheffield, "Efficient embedded computing," Computer, vol. 41, no. 7, pp. 27–32, Jul. 2008.

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

[3] R. Hegde and N. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design, Aug. 1999, pp. 30–35.

[4] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," IEEE Trans. Very Large Scale Integr. Syst., vol. 9, no. 6, pp. 813–823, Jun. 2001.

[5] B. Shim, S. Sridhara, and N. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Trans. Very Large Scale Integr. Syst., vol. 12, no. 5, pp. 497–510, May 2004.

[6] G. Varatkar and N. Shanbhag, "Energy-efficient motion estimation using error-tolerance," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design, Oct. 2006, pp. 113–118.

[7] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design, Aug. 2009, pp. 195–200.

[8] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," in Proc. Design, Automat. Test Eur., 2007, pp. 1–6.

[9] G. Karakonstantis, D. Mohapatra, and K. Roy, "System level DSP synthesis using voltage overscaling, unequal error protection and adaptive quality tuning," in Proc. IEEE Workshop Signal Processing Systems, Oct. 2009, pp. 133–138.

[10] L. N. Chakrapani, K. K. Muntimadugu, L. Avinash, J. George, and K. V. Palem, "Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation," in Proc. CASES, 2008, pp. 187–196.

[11] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. Design, Automat. Test Eur., 2008, pp. 1250–1255.

[12] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power highspeed adder for error-tolerant application," in Proc. IEEE Int. Symp. Integr. Circuits, Dec. 2009, pp. 69–72.
[13] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.
[14] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. Design, Automat. Test Eur., 2010, pp. 957–960.
[15] B. J. Phillips, D. R. Kelly, and B. W. Ng, "Estimating adders for a low density parity check decoder," Proc. SPIE, vol. 6313, p. 631302, Aug. 2006.
[16] 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. Part I, vol. 57, no. 4, pp. 850–862, Apr. 2010.
[17] D. Shin and S. K. Gupta, "A re-design technique for datapath modules in error tolerant applications," in Proc. 17th Asian Test Symp., 2008, pp. 431–437.
[18] D. Kelly and B. Phillips, "Arithmetic data value speculation," in Proc. Asia-Pacific Comput. Syst. Architect. Conf., 2005, pp. 353–366.
[19] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.