

Design of Optimizing Adders for Low Power Digital Signal Processing

Mr. Akhil M S

Dept of Electronics and Communication,
Francis Xavier Engineering College,
Tirunelveli-627003,
akhilmsakhilam@gmail.com

Mr. N. Muthukumar

Dept of Electronics and Communication,
Francis Xavier Engineering College,
Tirunelveli-627003
kumaranece@gmail.com

Abstract –

Low power is the important requirement for portable devices and multimedia devices. Which use several signal processing algorithms and architectures. Previous research uses the error resiliency primarily through voltage over scaling, then using algorithmic and architectural techniques to reduce the resulting errors. In this paper propose logic complexity reduction at the transistor level. This is used as the alternative approach to take advantage of the relaxation of numerical accuracy. We implement this concept by proposing several imprecise or approximate full adder cells with reduced complexity at the transistor level, and these approximate adders are used to design approximate multi-bit adders. In addition to an inherent reduction in switched capacitance as the number of transistor reduces, and which also result in shorter critical paths it also produce voltage scaling

Index Terms— Approximate computing, less power, mirror adder.

I. INTRODUCTION

Human beings can gather useful information from slightly erogenous environments. This particular ability helps when interpreting a noise; this allows the results of these low power algorithms not to be accurate numerically rather than approximately accurate. This relaxation on numerical computation allows us to carry out imprecise or approximate computation. We can use this to implement low power design at transistor level and then use this to implement it in higher architectures and algorithms.

In [1] it is shown that an embedded reduced instruction set computing processor uses 70% of the energy for the distribution of instructions and data, and only 6% of the energy to perform the arithmetic operations. A power-efficient multiplier architecture is proposed in [1] it use a 2×2 inaccurate multiplier block from the Karnaugh map simplification. This paper also uses the logic complexity reduction by Karnaugh map simplification. Some other works that aims that related to the logic complexity reduction at gate level are implemented in [4]. Complexity reduction at the algorithm level to obtain real-time energy constraints are used in [5], [6]. Earlier works related to logic complexity reduction is based only on the algorithm, logic level, and gate levels. We used logic complexity reduction at the transistor level. We implement this concept in

bit level by reducing the number of transistor in the conventional mirror adder (MA) circuit. We implement approximate but reduced arithmetic units; it gave us an extra layer of power savings than that of conventional low-power design techniques. This is results to the decreased logic complexity of the proposed approximate arithmetic units. Complexity reduction case power reduction in two ways. First, a reduction in the internal node capacitance and leakage results from lesser transistor. Second, complexity reduction frequently gives shorter critical paths, facilitating voltage reduction with no timing errors. Our focus is to target low-power design using simplified and approximate logic implementations.

An example version of our work appeared in [3]. The extension of the paper [3] by giving two more simplified versions of the MA. Also introduce a methodology that can be used to obtain maximum power savings using approximate adders, subject to a specific quality constraint. The contributions in this paper are summarized as follows.

- 1) The logic complexity reduction of a conventional mirror adder cell by removing the number of transistors and thus reducing the switched capacitances. By using this methodology, we propose five different simplified versions of the

- MA, and also ensure minimum errors in the full adder truth table.
- 2) To obtain reasonable result, we use approximate FA cells only in the least significant bits (LSBs) and use the conventional adder cell in most significant bits (MSBs). Using this FA cells as the basic building blocks. We use this approximate adders to built larger adder cells such as carry save adders (CSA) and ripple carry adders (RCA).
 - 3) Voltage over scaling (VOS) was a very common technique to reduce the power consumption. However, VOS will introduce delay failures in the most significant bits (MSBs). This might lead to huge errors in corresponding outputs and severely mess up the output quality of the application. We use approximate FA cells particularly in the LSBs, while the MSBs use accurate FA cells.
 - 4) We proposed designs for Noise Cancellation algorithms (LMS algorithm) using the proposed approximate arithmetic units and evaluate the approximate architectures in terms of output quality and power dissipation.

The remainder of our paper is organized as follows. In Section II, we discuss various approximate FA cells. In Section III we propose 9T Full Adder Design. In Section IV, We derive mathematical models for mean error and power consumption of approximate Adders. Finally, conclusions are drawn in Section V.

II. APPROXIMATE ADDERS

In this section, we propose discuss various methodologies for designing approximate adders. We use RCAs and CSAs throughout our simultaneous discussions in all sections..

A. Approximation Strategies for the Mirror Adder

In this section, 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/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.

1) *Conventional MA*: 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.

2) *Approximation 1*: 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 C_{in} 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

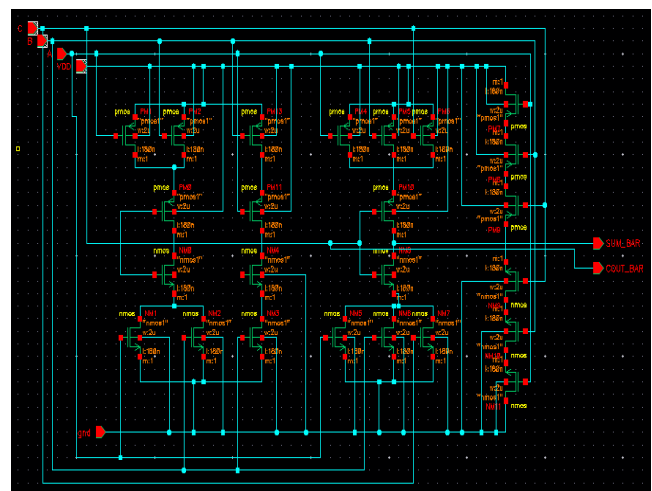


Fig. 1. Conventional MA

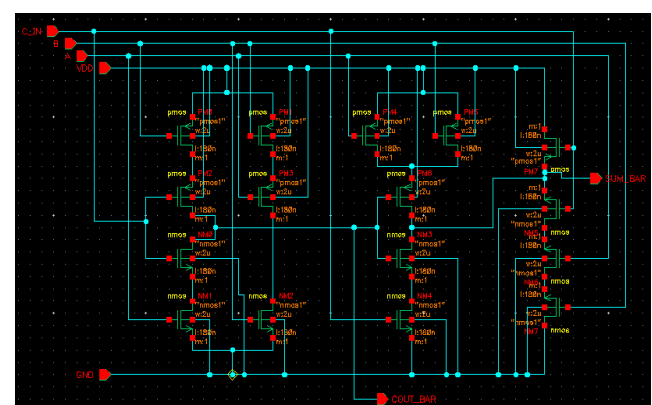


Fig. 2. MA approximation 1.

3) *Approximation 2*: The truth table of an FA indicates that $Sum = Cout$ for six out of eight cases, except for the input combinations $A = 0, B = 0, C_{in} = 0$ and $A = 1, B = 1, C_{in} = 1$. 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 C_{out} (see Fig. 3) to produce the same functionality.

4) *Approximation 3*: Further simplification can be

obtained by integrating approximations 1 and 2. Note that this produces one error in C_{out} and three errors in Sum , as shown in Table I.

TABLE - I

Truth Table for Conventional FA and Approximations 1-4

| 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× | 0✓ | 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✓ |

5) *Approximation 4*: A deep observation of the FA truth table shows that $C_{out} = A$ for six out of eight cases. Obviously, $C_{out} = B$ for six out of eight cases. Since A and B are interchangeable, we consider $C_{out} = A$. Thus, we propose approximation 4 where we just use an inverter with input A to calculate C_{out} and Sum is calculated similar to approximation 1.

6) *Approximation 5*: If we need to make Sum

independent of C_{in} , we have two choices, $Sum = A$ and $Sum = B$. Thus, we had two alternatives for approximation 5, namely, $Sum = A$, $C_{out} = A$ and $Sum = B$, $C_{out} = A$. If we focus choice 1, we find that both Sum and C_{out} match with accurate outputs in only two out of eight cases. In choice 2, Sum and C_{out} match with perfect outputs in four out of eight cases. Therefore, to reduce errors both in Sum and C_{out} , we go for choice 2 as approximation 5.

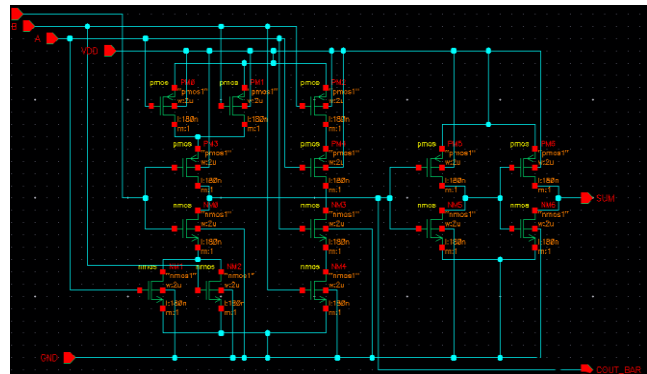


Fig. 3. MA approximation 2.

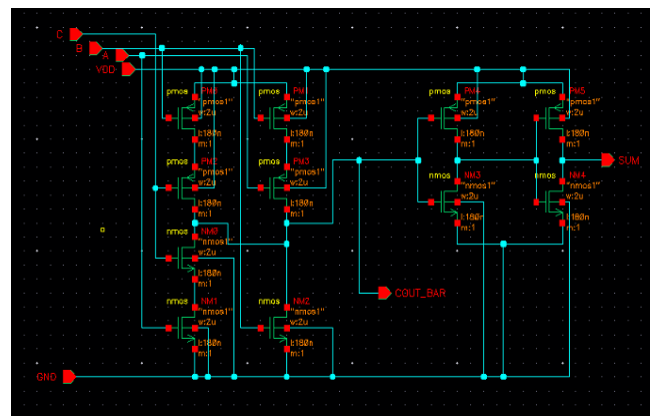


Fig. 4. MA approximation 3.

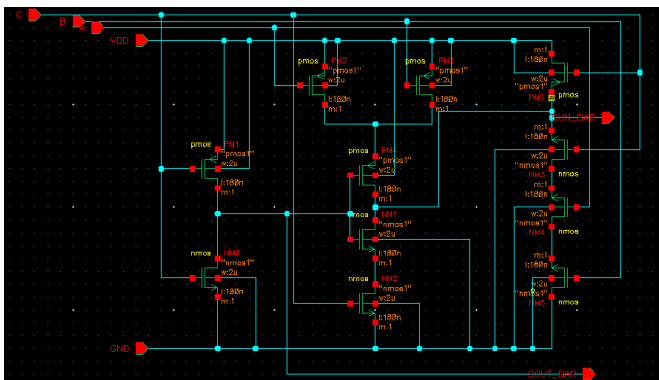


Fig. 5. MA approximation 4

III. PROPOSED WORK

A. XOR-XNOR Full Adder Design

The schematic of 9T full adder cell is shown in Figure 3 and its truth table is given in Table 1. 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 C_{in} , it is desired to compute the two 1-bit outputs Sum and $Cout$, given by.

$$Sum = A \oplus B \oplus C_{in}.$$

$$Cout = A \cdot B + C_{in} (A \oplus B).$$

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 C_{in} . Similarly, when $A = "1"$, Sum focusing the XNORing between inputs B and C_{in} . Therefore, the operation of Sum module depends on implementing XOR operation and XNOR operation between inputs B and C_{in} which is indicated below. The logic for $Cout$ output is shown As following,

$$\text{When } A = 0, Sum = B \oplus C_{in}.$$

$$\text{When } A = 1, Sum = B \odot C_{in}$$

Total power consumption in CMOS logic circuits will be expressed as sum of three components are shown:

$$P_{Total} = P_{Switching} + P_{Sub} + P_{Short},$$

where $P_{switching}$ indicates the average switching power consumption and is given by,

$$P_{Switching} = \alpha T_{load} V^2 DDf.$$

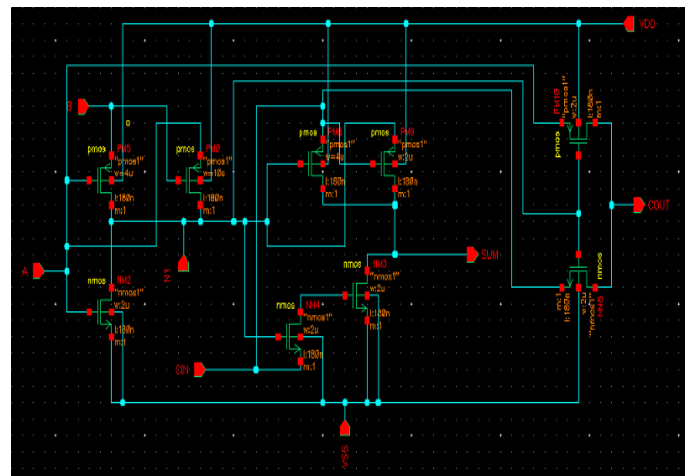
P_{Sub} indicates sub threshold power consumption and is given by

$$P_{Sub} = VDD \times I_{Sub},$$

Where

$$I_{Sub} = K \times \exp[(VGS - Vt)/q\eta KT] \times [1 - \exp(-VDS.q/KT)]$$

The simplified value of VGS in sub threshold mode decreases current exponentially and thus reduces sub threshold power



. Fig. 6. Proposed XOR-XNOR full adder cell.

B. Approximate XOR-based adder

Fig. 7 shows the first approximate adder. In this design, the XOR operation is achieved by an inverter and two pass transistors connected to A and B respectively. When A is "1", $I = A'$, otherwise, $I = A$, i.e., $I = A \oplus B$.

Both Sum and $Cout$ are accurate for 4 out of the total 8 input combinations. The total error distance achieved with this design is 4

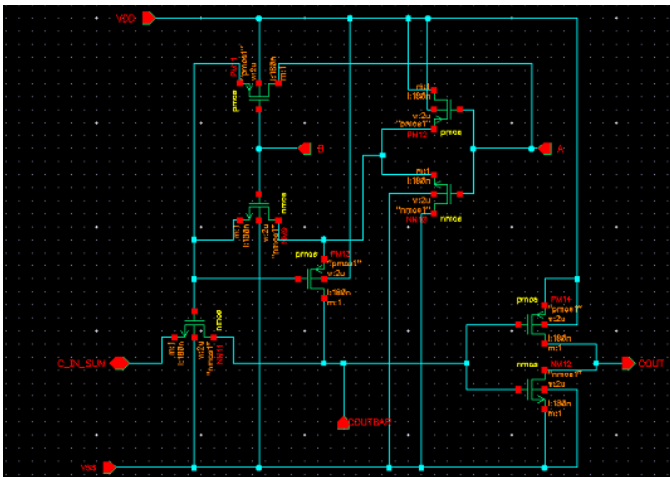


Fig. 6. Proposed Approximate XOR-based adder

TABLE -II

Truth Table for 9t full adder and Approximate XOR adder

| Inputs | | | 9T full adder | | Approximate XOR adder | |
|--------|---|-----------------|---------------|------------------|-----------------------|-------------------|
| A | B | C _{in} | Sum | C _{out} | Sum ₁ | C _{out1} |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

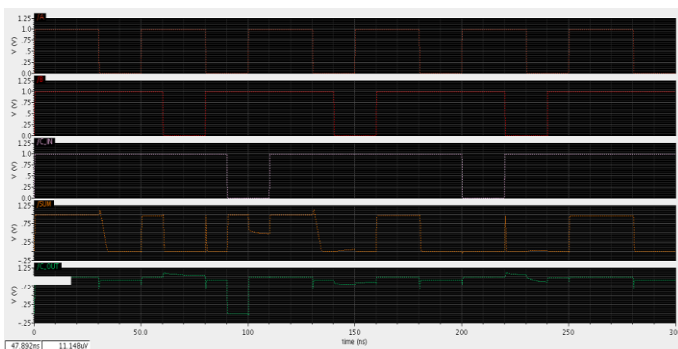


Fig.7 output wave form XOR-XNOR full adder cell.

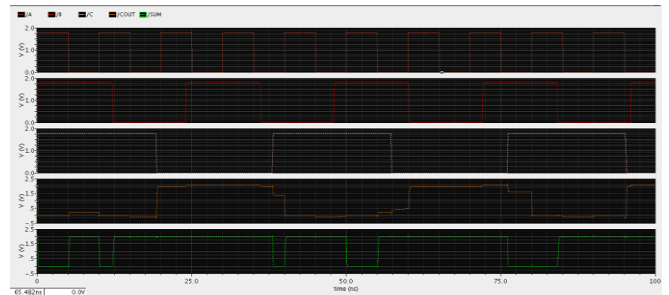


Fig.8. output wave form XOR-XNOR full adder cell.

C.Simulations and Comparison

All schematic simulations are done on cadence version 6.1 using 180 nm technology and input voltage varies from 0.2 V to 0.3 V in steps of 0.02 V. In order to justify that proposed design is consuming less power and have better performance, simulations are carried out for power-delay product at rising frequency, input voltage.

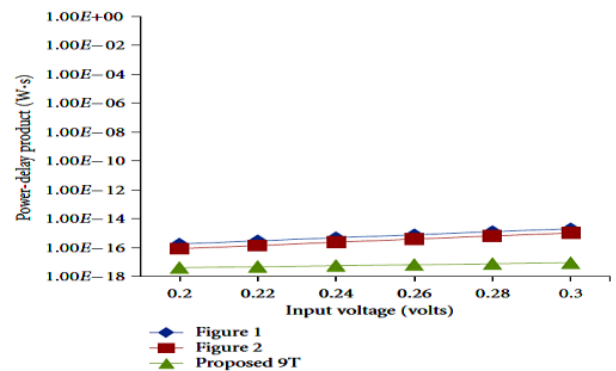


Fig. 9. Power-delay product with increasing input voltage.

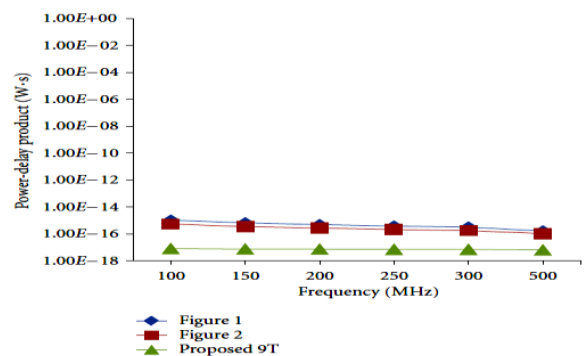


Fig. 10. Power-delay product with increasing operating frequency at 0.3 V input voltage and supply voltage

IV. IMAGE AND VIDEO COMPRESSION USING APPROXIMATE ARITHMETIC UNIT

Several approximate FA cells were introduced. Using FA cells also introduces errors in the truth table of an FA. 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. In 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.

Image compression using approximate FA cells. We use approximate FA cells only in the LSBs, thus ensuring that the final output quality does not degrade too much. The DCT and inverse discrete cosine transform (IDCT) are integral components of a Joint Photographic Experts Group (JPEG) image compression system

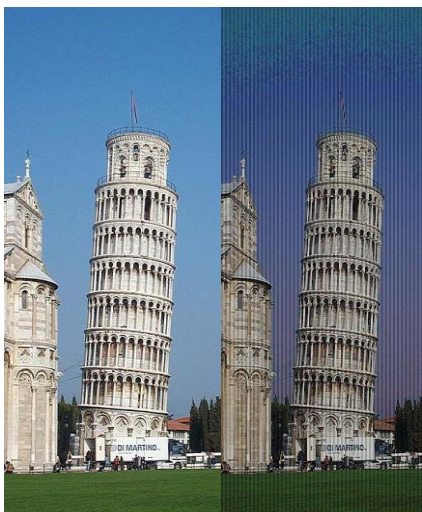


Fig a approximation in LSB Fig b approximation in MSB

V. POWER AND ERROR MODELING FOR APPROXIMATE ADDER

Mathematical models for mean error and power consumption of approximate adders

A. Modeling Error In Approximate Adders

Let denote the signal probabilities $P(A[x] = 1)$ and $P(B[x] = 1)$ by a_x and b_x , respectively. Here x is the bit position (starting from 0). Let the

accurate signal probabilities $P(\text{Sum}[x] = 1)$ and $P(\text{Cin}[x] = 1)$ be denoted by s_x and c_x , respectively. Similarly, let the approximate signal probabilities $P(\text{Sum}'[x] = 1)$ and $P(\text{Cin}'[x] = 1)$ be denoted by s'_x and c'_x , respectively. Without loss of generality, let us assume the input bit probabilities $a_x = 0.5$ and $b_x = 0.5$. We calculate the signal probabilities s_x , c_x , s'_x , and c'_x for both conventional FA and approximate FA cells

For the accurate FA, $\text{Sum} = ABC_{in} + A'B'C_{in} + A'B'C'_{in} + AB'C'_{in}$, $\text{Cout} = AB + A'BC_{in} + AB'C_{in}$. This gives us the following expressions for s_x and c_{x+1} .

$$\begin{aligned} s_x &= a_x b_x c_x + (1 - a_x)(1 - b_x)c_x + (1 - a_x)b_x(1 - c_x) + a_x(1 - b_x)(1 - c_x) \\ &= a_x + b_x + c_x - 2a_x b_x - 2a_x c_x - 2b_x c_x + 4a_x b_x c_x \\ &= 0.5 \\ c_{x+1} &= a_x b_x + (1 - a_x)b_x c_x + (1 - b_x)a_x c_x \\ &= a_x b_x + b_x c_x + a_x c_x - 2a_x b_x c_x \\ &= (2c_x + 1)/4. \end{aligned}$$

$c_0 = 0$, using the recursive relationship for c_{x+1} in the above equation, $c_x = (0.5 - 2^{-(x+1)})$, $x = 1, 2, \dots$. Similar to the accurate case, Sum and Cout expressions for all approximations can be calculated. The error in approximate addition is given by

$$\begin{aligned} E &= (\text{Sum}'[0] - \text{Sum}[0]) + 2(\text{Sum}'[1] - \text{Sum}[1]) + \dots \\ &+ 2^{y-1} (\text{Sum}'[y-1] - \text{Sum}[y-1]) + 2^y (\text{Cin}'[y] - \text{Cin}[y]) \\ &= e[0] + 2e[1] + \dots + 2^{y-1}e[y-1] + 2^y e[y]. \end{aligned}$$

B. Modelling Power Consumption of Approximate Adders

Now we derive simple mathematical models for estimating the power consumption of an approximate Adders. Let C_{gn} and C_{gp} be the gate capacitance of a minimum size nMOS and pMOS transistor, respectively. Similarly, let C_{dn} and C_{dp} be the drain diffusion capacitances respectively. If the pMOS transistor has three times the width of the nMOS transistor, then $C_{gp} \approx 3C_{gn}$ and $C_{dp} \approx 3C_{dn}$. Let us also assume that $C_{dn} \approx C_{gn}$. In a multilevel adder tree, the Sum bits of intermediate outputs become the input bits A and B for the subsequent adder level. The output capacitance at each Sum node is $C_{dn} + C_{dp}$. The schematic of the conventional MA in Fig. 1 can be used to calculate

the input capacitances at nodes A,B and Cin. Thus, the total capacitance at node A can be written as $(C_{dn} + C_{dp}) + 4(C_{gn} + C_{gp}) \approx 20C_{gn}$. Similarly, the total capacitance at node B is $(C_{dn} + C_{dp}) + 4(C_{gn} + C_{gp}) \approx 20C_{gn}$, while the capacitance at node Cin is $(C_{dn} + C_{dp}) + 3(C_{gn} + C_{gp}) \approx 16C_{gn}$. Continuing this way, the total capacitances at nodes A,B and Cin for all approximations can be calculated using their transistor-level schematics.

VI. CONCLUSION

Low and ultra low-power circuits are becoming more desirable as portable devices markets are growing and they also become more interested and applicable in biomedical, pharmacy and sensor networking application because of the CMOS reliability improvement and the nano metric scaling. These imprecise or approximate adders that can be effectively utilized to trade off power and quality for error-resilient systems. This approach focused to simplify the complexity of a conventional MA cell by reducing the number of transistors and also the load capacitances. This approach is different from earlier approaches where errors were introduced due to VOS. A decrease in the number of series connected transistors helped in decreasing the effective switched capacitance and achieving voltage scaling. Using these models, discussed how to apply these approximations to get maximum power savings subject to a given quality constraint. This procedure has been illustrated for Digital image processing

REFERENCES

- [1] P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading accuracy for power with an underdesigned multiplier architecture," in Proc. 24th IEEE Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.
- [2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: Imprecise adders for low-power approximate computing," in Proc. IEEE/ACM Int. Symp. Low-Power Electron. Design, Aug. 2011, pp. 409–414
- [3] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in Proc. Design, Automat. Test Eur., 2010, pp. 957–960.
- [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. Part I, vol. 57, no. 4, pp. 850–862, Apr. 2010

- [5] Y. V. Ivanov and C. J. Bleakley, "Real-time h.264 video encoding in software with fast mode decision and dynamic complexity control," ACM Trans. Multimedia Comput. Commun. Applicat., vol. 6, pp. 5:1–5:21, Feb. 2010.
- [6] M. Shafique, L. Bauer, and J. Henkel, "enBudget: A run-time adaptive predictive energy-budgeting scheme for energy-aware motion estimation in H.264/MPEG-4 AVC video encoder," in Proc. Design, Automat. Test Eur., Mar. 2010, pp. 1725–1730
- [7] E. Lyons, V. Ganti, R. Goldman, V. Melikyan, and H. Mahmoodi, "Full-custom design project for digital VLSI and IC design courses using synopsys generic 90nm CMOS library," in Proc. IEEE Int.
- [8] J. Choi, N. Banerjee, and K. Roy, "Variation-aware low-power synthesis methodology for fixed-point FIR filters," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 28, no. 1, pp. 87–
- [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
- [10] W. Dally, J. Balfour, D. Black-Shaffer, J. Chen, Harting, V. Parikh, J. Park, and D. Sheffield, "Efficient embedded computing," Computer, vol. 41, no. 7, pp. 27–32, Jul.