International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

Design Of Low-Power Approximate Adders For Signal Processing Applications

R. Sabitha[1], K. Sharmila[2], M. Sindhuja[3], T. Suganya[4]

VSB Engineering College/Dept of ECE, Karur,Tamilnadu,India.

***Abstract*—Low power is the imperative requirement for portable multimedia devices employing several signal processing algorithms and architectures. 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 multi-bit adders. In addition to an inherent reduction in switched capacitance, our tier result in significantly shorter critical paths, enabling voltage scaling. We implement our concept in the application of Noise Cancellation algorithms (LMS algorithm). Simulation outputs indicate up to 69% power savings using the proposed approximate adders, when compared to previous implementations using accurate adders.**

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

I. INTRODUCTION

Human beings have restricted perceptual abilities when interpreting a noise. This allows the results of these algorithms to be numerically approximate rather than accurate. This relaxation on numerical exactness gives some freedom to carry out imprecise or approximate computation. We can use this freedom to come up with low power designs at various levels of design abstraction, namely, logic, architecture, and algorithm.

It is shown in [1] that an embedded deduced instruction set computing processor consumes 70% of the energy in supplying data and instructions, and 6% of the energy while performing arithmetic only. In our paper, we consider application-specific integrated circuit implementations of error-resilient applications like Noise Cancellation algorithms (LMS –least mean square algorithm).

A power-efficient multiplier architecture was proposed in [1] that uses a 2 × 2 inaccurate multiplier block resulting from Karnaugh map simplification. Our paper considers logic complexity reduction using Karnaugh maps. Other works that aims on logic complexity reduction at the gate level are [4]. Various approaches use complexity reduction at the algorithm level to meet real-time energy constraints [5], [6].Previous works on logic complexity reduction have aimed on algorithm, logic, and gate levels. We used logic complexity reduction at the transistor level. We apply this to addition at the bit level by reducing the mirror adder (MA) circuit. We develop imprecise but reduced arithmetic units, which gives an extra layer of power savings over conventional low-power design techniques. This is attributed to the decreased logic complexity of the proposed approximate arithmetic units. Complexity reduction brings power reduction in two different ways. First, an inherent reduction in internal node capacitance and leakage results from having lesser hardware. 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 exampled 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.

1) 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.
2) To maintain a reasonable result, we use approximate FA cells only in the least significant bits (LSBs). We particularly aimed on adder structures that use FA cells as their basic building blocks. We prefered carry save adders (CSA) and ripple carry adders (RCA).
3) VOS was a very popular technique to get large improvements in power consumption. However, VOS will bring 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.
5) Finally, we demonstrate the optimization methodology with the help of Adaptive Noise Canceller System (LMS Algorithm).

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 an approximate RCA. In Section V, we use the approximate FA cells to design architectures for Noise Cancellation (LMS) algorithm and highlight the potential benefits. Finally, conclusions are drawn in Section VI.

II. APPROXIMATE ADDERS

In this section, we 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

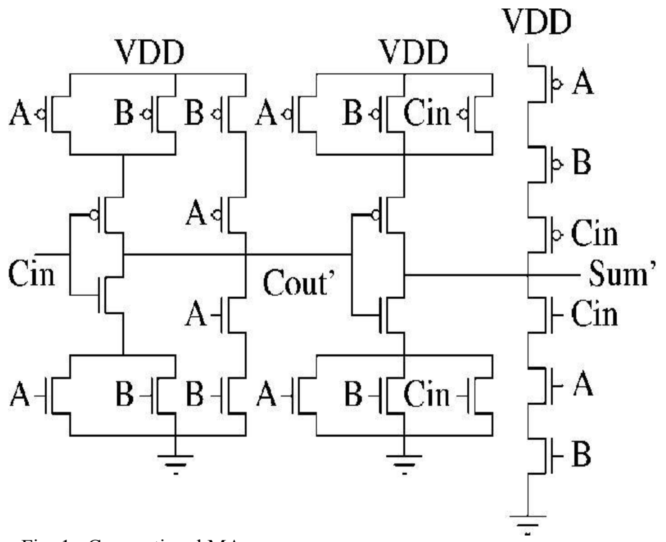


Fig. 1. Conventional MA.

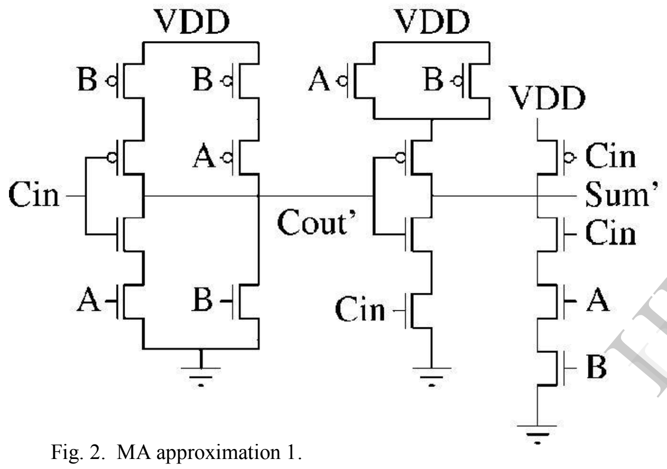


Fig. 2. MA approximation 1.

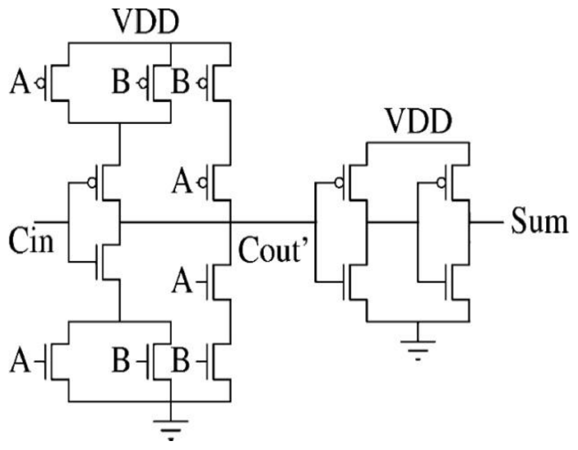


Fig. 3. MA approximation 2.

transistors also leads in reducing the αC term (switched capacitance) in the dynamic power expression

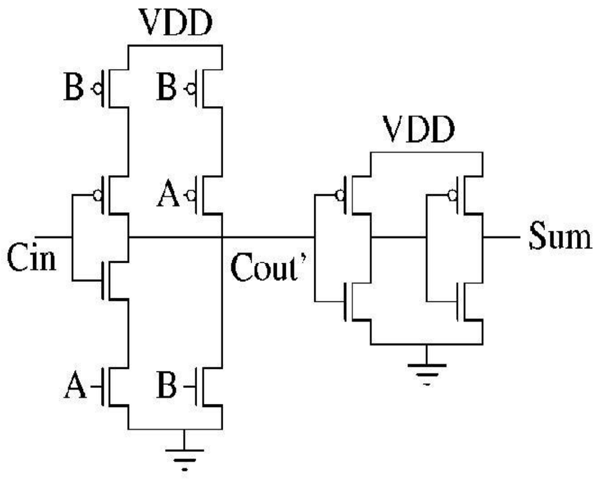


Fig. 4. MA approximation 3.

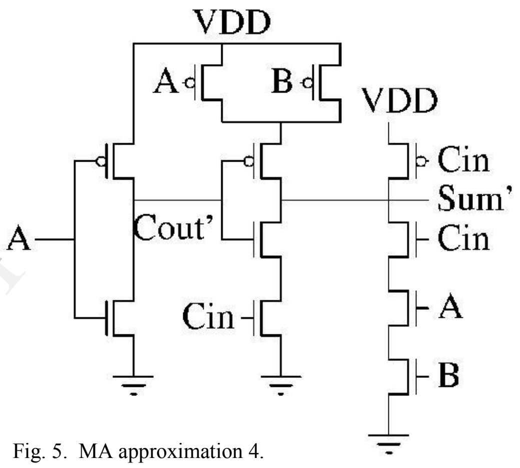


Fig. 5. MA approximation 4.

$P_{\text{dynamic}} = \alpha C V^2_{DD} f$, 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

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 ×	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 ✓

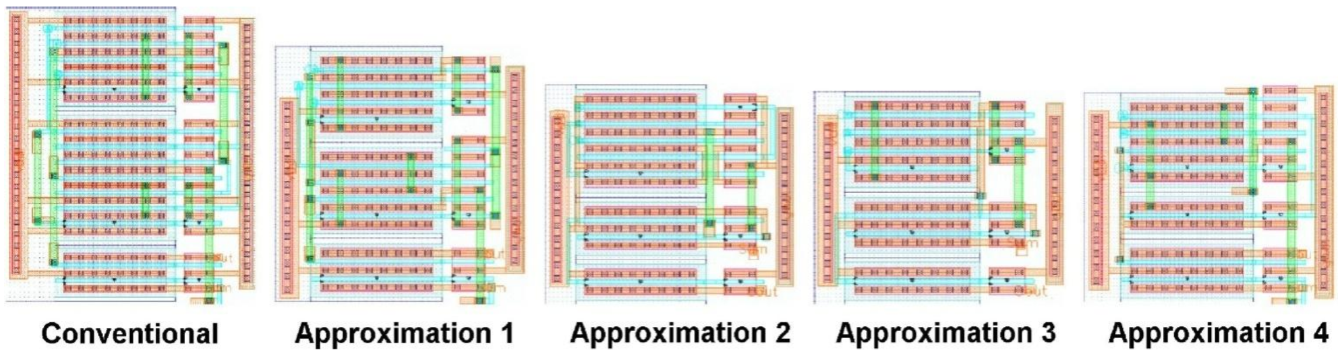


Fig. 6. Layouts of conventional and approximate MA cells.

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

3) *Approximation 2:* The truth table of an FA indicates that Sum= $\overline{C^1_{out}}$ 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, $\overline{C_{out}}$ is calculated in the first stage. Thus, an simple way to get a simplified schematic is to set Sum=$\overline{C_{out}}$. However, we introduce a buffer stage after $\overline{C_{out}}$ (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.

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 $\overline{C_{out}}$ 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. Our important thrust here is to ensure that for a particular input combination (A, B and C_{in}), ensuring correctness in Sum also makes C_{out} correct. Layouts of conventional MA and various approximations in IBM 90-nm technology are shown in Fig. 6. Layout area for the conventional MA and various approximations are compared in Table III. Approximation 5 uses particularly buffers. The layout area of a buffer is 6.77 μm^2.

B. Voltage Scaling Using Approximate Adders

TABLE II Choosing Approximation 5

Choice 1		Choice 2	
Sum= A	$C_{out} = A$	Sum= B	$C_{out} = A$
0 ✓	0 ✓	0 ✓	0 ✓
0 ×	0 ✓	0 ×	0 ✓
0 ×	0 ✓	1 ✓	0 ✓
0 ✓	0 ×	1 ×	0 ×
1 ✓	1 ×	0 ×	1 ×
1 ×	1 ✓	0 ✓	1 ✓
1 ×	1 ✓	1 ×	1 ✓
1 ✓	1 ✓	1 ✓	1 ✓

III. PROPOSED 9T FULL ADDER DESIGN

A. 9T 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 Cin, it is desired to compute the two 1-bit outputs Sum and Cout, given by

$$\text{Sum} = A \oplus B \oplus \text{Cin}.$$

$$\text{Cout} = A \cdot B + \text{Cin}\,(A \oplus B).$$

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 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 which is

TABLE III

Truth table of 1-bit full adder.

A	*B*	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Indicated below. The logic for Cout output is shown
As following,
When *A* = 0,

$$\text{Sum} = B \oplus \text{Cin}.$$

When *A* = 1,

$$\text{Sum} = B \odot \text{Cin}.$$

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

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

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

$$P_{\text{Switching}} = \alpha\,\tau C_{\text{load}} V^{2}_{DD} f\,.$$

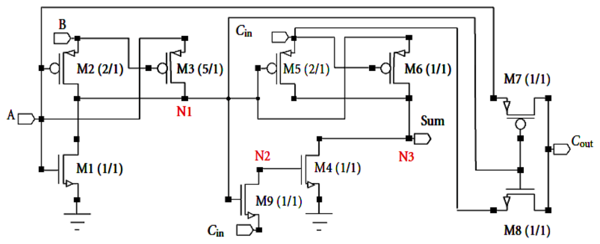


Fig. 8. Proposed 9T full adder cell.

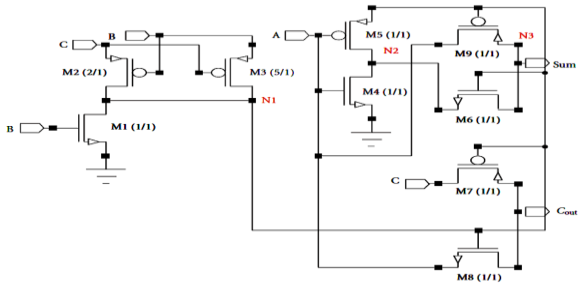


Fig. 9. Proposed 9T full adder cell

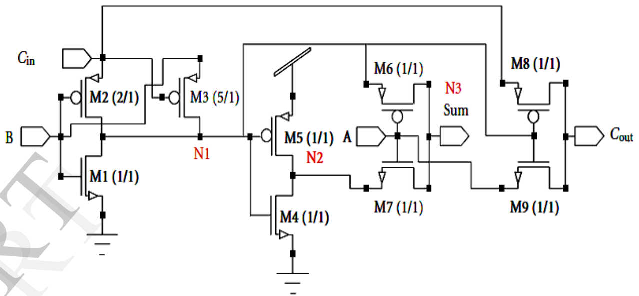


Fig. 10. Proposed 9T full adder cell

P_{Sub} indicates subthreshold power consumption and is given by

$$P_{\text{Sub}} = V_{\text{DD}} \times I_{\text{Sub}},$$

Where

$$I_{\text{Sub}} = K \times exp\left[\frac{(V_{GS}-V_t)q}{\eta KT}\right],$$

$$\times\left[1-\exp(V_{DS}\,\frac{q}{KT})\right]$$

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

TABLE IV

Figure 8			Figure 9			Proposed 9T (Figure 10)		
N1	N2	N3	N1	N2	N3	N1	N2	N3
1	4	3	1	3	3	2	2	3
Total = 8			Total = 7			Total = 7		

In a nutshell, the proposed 9T full adder proves itself to be a better option for less-power compact systems. All the substrate terminals in Figures 8, 9 and 10 are connected to their respective source terminals to nullify the substrate-bias effect.

B.Simulations and Comparison

All schematic simulations are done on Tanner EDA tool version 13.0 using 45 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 and temperature.

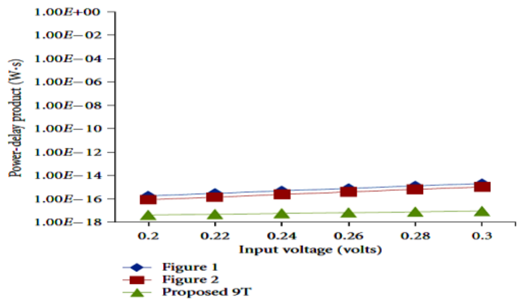


Fig. 11. Power-delay product with increasing input voltage.

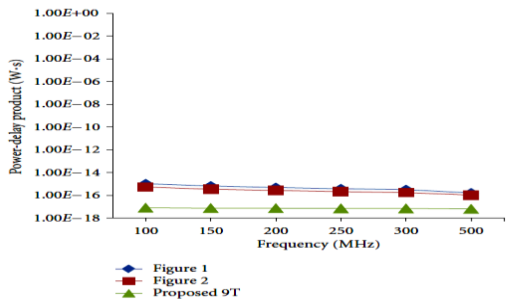


Fig. 12. Power-delay product with increasing operating frequency at 0.3 V input voltage and supply voltage.

Figures 11, 12, and 13 shows that the proposed 9T full adder cell proves its superiority in terms of power-delay product at different input voltages and frequencies and temperature sustainability over existing 9T adders.

The current equation of a pn-junction is given as

$$I=I_0\left(e^{q\frac{V}{KT}}-1\right)$$

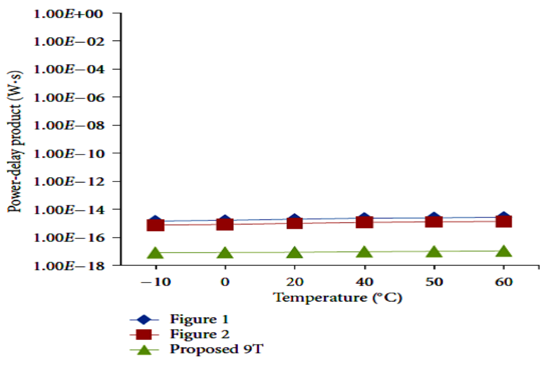


Fig. 13. Power-delay product with varying temperature at 0.3 V input voltage and supply voltage.

This shows that with increasing in temperature, current increases exponentially and produces high power consumption.

IV.MEAN ERROR AND VARIANCE IN APPROXIMATE ADDITION

In this section, we show mean error and mean variance in approximate addition.

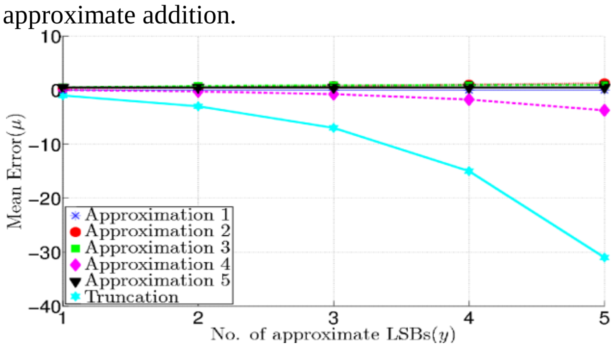


Fig. 14. Mean error in approximate addition.

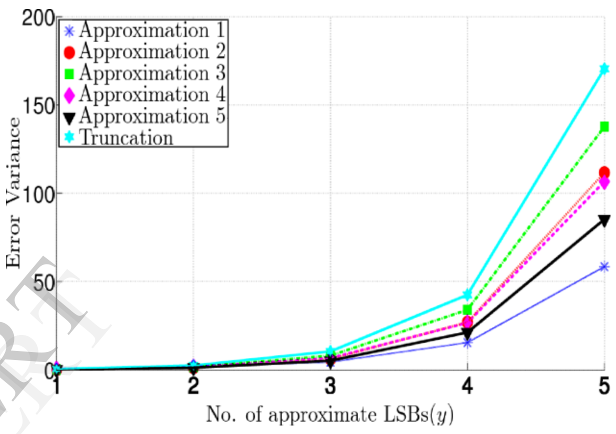


Fig. 15. Error variance in approximate addition.

We also related the mean error for the proposed approximations with truncation, another wellknown approximation technique. From the graph, it is clear that the proposed approximations are better than truncation in terms of mean error. Approximation 1 is the best (having an mean error of zero), where Approximation 4 is the poor.

B. Modelling Power Consumption of Approximate Adders

Now we calculate simple mathematical models for estimating the power consumption of an approximate RCA. Let C_{gn} and C_{gp} be the gate capacitance of a lower size nMOS and pMOS transistor, respectively. Obviously, 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 consider that $C_{dn} \approx C_{gn}$. In a multilevel adder tree, the Sum bits of intermediate results become the input bits *A* and *B* for the subsequent adder level. The output capacitance at every Sum node is $C_{dn} + C_{dp}$. The schematic of the conventional MA in Fig. 1 is used to calculate the input capacitances at nodes *A*,*B* and C_{in}. Thus, the total capacitance at node *A* shall be written as $(C_{dn} + C_{dp}) + 4(C_{gn} + C_{gp}) \approx 20C_{gn}$.

TABLE V
Capacitances for Different Approximations

Approximation Technique	Node *A*	Node *B*	Node C_{in}
Approximation 1	12	15	13
Approximation 2	12	12	8
Approximation 3	8	11	8
Approximation 4	12	8	9
Approximation 5	4	8	0

Obviously, thetotal capacitance at node B is $(C_{dn} + C_{dp})+ 4(C_{gn} + C_{gp}) \approx 20C_{gn}$, while the capacitance at node C_{in} is $(C_{dn} + C_{dp}) + 3(C_{gn} + C_{gp}) \approx 16C_{gn}$. Continuing this way, the total capacitances at nodes *A*,*B* and C_{in} for all approximations can be calculated by their transistor-level schematics. Table V gives these values (normalized with respect to C_{gn}). Note that C_{in} [1], C_{in} [2]. . . C_{in} [$y-1$] are not computed in approximation 5 (if y bits are approximate). Thus, the capacitance at node C_{in} for approximation 5 is zero. Therefore, we will use normalized capacitances for all our subsequent discussions.

Since V_{DD} α 1/delay, the scaled voltage is given by

$$V_{DDapp} = V_{DD}\ (1\text{-}(yp/T_C))$$

V_{DD} is the nominal voltage (perfect case), $T_c = 1/f_c$ is the clock period, and f_c is the operating frequency. The term p can be estimated by simulating an *N*-b approximate RCA for different values of y.

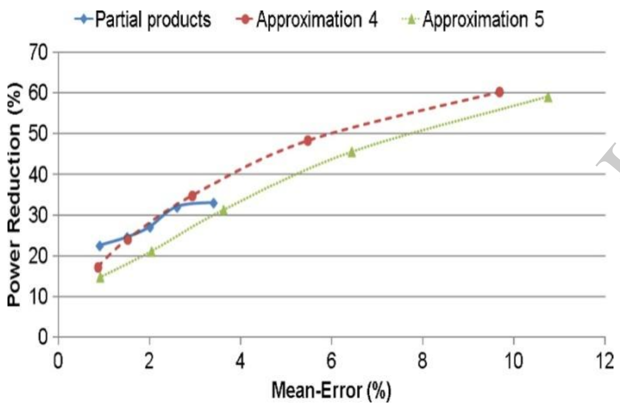


Fig. 16. Comparison with partial products approach.

Hence, the same value of p shall be used for determining the scaled voltage as a function of y in an adder tree. Using the above equations, a first order estimate P_{app} for the power consumption of an approximate RCA shall be written as $P_{app} = (1/2)C_{sw}V_{DDapp}{}^{2}f_c$, which is a function of y.

C. Comparison With Approximate Multipliers

Approximate multipliers based on Karnaugh map simplification had proposed in [1]. There the errors are introduced through partial products. We constructed an 8 × 8 multiplier using the approximate FA cells. The accumulation of partial products (accurate) was done by using a carry-save tree followed by an RCA (approximate). The power consumption was computed by running exhaustive nanosim simulations for both accurate and approximate 8 × 8 multipliers. The mean errors for the respective case were obtained using MATLAB. The error-power tradeoff for the proposed approximate multipliers were compared with the data in [1].In Fig. 16 we plot percentage power reduction versus mean error. We find that approximation 4 performs better than the partial product approach for higher mean errors. Also, the partial product approach reaches the saturation point quicker than approximations 4 and 5.

V. LEAST MEAN SQUARE ALGORITHM

A. Adaptive filter algorithms

An adaptive filter is a time-variant filter whose coefficients are adjusted to optimize a cost function or to satisfy some predetermined optimization criterion. Two main Characteristics of adaptive filters are they can automatically adapt (self-optimize) in the face of changing environments and changing system requirements. They can be trained to perform particular filtering and decision-making tasks according to some updating equations (training rules). It can subsequently operate in, changing environments (e.g. signal detection in wireless channel) . nonstationary signal/noise conditions (e.g. LPC of a speech signal). time-varying parameter estimation (e.g. position tracking of a moving source)

It simply changes the cost function ξ (n) = E [e^2 (n)] by its instantaneous coarse estimate. The error estimation e(n) is

$$e\ (n) = \mathbf{d}(n) - w(n)\ X(n)$$

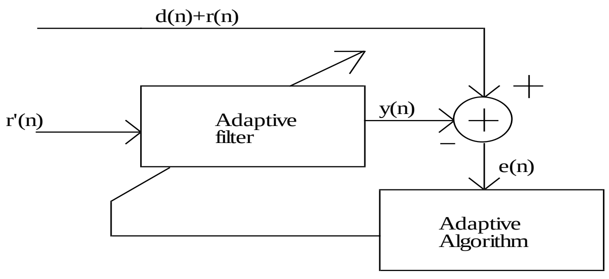
Coefficient updating equation is

$$w\ (n+1) = \mathbf{w}(n) + \mu\ x(n)\ e(n),$$

Where μ is an appropriate step size will be chosen as $0 < \mu < 0.2$ for the convergence of the algorithm. The larger step sizes make the coefficients to fluctuate and eventually become unstable.

B. Noise canceller

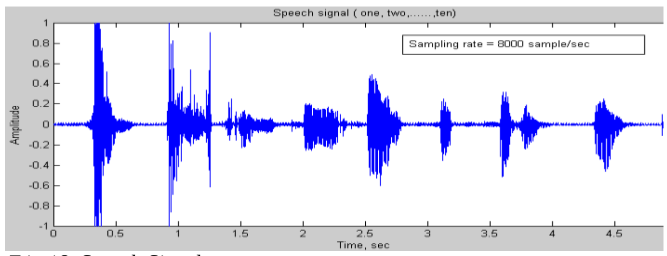
The noise cancellers are used to cancel intense background noise. This configuration can be applied in mobile phones and radio communications, where in some situations these devices are used in high-noise environments. An adaptive noise cancellation system is shown in fig.6.



F ig.17. Adaptive Noise Canceller System

The canceller employs a directional microphone to calculate and estimate the instantaneous amplitude of ambient noise r'(n), and another microphone used to take the speech signal which is contaminated with noise d(n) + r(n). The ambient noise is processed by the adaptive filter to make it same as the noise contaminating the speech signal, and then is subtracted to eliminate the noise in the desired signal. In order for effective noise cancellation the ambient noise should be highly correlated with the noise components in the speech signal, if there is not an access to the instantaneous value of the contaminating signal, the noise does not be cancelled out, but it should be reduced using the statistics of the signal and the noise process.

Speech Signals: A speech signal contains of three classes of sounds. There are voice, fricative and plosive sounds. Voiced sounds are generated by excitation of the vocal tract with quasi-periodic pulses of airflow. Fricative sounds are formed by constrict in the vocal tract and passing air via it, causing turbulence that results in a noise-like sound



F ig.18. Speech Signal

speech signal can be assumed as a linear composite of the above three classes of sound, each of the sounds are stationary and remains fairly constant over intervals of the order of 40 ms. By using matlab real time voice signal is recorded for 3 seconds and it will generate 24000 samples in the voice signal.

The adaptive white Gaussian noise (AWGN) is mixed with the recorded signal to generate the error input, along with that desired signal is generated. This error input signal and desired signal are stored in text file saved as errorsignal.txt and desiredsignal.txt. These text files will be processed by VHDL code in Xilinx 13.1 and simulated using Modelsim. During simulation it produces another two text file named as error.txt and output.txt. At last these test files are read by matlab and generate the ouput signal and error signal.

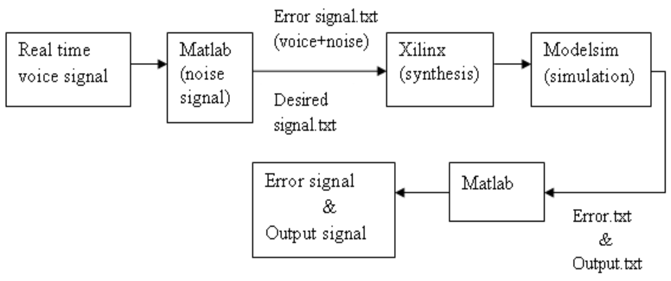
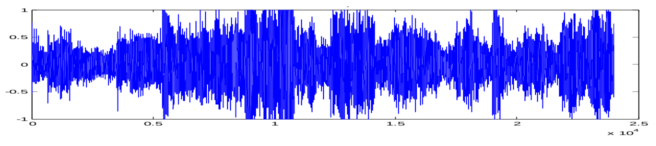
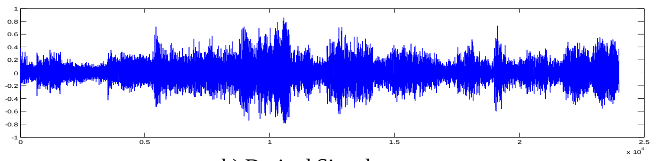


Fig. 19. Experimental Setup for Adaptive Noise Cancellation

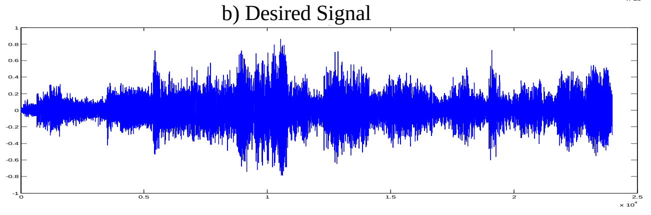
The real time voice signal is fed to the matlab code and it produces 24000 samples. These samples are synthesized for different adaptive algorithm such as LMS, SLMS, SSLMS and RLS for different μ values. The resulting outputs are shown in the figure 20.



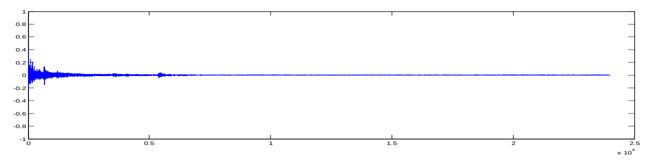
a)Audio Error Input



b) Desired Signal



c) Output for LMS Algorithm



d)Error Signal for LMS Algorithm

Fig. 20.Output Waveform for Noise Cancellation

VI. CONCLUSION

In this paper, we proposed various imprecise or approximate adders that can be effectively utilized to trade off power and quality for error-resilient systems. Our approach focused to simplify the complexity of a conventional MA cell by reducing the number of transistors and also the load capacitances. Note that our approach differed 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, we discussed how to apply these approximations to get maximum power savings subject to a given quality constraint. This procedure has been illustrated for Adaptive Filter in Noise Cancellation algorithms (LMS algorithm). We believe that the proposed approximate adders can be used on top of previously existing low-power techniques like SDC and ANT to extract multifold benefits with a very minimal loss in output quality.

APPENDIX
DERIVATION OF MEAN ERROR FOR AN APPROXIMATE RCA

As mentioned in Section IV-A, the expression for mean error is given by

$$\mu(y) = E(\varepsilon)$$
$$=\sum_{x=0}^{y} 2^{x} E(e[x])$$

Clearly, $e[x] \; \varepsilon \; \{-1, 0, 1\}$. Thus, $E(e[x])$ forall x ε $\{0, 1, \dots, y\}$ can be calculated as following:

$$E(e[x]) = \mathrm{P}(e[x] = 1) - \mathrm{P}(e[x] = -1)$$
$$= \mathrm{P}(\{Sum'[x] = 1\}` \cap \mathrm{P}\{\mathrm{Sum}\,[x] = 0\})$$
$$- \mathrm{P}(\{Sum'[x] = 0\} \cap \mathrm{P}\{\mathrm{Sum}\,[x] = 1\})$$
$$= s_x'\,(1 - s_x) - s_x(1 - s_x')$$
$$= s_x' - s_x$$
$$= s_x' - \frac{1}{2}, \ x\ \varepsilon\ \{0, 1, \dots, y-1\}$$
$$E(e[y]) = \mathrm{P}(\{C_{in}'[y] = 1\} \cap \mathrm{P}(\{C_{in}[y] = 0\}$$
$$-\mathrm{P}(\{C_{in}'[y] = 0\} \cap \mathrm{P}(\{C_{in}[y] = 1\}$$
$$= C_y' - C_y$$
$$= C_y' - \left(\frac{1}{2} - \frac{1}{2^{y+1}}\right)$$

Expressions for s_x' and C_y' were derived in Section IV-A. Thus, the mean error for the proposed approximations and truncation can be calculated as follows.

A. Approximation 1

$$\mu(\mathrm{y}) = -\frac{1}{6}\sum_{x=0}^{y-1} 2^x - \frac{1}{3}\sum_{x=0}^{y-1}\frac{1}{2^x} + \left(\frac{1}{6} - \frac{1}{3.2^{2y-1}} + \frac{1}{2^{y+1}}\right)2^y$$

$$\mu(\mathrm{y}) = -\frac{1}{6}\,(2^y - 1) - \frac{2}{3}\,(1 - 2^{-y}) + \left(\frac{2^y}{6} - \frac{1}{3.2^{2y-1}} + \frac{1}{2}\right)$$

$$= 0.$$

B. Approximation 2

$$\mu(\mathrm{y}) = \sum_{x=0}^{y-1}\frac{1}{2^{x+2}} \times 2^x + 0$$
$$= \frac{y}{4}.$$

C. Approximation 3

$$\mu(\mathrm{y}) = -\frac{1}{6}\sum_{x=0}^{y-1} 2^x + \frac{1}{3}\sum_{x=0}^{y-1}\frac{1}{2^{x+1}} + \left(\frac{1}{6} - \frac{1}{3.2^{2y-1}} + \frac{1}{2^{y+1}}\right)2^y$$

$$\mu(\mathrm{y}) = -\frac{1}{6}\,(2^y - 1) + \frac{1}{3}\,(1 - 2^{-y}) + \left(\frac{2^y}{6} - \frac{1}{3.2^{2y-1}} + \frac{1}{2}\right)$$

$$= 1 - 2^{-y}.$$

D. Approximation 4

$$\mu(\mathrm{y}) = -\frac{1}{2} + \frac{1}{8}\sum_{x=1}^{y-1} 2^x + \frac{1}{2} \times \frac{1}{2^{y+1}} \times 2^y$$

$$= \frac{1 - 2^{y-1}}{4}.$$

E. Approximation 5

$$\mu(\mathrm{y}) = 0 + \frac{1}{2^{y+1}} \times 2^y$$

$$= \frac{1}{2}.$$

F. Approximation 6

$$\mu(\mathrm{y}) = -\frac{1}{2}\sum_{x=0}^{y-1} 2^x + \left(\frac{1}{2} - \frac{1}{2^{y+1}}\right)2^y$$

$$= 1 - 2^{-y}$$

REFERENCES

[1] 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.

[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. Conf. Microelectron. Syst. Edu., Jul. 2009, pp. 45–48.

[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. 87–97, Jan. 2009.

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