

Quality Improvement and Power Reduction in Digital Signal Processing Using Approximate Adders

Malu. M. Edavackal¹, M. Mahalakshmi M.E.²

¹(Student, EC Department, Vivekanandha College of Engineering for Women, Tamil Nadu, India)

²(Assistant Professor, EC Department, Vivekanandha College of Engineering for Women, Tamil Nadu, India)

Abstract: Digital Signal Processing (DSP) blocks are the backbone of all multimedia applications and Low Power is an imperative requirement for portable multimedia devices handling different signal processing algorithms and architectures. Human beings can capture useful information even if the output is having some error. Thus, we need not produce exact numerical outputs. Previous researches provide the error resiliency through Voltage Over Scaling (VOS) by various algorithms and architectures to reduce the errors. In this paper, the logic complexity reduction is proposed by using approximate adders which is obtained by reducing the number of transistors without reducing the quality of the output signal. Simulation results shows the power reduction by using approximate adders compared to the actual full adder cells.

Keywords: Approximate Adders, Digital Signal Processing, Low Power, Mirror Adders.

I. Introduction

Digital signal processing (DSP) blocks are used in various multimedia applications for portable devices. These DSP blocks implement image and video processing algorithms, where the output is either an image or a video. This can be identified by human beings only with limited perceptual abilities. This allows the outputs of these algorithms to be numerically approximate or imprecise rather than accurate. This gives some freedom to carry out imprecise or approximate computation. We can use this freedom for different low-power designs in logic, algorithmic and architectural areas.

The paradigm of approximate computing is specific to select hardware implementations of DSP blocks. It is shown in [1] that an embedded reduced instruction set computing processor consumes 70% of the energy in supplying data and instructions, and 6% of the energy while performing arithmetic only. Therefore, using approximate arithmetic in such a scenario to reduce energy. Programmable processors are designed for general-purpose applications with no application-specific specialization. The target is most computationally intensive blocks are in these applications and build them using approximate hardware to show substantial improvements in power consumption with little loss in output quality.

Previous works that focus on low-power design through approximate computing at the algorithm and architecture levels include algorithmic noise tolerance (ANT) [3]–[6], significance driven computation (SDC) [7]–[9], and non-uniform voltage over scaling (VOS) [10]. All these techniques are based on the central concept of VOS, coupled with additional circuitry for correcting or limiting the resulting errors. In [11], a fast but “inaccurate” adder is proposed. It is based on the idea that on average, the length of the longest sequence of propagate signals is approximately $\log n$, where n is the bit-width of the two integers to be added. An error-tolerant adder is proposed in [12] that operates by splitting the input operands into accurate and inaccurate parts. However, neither of these techniques target logic complexity reduction. A power-efficient multiplier architecture is proposed in [13] that uses a 2×2 inaccurate multiplier block resulting from Karnaugh map simplification. This paper considers logic complexity reduction using Karnaugh maps. Shin and Gupta [14] and Phillips *et al.* [15] also proposed logic complexity reduction by Karnaugh map simplification. Other works that focus on logic complexity reduction at the gate level are Other approaches use complexity reduction at the algorithm level to meet real-time energy constraints .

Previous works on logic complexity reduction have focused on algorithm, logic, and gate levels. [1]We propose logic complexity reduction at the transistor level. We apply this to addition at the bit level by simplifying the mirror adder (MA) circuit. We design imprecise but simplified arithmetic units, which gives more power savings over conventional low-power design techniques. Complexity reduction leads to power reduction in two different ways. First, an inherent reduction in switched capacitance and number of transistors. Second, logic complexity reduction frequently leads to shorter critical paths, which leads to voltage reduction without any timing-induced errors. Since DSP blocks mainly consist of adders and multipliers (which are, in turn, built using adders), the propose system consists of several approximate adders, which can be used effectively in such blocks.

The main concepts of this paper are as follows:

1. The proposed system had logic complexity reduction at the transistor level as an alternative approach to approximate computing for DSP applications.
2. It is described that how to simplify the logic complexity of a conventional MA cell by reducing the number of transistors and internal node capacitances. Keeping this aim in mind, five different simplified versions of the MA, ensuring minimum errors in the full adder (FA) truth table is proposed.
3. Here utilized the simplified versions of the FA cell to propose several imprecise or approximate multi-bit adders that can be used as building blocks of DSP systems. To obtain a reasonable output quality, we need approximate FA cells in the Least Significant Bits (LSBs) only. This paper particularly focused on adder structures that use FA cells as their basic building blocks.
4. VOS is a very popular technique to get large improvements in power consumption. However, VOS will lead to delay failures in the Most Significant Bits (MSBs). This might lead to large errors in corresponding outputs and severely mess up the output quality of the application. We use approximate FA cells only in the LSBs, while the MSBs use accurate FA cells. Therefore, at isofrequency, the errors introduced by VOS will be much higher, when compared to proposed approximate adders. Since truncation is a well-known technique to facilitate voltage scaling, and compared the performance of proposed approximate adders with truncated adders.
5. The designs are proposed for image and video compression algorithms using the proposed approximate arithmetic units and evaluate the approximate architectures in terms of output quality by calculating area and power dissipation.

II. Proposed Approximate Adders

The Mirror Adder is one of the widely used economical implementations of a Full Adder (FA). MA is used as the basis for proposing different approximations of the FA cell.

2.1 Approximation Strategies For The MA

In this section, we explain step-by-step procedures for coming up with various approximate MA cells with fewer transistors. Removal of some series connected transistors will facilitate faster charging/discharging of node capacitances and the complexity reduction by the removal of transistors also aids in reducing the αC term (switched capacitance) in the dynamic power expression $P_{dynamic} = \alpha CV_{DD}^2 f$, where α is the switching activity or average number of switching transitions per unit time and C is the load capacitance being charged/discharged. This directly results in lower power dissipation. Area reduction is also achieved by this process. Now, let us discuss the conventional MA implementation followed by the proposed approximations.

2.1.1 Conventional Mirror Adder:

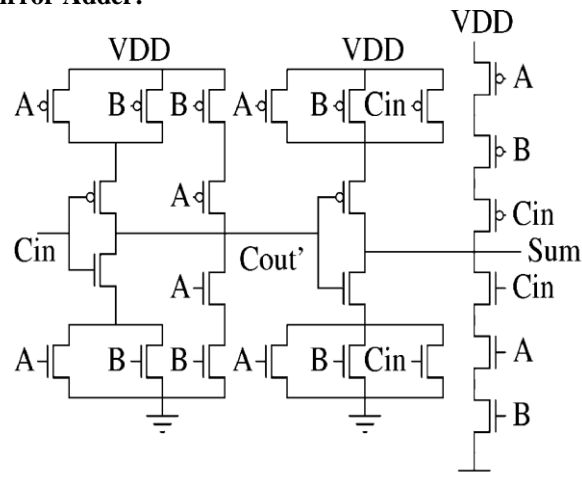


Fig.1. Conventional MA

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

2.1.2 Approximation 1:

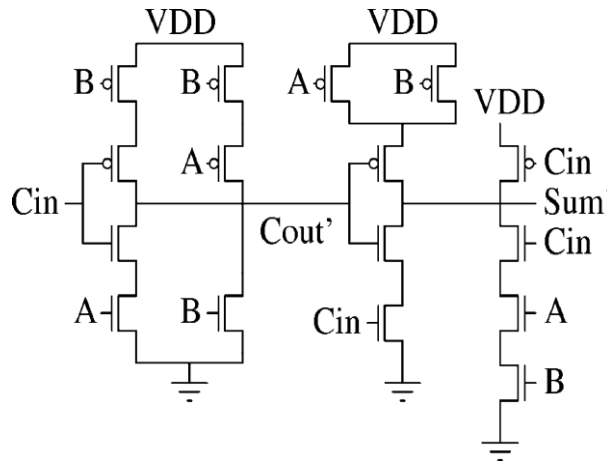


Fig.2. MA Approximation 1

In order to get an approximate MA with fewer transistors, we start to remove transistors from the conventional schematic one by one. However, we cannot do this in an arbitrary fashion. We need to make sure that any input combination of A, B and C_{in} does not result in short circuits or open circuits in the simplified schematic. Another important criterion is that the resulting simplification should introduce minimal errors in the FA truth table. A judicious selection of transistors to be removed (ensuring no open or short circuits) results in a schematic shown in Fig.2, which we call approximation 1. Clearly, this schematic has eight fewer transistors compared to the conventional MA schematic. In this case, there is one error in C_{out} and two errors in Sum, as shown in Table I. A tick mark denotes a match with the corresponding accurate output and a cross denotes an error.

TABLE I
Truth Table For Conventional Ma And Ma 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 ✓

2.1.3 MA Approximation 2:

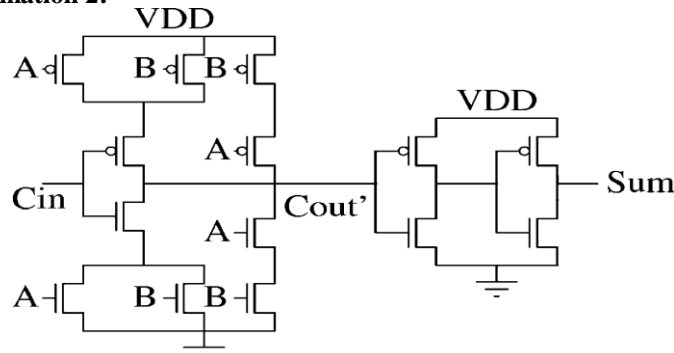


Fig.3. MA Approximation 2

The truth table of an FA shows that $Sum = C_{out1}$ 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, C_{out} is computed in the first stage. Thus, an easy way to get a simplified schematic is to set $Sum = C_{out}$. However, we introduce a buffer stage after C_{out} (see Fig.3) to implement the same functionality. The reason for this can be explained as

follows. If we set $Sum = C_{out}$ as it is in the conventional MA, the total capacitance at the Sum node would be a combination of four source-drain diffusion and two gate capacitances. This is a considerable increase compared to the conventional case or approximation 1. Such a design would lead to a delay penalty in cases where two or more multi-bit approximate adders are connected in series, which is very. Fig. 3 shows the schematic obtained using the above approach. We call this approximation 2. Here, Sum has only two errors, while Cout is correct for all inputs as per Table I.

2.1.4 MA Approximation 3:

Further simplification can be obtained by combining approximations 1 and 2. Note that this introduces one error in C_{out} and three errors in Sum, as shown in Table I. The corresponding simplified schematic is shown in Fig.4.

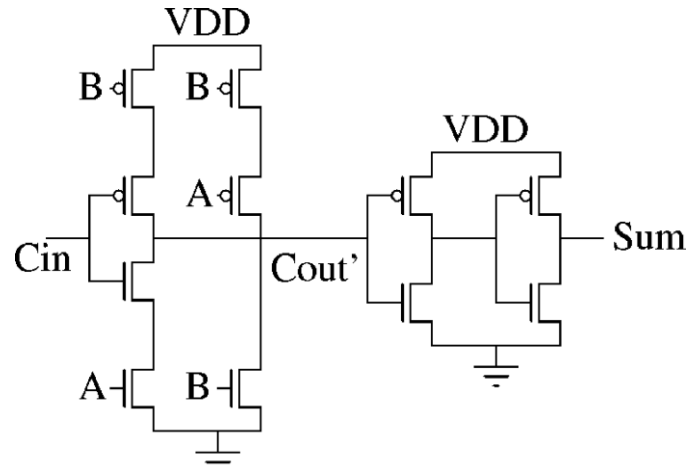


Fig.4. MA Approximation 4

2.1.5 MA Approximaion 5:

A close observation of the FA truth table shows that $C_{out} = A$ for six out of eight cases. Similarly, $C_{out} = B$ for six out of eight cases. Since A and B are interchangeable, we consider $C_{out} = A$. Thus, we propose a fourth approximation where we just use an inverter with input A to calculate C_{out} and Sum is calculated similar to approximation 1. This introduces two errors in C_{out} and three errors in Sum, as shown in Table I. The corresponding simplified schematic is shown in Fig.5. In all these approximations, C_{out} is calculated by using an inverter with C_{out} as input.

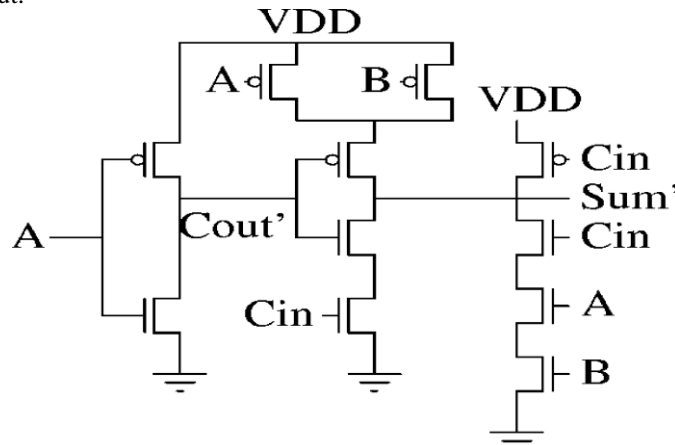


Fig.5. MA Approximation 4

Table 1 shows the truth table for the conventional full adder and mirror adder approximations. The inputs given are A, B, C_{in} and the outputs for conventional MA are obtained as Sum and Carry as accurate outputs. The approximations are done by creating errors in the accurate output Sum and carry as per the table 1. The approximate outputs are Sum_1 and C_{out1} for MA approximation1, Sum_2 and C_{out2} for MA approximation 2, Sum_3 and C_{out3} for MA approximation 3 and Sum_4 and C_{out4} for MA approximation 4.

III. Comparitive Performance Evaluation Through Simulation

The power is calculated by simulating the conventional MA and MA approximations 1-4 in Tanner EDA 13.0 and the corresponding area of the circuits are calculated in Microwind Software. The simulated results obtained are included here as snap shots.

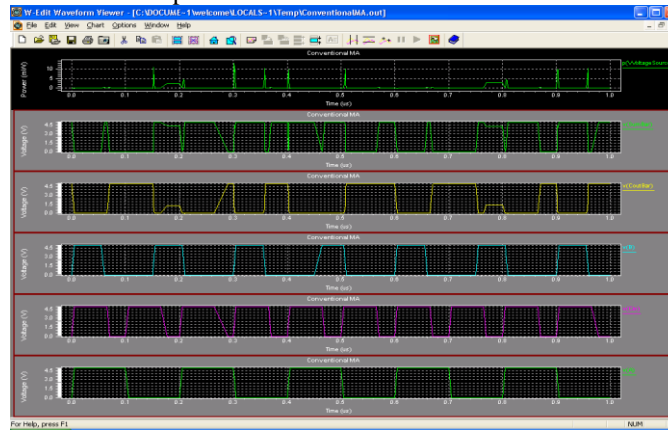


Fig.6. Conventional MA output

The inputs given are A, B, C_{in} and all are pulse signals. The number of transistors in use is 24. The “print voltage ” command is given to get the output voltages. The fig. 6 shows the output power waveform for the conventional MA. The maximum power obtained is 15 mW and average power is $3.427e-004$ W.

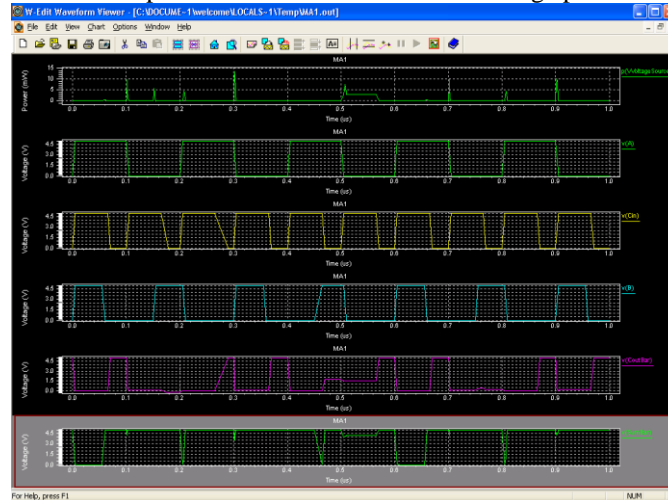


Fig.7. Approximation 1 output

The number of transistors is reduced to 16 in this approximation. In Sum 2 errors are introduced and in C_{out} only 1 error is introduced. The output waveform for power is obtained in fig.7 with maximum power as 14.8 mW and average power as $3.3326e-004$. The power is got reduced.

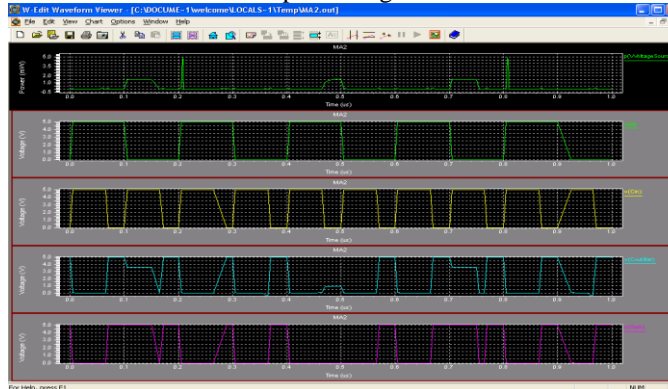


Fig.8. Approximation 2 output

The number of transistor is reduced to 14 and the Sum is having 2 errors and C_{out} is same as that of conventional MA. The output waveform for power is obtained as in fig.8 with a reduced average power of $2.497e-004$ W with a maximum power of 5 mW.

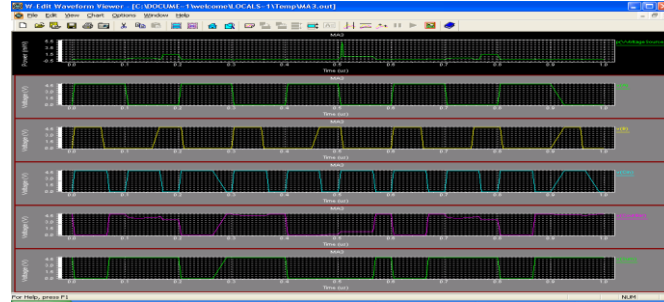


Fig.9. Approximation 3 output

In this approximation 3, the reduced number of transistors is 11 with 3 errors in Sum and 1 error in C_{out} . The power output waveform is obtained with an average power of $2.8305e-004$ W and the maximum power is obtained as 5.5 mW as shown in fig.9 given above.

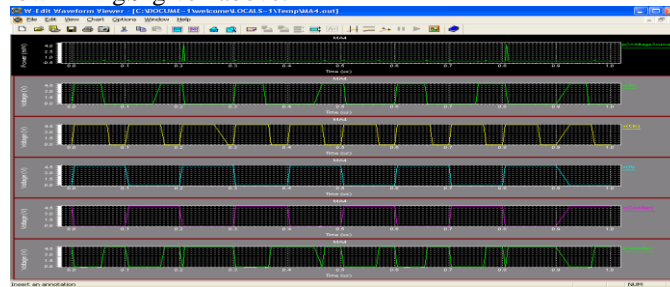


Fig.10. Approximation 4 output

In MA approximation 4, the number of transistors is 11 with 3 errors in Sum and 2 errors in C_{out} . The average power is reduced to $3.5273e-005$ W with a maximum power of 4.5 mW is obtained for approximation 4 as per the output power waveform shown in fig.10.

TABLE II
Average Power Estimation

Mirror Adder Type	Maximum Power Consumption (mW)	Average Power
Conventional MA	15	($3.427e-004$)W
MA Approximation-1	14.8	($3.33268e-004$)W
MA Approximation-2	5	($2.497 e-004$)W
MA Approximation-3	5.5	($2.83051e-004$)W
MA Approximation-4	4.5	($3.5273e-005$)W

The table 2 show the maximum power consumption of conventional MA, MA approximation1, MA approximation 2, MA approximation 3, MA approximation 4 and it is clear that the maximum power is consumed is 15 mW by the conventional FA. The minimum power consumed is 4.5 mW and is consumed by the MA approximation 4. As the approximation level increases the power consumption is getting reduced. The average power obtained is also shown in the table.

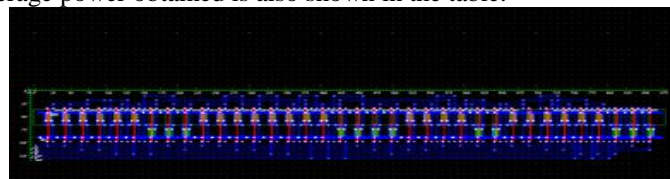


Fig.11. layout of Conventional MA

The layout of conventional MA is given in fig.11. The area is obtained as $850*130 = 11.37 \mu\text{m}^2$ in microwind 2.0 software .

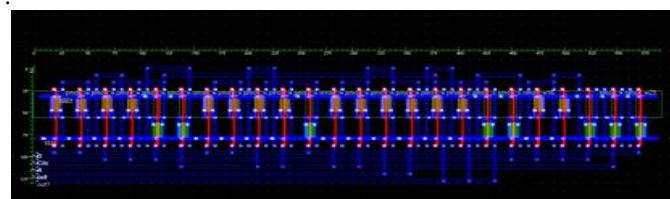


Fig.12. Layout of MA Approximation 1

The area for MA approximation 1 is obtained as $575*130 = 7.47 \mu\text{m}^2$ as shown in fig.12. The area is reduced compared to the conventional MA.

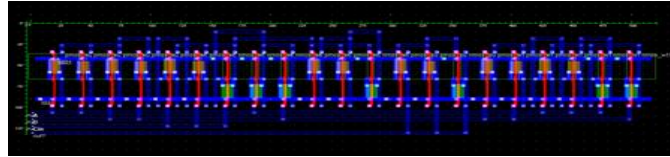


Fig.13. Layout of MA Approximation 2

Fig.13 shows the area for MA approximation 2 and is obtained as $525 \times 130 = 6.8 \mu\text{m}^2$ in microwind software. The area is reduced compared to approximation 1.

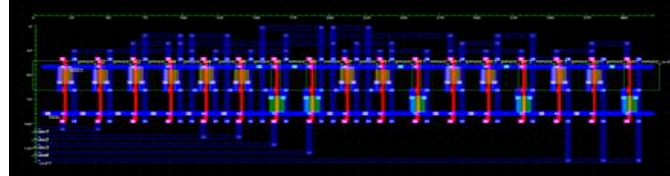


Fig.14. Layout of MA Approximation 3

The area for MA approximation 3 is obtained from the fig.14 as $425 \times 130 = 5.5 \mu\text{m}^2$. The area is obtained from the layout drawn in microwind software and is reduced than other approximations.

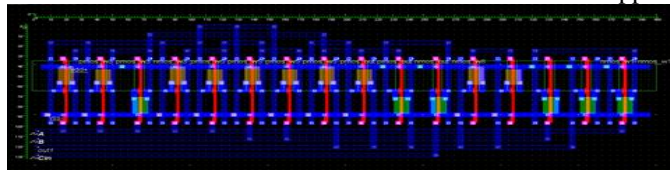


Fig.15. Layout of MA Approximation 4

Fig.15 shows the layout drawn in microwind software for MA approximation 4 and the area is obtained as $400 \times 130 = 5.2 \mu\text{m}^2$ which is very less than other approximations.

TABLE III
Area Analysis

Mirror Adder Type	Area(μm^2)
Conventional MA	11.37
MA Approximation-1	7.47
MA Approximation-2	6.8
MA Approximation-3	5.5
MA Approximation-4	5.2

The area is calculated for the Conventional MA, MA approximations 1-4 and as per the result obtained the maximum area is $11.37 \mu\text{m}^2$ for conventional MA. As the approximation level increases the area also getting reduced like power and the minimum area is $5.2 \mu\text{m}^2$ for the MA approximation 4. Thus it is concluded that the area and power are reduced by using approximate adders and this will increase the quality of the DSP block which was made up by these approximate adders.

IV. Conclusion

In this paper, we proposed several imprecise or approximate adders that can be effectively utilized to trade off power, area and thereby 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 are 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 where errors were introduced due to VOS [3]–[10]. A decrease in the number of series connected transistors helped in reducing the effective switched capacitance and achieving voltage scaling. We believe that the proposed approximate adders can be used on top of already existing low-power techniques like SDC and ANT to extract multi-fold benefits with a very minimal loss in output quality.

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] 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.
- [10] A. K. Verma, P. Brisk, and P. Jenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in *Proc. Design, Automat. Test Eur.*, 2008, pp. 1250–1255.
- [11] 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.
- [12] P. Kulkarni, P. Gupta, and M. D. Ercegovic, "Trading accuracy for power in a multiplier architecture," *J. Low Power Electron.*, vol. 7, no. 4, pp. 490–501, 2011.
- [13] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in *Proc. Design, Automat. Test Eur.*, 2010, pp. 957–960.
- [14] 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.
- [15] 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.
- [16] S.-L. Lu, "Speeding up processing with approximation circuits," *Computer*, vol. 37, no. 3, pp. 67–73, Mar. 2004.
- [17] 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.