

# Pushing the Limits of Voltage Over-Scaling for Error-Resilient Applications

Rengarajan Ragavan, Benjamin Barrois, Cedric Killian  
Univ. Rennes 1 – IRISA/INRIA  
{rengarajan.ragavan, benjamin.barrois, cedric.killian}@irisa.fr

Olivier Sentieys  
INRIA/IRISA  
olivier.sentieys@inria.fr

**Abstract**—Voltage scaling has been used as a prominent technique to improve energy efficiency in digital systems, scaling down supply voltage effects in quadratic reduction in energy consumption of the system. Reducing supply voltage induces timing errors in the system that are corrected through additional error detection and correction circuits. In this paper we are proposing voltage over-scaling based approximate operators for applications that can tolerate errors. We characterize the basic arithmetic operators using different operating triads (combination of supply voltage, body-biasing scheme and clock frequency) to generate models for approximate operators. Error-resilient applications can be mapped with the generated approximate operator models to achieve optimum trade-off between energy efficiency and error margin. Based on the dynamic speculation technique, best possible operating triad is chosen at runtime based on the user definable error tolerance margin of the application. In our experiments in 28nm FDSOI, we achieve maximum energy efficiency of 89% for basic operators like 8-bit and 16-bit adders at the cost of 20% Bit Error Rate (ratio of faulty bits over total bits) by operating them in near-threshold regime.

## I. INTRODUCTION

Scaling techniques have evolved and been explored in greater extent over the time to unlock the opportunities of higher energy efficiency by operating the transistors near or below the threshold voltage [1], [2]. After the advent of inherent low-leakage technologies like FDSOI (Fully Depleted Silicon On Insulator), near-threshold computing has gained more importance in VLSI due to improved resistance towards various variability effects like Random Dopant Fluctuations (RDF). Body-biasing technique in FDSOI provides greater flexibility to control trade-off between performance and energy efficiency based on the application need. In spite of the improvement in techniques and technology, near-threshold computing is still seen as no go zone for conventional sub-nanometer designs, due to timing errors introduced by the supply voltage scaling and need for additional hardware such as double-sampling registers [3] to detect and correct such timing errors. There are techniques, like algorithmic noise-tolerance based error correction approach proposed in [4], which contain error correcting circuit along with computing circuit to handle errors due to drastic reduction in  $V_{dd}$ .

Error-resilient computing is an emerging trend in VLSI, in which accuracy of the computing can be traded to improve the energy efficiency and to lower the silicon footprint of the design [5]. Emerging classes of applications based on statistical and probabilistic algorithms used for video processing, image recognition, text and data mining, machine learning, have the inherent ability to tolerate hardware uncertainty. Such error-

resilient applications that can live with errors, void the need for additional hardware to detect and correct errors. Also, error-resilient applications provide an opportunity to design approximate hardware to meet the computing needs with higher energy efficiency and tolerable accuracy loss. In error-resilient applications, approximations in computing can be introduced at different stages of computing and at varying granularity of the design. Using probabilistic techniques, computations can be classified as significant and non-significant at different design abstraction levels like algorithmic, architecture, and circuit levels. In this paper, we are targeting circuit-level approximation, which can further be extended to algorithmic level by using the proposed statistical model for approximate operators. This work makes the following contributions:

- We characterize energy efficiency and accuracy of different adder configurations using several operating triads (supply voltage, body-biasing voltage, clock period).
- We formulate a framework to model the statistical behavior of arithmetic operators subjected to voltage over-scaling that can be used at algorithmic level.
- Simulation results show that a maximum of 89% energy reduction is achieved at the tolerable output bit error rate of 20%.

The paper is organized as follows, Section II presents some of the existing approximate operators and modelling techniques. Characterization of arithmetic operators is discussed in Section III, followed by modelling of approximate operators in Section IV. Section V charts out the experimental setups and results, finally Section VI gives the conclusion and perspectives.

## II. APPROXIMATION IN ARITHMETIC OPERATORS

Approximations in arithmetic operators are broadly classified based on the level at which approximations are introduced [5]. This section reviews methods proposed in the literature at physical and architectural levels.

In [6], operators of a Functional Unit (FU) are characterized by analyzing relationship between  $V_{dd}$  scaling and BER (Bit Error Rate, ratio of faulty output bits over total output bits). Based on the characterization, for every FU in the pipeline, one more *imprecise* FU running at lower  $V_{dd}$  is designed. According to the application's need, selected set or all the FUs in the pipeline are accompanied by an *imprecise* counterpart in the design. Based on the user defined precision level, computations are performed either by *precise* or *imprecise* FU in the pipeline. Instead of duplicating every

FU with an imprecise counterpart, a portion of the FU is replaced by imprecise or approximate design as discussed in [5]. Fig. 1 shows this principle where least significant inputs are processed by *approximate* operator and most significant inputs are processed by *accurate* operator to increase the energy efficiency at the cost of acceptable accuracy loss. In [7]  $n$ -bit RCA adder based on near-threshold computing is proposed with two parts;  $k$ -bit LSBs approximated while  $(n - k)$  bits computed by precise RCA.

Another class of physical-level approximation is achieved by applying dynamic voltage and frequency scaling to an accurate operator, as depicted in Fig. 2. Due to the dynamic control of voltage and frequency, timing errors due to scaling can be controlled flexibly in terms of trade-off between accuracy and energy. This method is referred as Voltage Over-Scaling (VOS) in [6]. Similar to VOS, clock overgating based approximation is introduced in [8]. Clock overgating is done by gating clock signal to selected flip-flops in the circuit during execution cycles in which the circuit functionality is sensitive to their state. In all the approximation methods at physical level, in addition to the deliberate approximation introduced, impact of variability has to be considered to achieve optimum balance between accuracy and energy. Decoupling the data and control processing is proposed in [9] to mitigate the impact of variation in near-threshold approximate designs. Also, technologies like FDSOI provide good resistance towards the impact of variability.

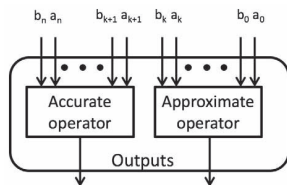


Fig. 1: Accurate approximate configuration of [5]

Approximation at architectural level is discussed in [10], where accuracy control is handled by bitwidth optimization and scheduling algorithms. Also other forms of architectural-level approximations are discussed in [11], where probabilistic pruning based approximation method is proposed. In this method, design is optimized by removing certain hardware components of the design and/or by implementing alternate way to perform the same functionality with reduced accuracy. In [12], a probabilistic approach is discussed in the context of device modeling and circuit design. In this method, noise is added to the input and output nodes of an inverter and the probability of error is calculated by comparing the output of the inverter with a noise-free counterpart. In [13], new class of pruned speculative adder are proposed by adding gate-level pruning in speculative adders to improve Energy Delay Area Product (EDAP). Though there is claim that the pruned speculative adder will show higher gains when operated at sub-threshold region, no solid justification is given in [13]. In general, approximations introduced by pruning methods are more rigid in nature, which lacks the dynamicity to switch between various energy-accuracy trade-off points.

On contrast, voltage scaling based approximations are more flexible, easy to implement and offer dynamic control over energy-accuracy trade-off. Approximation introduced by sup-

ply voltage scaling offers *dynamic approximation*, by changing the operating triad (combination of supply voltage, body-biasing scheme, and clock frequency) of the design at runtime, which makes the user to control the energy-accuracy trade-off efficiently. In [14], limitations of voltage over-scaling based approximate adders such as need for level shifters and multiple voltage routing lines are mentioned. These limitations can be overcome by employing uniform voltage scaling along the pipeline or at larger granularity.

### III. CHARACTERIZATION OF ARITHMETIC OPERATORS

In this section, characterization of arithmetic operators is discussed for voltage over-scaling based approximation, as shown in Fig. 2. As adder is the most common operator used in datapaths, we consider Ripple Carry Adder (RCA) and Brent-Kung Adder (BKA) configurations.

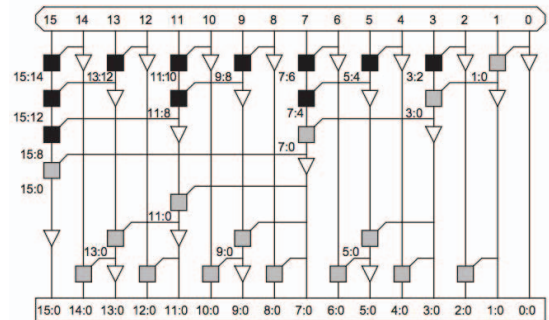


Fig. 3: Carry Chain of Brent-Kung Adder [15]

Ripple carry adder is a sequence of full adders with serial prefix based addition. RCA takes  $n$  stages to compute  $n$ -bit addition. In contrast, Brent-Kung adder is a parallel prefix adder. Fig. 3, shows the carry chain of Brent-Kung adder. BKA takes  $2 \log_2(n - 1)$  stages to compute  $n$ -bit addition. In BKA, carry generation and propagation are segmented into smaller paths and executed in parallel. Black and gray cells in Fig. 3 represent carry generate and propagate, respectively.

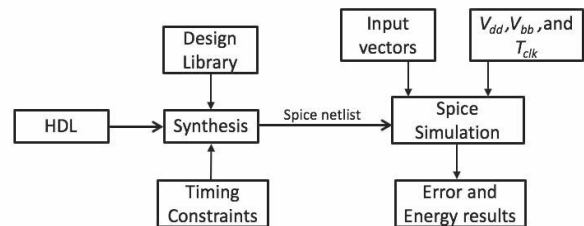


Fig. 4: Flow of arithmetic operator characterization

Fig. 4 shows the characterization flow of the arithmetic operators. Structured gate-level HDL is synthesized with user-defined constraints. The output netlist is then simulated at transistor level using SPICE (Simulation Program with Integrated Circuit Emphasis) platform by varying operating triads ( $V_{dd}$ ,  $V_{bb}$ ,  $T_{clk}$ ), where  $V_{dd}$  is supply voltage,  $V_{bb}$  is body-biasing voltage, and  $T_{clk}$  is clock period. In ideal condition, the arithmetic operator functions without any errors. Also, EDA tools introduce additional timing margin in the datapaths during Static Timing Analysis (STA) due to clock path pessimism.

This additional timing prevents timing errors due to variability effects. Due to the limitation in availability of design libraries for near/sub-threshold computing, it is necessary to use SPICE simulation to understand the behaviour of arithmetic operators in different voltage regimes. By tweaking the operating triads, timing errors  $e$  are invoked in the operator and can be represented as

$$e = f(V_{dd}, V_{bb}, T_{clk}) \quad (1)$$

Characterization of arithmetic operator helps to understand the point of generation and propagation of timing errors in arithmetic operators. Among the three parameters in the triad, scaling  $V_{dd}$  causes timing errors due to the dependence of operator's propagation delay  $t_p$  on  $V_{dd}$ , such as

$$t_p = \frac{V_{dd} \cdot C_{load}}{k(V_{dd} + V_t)^2} \quad (2)$$

Body-biasing potential  $V_{bb}$  is used to vary the threshold voltage ( $V_t$ ); thereby increasing the performance (decreasing  $t_p$ ) or reducing leakage of the circuit. Due to the dependence of  $t_p$  on  $V_t$ ,  $V_{bb}$  is used solely or in tandem with  $V_{dd}$  to control timing errors. Scaling down  $V_{dd}$  improves the energy efficiency of the operator due to its quadratic dependence to total energy.  $E_{total} = V_{dd}^2 \cdot C_{load}$ . Mere scaling down  $F_{clk}$  does not reduce the energy consumption, though it will reduce the total power consumption of the circuit. Therefore,  $F_{clk}$  is scaled along with  $V_{dd}$  and  $V_{bb}$  to achieve high energy efficiency.

Behaviour of arithmetic operator in near/sub-threshold region is different from the super-threshold region. In case of an RCA, when the supply voltage is scaled down, the expected behaviour is failure of critical path(s) from longest to the shortest with respect to the reduction in the supply voltage. Fig. 5 shows the effect of voltage over-scaling in 8-bit RCA. When the supply voltage is reduced from 1V to 0.8V, MSBs starts to fail. As the voltage is further reduced to 0.7V and 0.6V more BER is recorded in middle order bits rather than most significant bits. For 0.5V  $V_{dd}$ , all the middle order bits reaches BER of 50% and above. Similar behaviour is observed in 8-bit BKA for different values of  $V_{dd}$ . This behaviour imposes limitations in modelling approximate arithmetic operators in near/sub-threshold using standard models. Behaviour of arithmetic operators during voltage over-scaling in near/sub-threshold region can be characterized by SPICE simulations. But SPICE simulators take long time to simulate exhaustive set of input patterns needed to characterize arithmetic operators.

#### IV. MODELLING OF FAULTY ARITHMETIC OPERATORS

As stated previously, there is a need to develop models that can simulate the behavior of faulty arithmetic operators at functional level. In this section, we propose a new modelling technique that is scalable for large-size operators and compliant with different arithmetic configurations. The proposed model is accurate and allows fast simulations at the algorithm level by imitating the faulty operator with statistical parameters.

As VOS provokes failures on the longest combinatory datapaths in priority, there is clearly a link between the impact of the carry propagation path on a given addition and the error issued from this addition. Figure 6 illustrates the needed relationship between hardware operator controlled by operating triads and statistical model controlled by statistical parameters

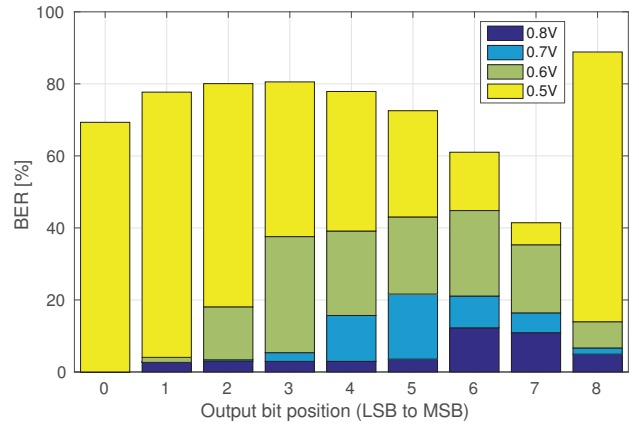


Fig. 5: Distribution of BER in output bits of 8-bit RCA under voltage scaling

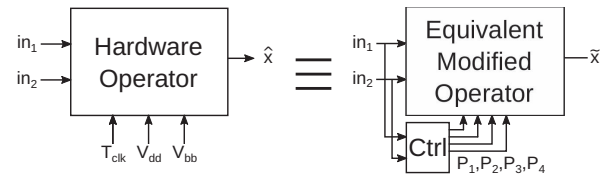


Fig. 6: Functional equivalence for hardware adder

$P_i$ . As the knowledge of the inputs gives necessary information about the longest carry propagation chain, the values of the inputs are used to generate the statistical parameters that control the equivalent model. These statistical parameters are obtained through an offline optimization process that minimizes the difference between the outputs of the operator and its equivalent statistical model, according to a certain metric. In this work, we used three accuracy metrics to calibrate the efficiency of the proposed statistical model:

- Mean Square Error (MSE) – average of squares of deviations between the output of the statistical model  $\tilde{x}$  and the reference  $\hat{x}$ .
- Hamming distance – number of positions with bit flip between the output of the statistical model  $\tilde{x}$  and the reference  $\hat{x}$ .
- Weighted Hamming distance – Hamming distance with weight for every bit position depending on their significance.

In the rest of the section, we develop a proof of concept by applying VOS on different adder configurations. All the adder configurations are subjected to VOS and characterized as shown in Fig. 4. In the case of adder, only one parameter  $P_i$  for the statistical model is used and is defined as  $C_{max}$ , the length of the maximum carry chain to be propagated. Hence, given the operating parameters ( $T_{clk}, V_{dd}, V_{bb}$ ) and a couple of inputs ( $in_1, in_2$ ), the goal is to find  $C_{max}$ , minimizing the distance between the output of the hardware operator and the equivalent modified adder. This distance can be defined by the above listed accuracy metrics. Hence,  $C_{max}$  is given by:

$$C_{max}(in_1, in_2) = \underset{C \in [0, N]}{\text{Argmin}} \|\hat{x}(in_1, in_2), \tilde{x}(in_1, in_2)\|$$

where  $\|x, y\|$  is the chosen distance metric applied to  $x$  and  $y$ . As the search space for characterizing  $C_{max}$  for all sets of

inputs is potentially very high,  $C_{max}$  is characterized only in terms of probability of appearing as a function of the theoretical maximal carry chain of the inputs, denoted as  $P(C_{max} = k | C_{max}^{th} = l)$ . This way, the mapping space of  $2^{2N}$  possibilities is reduced to  $(N + 1)^2/2$ . Table I gives the template of the probability values needed by the equivalent modified adder to produce an output.

TABLE I: Carry propagation probability table of modified 4-bit adder

$C_{max}$ \ $C_{max}^{th}$	0	1	2	3	4
0	1	$P(0 1)$	$P(0 2)$	$P(0 3)$	$P(0 4)$
1	0	$P(1 1)$	$P(1 2)$	$P(1 3)$	$P(1 4)$
2	0	0	$P(2 2)$	$P(2 3)$	$P(2 4)$
3	0	0	0	$P(3 3)$	$P(3 4)$
4	0	0	0	0	$P(4 4)$

The optimization algorithm used to construct the modified adder is shown in Algorithm 1. When the inputs ( $in_1, in_2$ ) are in the vector of training inputs, output of the hardware adder configuration  $\hat{x}$  is computed. Based on the particular input pair ( $in_1, in_2$ ), maximum carry chain  $C_{max}^{th}$  corresponding to the input pair is determined. Output  $\tilde{x}$  of the modified adder with three input parameters ( $in_1, in_2, C$ ) is computed. The distance between the hardware adder output  $\hat{x}$  and modified adder output  $\tilde{x}$  is calculated based on the above defined accuracy metrics for different iterations of  $C$ . The flow continues for the entire set of training inputs.

```

begin
   $P(0 : N_{bit\_adder} | 0 : N_{bit\_adder}) := 0;$ 
   $max\_dist := +\infty;$ 
   $C_{max\_temp} := 0;$ 
  for variable  $in_1, in_2 \in training\_inputs$  do
     $\hat{x} := add\_hardware(in_1, in_2)$ 
     $C_{max}^{th} := max\_carry\_chain(in_1, in_2);$ 
    for variable  $C \in C_{max}^{th}$  down to 0 do
       $\tilde{x} := add\_modified(in_1, in_2, C);$ 
       $dist := \|\hat{x}, \tilde{x}\|;$ 
      if  $dist \leq max\_dist$  then
         $dist\_max := dist;$ 
         $C_{max\_temp} := C;$ 
      end
    end
  end
   $P(C_{max\_temp} | C_{max}^{th}) + +$ 
end
 $P(:, :) := P(:, :)/size(training\_outputs);$ 
end

```

Algorithm 1: Optimization Algorithm

After the off-line optimization process is performed, the equivalent modified adder can be used to generate the outputs corresponding to any couple of inputs  $in_1$  and  $in_2$ . To imitate the exact operator subjected to VOS triads, the equivalent adder is used in the following way:

- 1) Extract the theoretical maximal carry chain  $C_{max}^{th}$  which would be produced by the exact addition of  $in_1$  and  $in_2$ .
- 2) Pick of a random number, choose the corresponding

row of the probability table, in the column representing  $C_{max}^{th}$ , and assign this value to  $C_{max}$ .

- 3) Compute the sum of  $in_1$  and  $in_2$  with a maximal carry chain limited to  $C_{max}$ .

Fig. 7 shows the estimation error of model of different adders based on the above defined accuracy metrics. SPICE simulations are carried out in 43 operating triads with 20K input patterns. Input patterns are chosen in such a way that all the input bits carry equal probability to propagate carry in the chain. Fig. 7a plots the Signal to Noise Ratio (SNR) of 8- and 16-bit RCA and BKA adders. MSE distance metric shows higher mean SNR, followed by Hamming distance and weighted Hamming distance metrics. Since MSE and weighted Hamming distance are taking the significance of bits into account, their resulting mean SNRs are higher than for the Hamming distance metric. Fig. 7b shows the plot of normalized Hamming distance of all the four adders. In this plot, MSE and Hamming distance metrics are almost equal, with a slight advantage for non-weighted Hamming distance, which is expected since this metric gives all bit positions the same impact. Both the 8-bit adders have same behaviour in terms of the distance between output of hardware adder and modified adder. On the other hand, 16-bit RCA is better in terms of SNR compared to its BKA counterpart. These results demonstrate the accuracy of the proposed approach to model the behavior of operators subjected to VOS in terms of approximation.

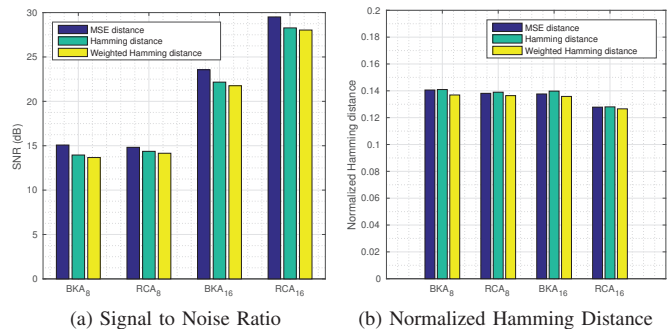


Fig. 7: Estimation error of the model for different adders and distance metrics

## V. EXPERIMENTS AND ENERGY EFFICIENCY RESULTS

In our experiments, we characterized 8- and 16-bit ripple carry adder (RCA) and Brent-Kung adder (BKA) using LVT (Low  $V_t$ ) transistor libraries of 28nm-FDSOI technology. Table II shows the synthesis results (area, static plus dynamic power, critical path) of different adder configurations at 1V  $V_{dd}$  without body bias. Post synthesis, SPICE netlist of all the adders are generated and simulated using Eldo SPICE (version 12.2a). Table III shows the different operating triads used to simulate the adders. Clock period ( $T_{clk}$ ) of the adders is chosen based on the synthesis timing report. Supply voltage ( $V_{dd}$ ) of all the simulations are scaled down from 1.0V to 0.4V in steps of 0.1V and body-biasing potential ( $V_{bb}$ ) of -2V, 0V, and, 2V. Pattern source function of SPICE testbench is configured with specific input vectors to test the adder configurations. Circuit under test is subjected to 20K simulations for every

different operating triad with same set of input patterns. Energy per operation corresponding to different operating triads is calculated from the simulation results. Output values generated from the SPICE simulation are compared against the golden (ideal) outputs corresponding to the input patterns. Automated test scripts calculate various statistical parameters like BER (ratio of faulty output bits over total output bits), MSE and bit-wise error probability (ratio of number of faulty bits over total bits in every binary position) for all the test cases.

TABLE II: Synthesis Results of 8 and 16 bit RCA and BKA

Benchmarks	Area ( $\mu m^2$ )	Total Power ( $\mu W$ )	Critical Path (ns)
8-bit RCA	114.7	170	0.28
8-bit BKA	174.1	267.7	0.19
16-bit RCA	224.5	341	0.53
16-bit BKA	265.5	363.4	0.25

TABLE III: Operating triads used in Spice simulation

Benchmarks	$T_{clk}$ (ns)	$V_{dd}$ (V)	$V_{bb}$ (V)
8-bit RCA	0.5, 0.28, 0.19, 0.13	1 to 0.4	-2 to 2
8-bit BKA	0.5, 0.19, 0.13, 0.064	1 to 0.4	-2 to 2
16-bit RCA	0.7, 0.53, 0.25, 0.20	1 to 0.4	-2 to 2
16-bit BKA	0.7, 0.25, 0.20, 0.15	1 to 0.4	-2 to 2

Fig. 8a and Fig. 8b show the plots of BER vs Energy/Operation of 8-bit RCA and BKA adders. Likewise plots of BER vs Energy/Operation of 16-bit RCA and BKA adders are shown in Fig. 8c and Fig. 8d respectively. The label of  $x$ -axis of the plots show the operating triads in the format  $T_{clk}$  (ns),  $V_{dd}$  (V), and  $V_{bb}$  (V) respectively. In all the adder configurations, energy/operation decreases and BER increases in sync with the supply voltage over-scaling. Table IV shows the maximum energy efficiency (amount of energy saving compared to ideal test case) achieved by 8-bit and 16-bit RCA and BKA in different BER ranges. Due to the parallel prefix structure, BKA adders show staircase pattern in BER plot shown in Fig. 8b and Fig. 8d. On other hand, RCA adders based on serial prefix show exponential pattern in BER plot.

Energy/operation curve of all the four plots show two patterns corresponding to 0% BER, and BER greater than 0%. Effect of voltage over-scaling is visible in the left half of the plots, where energy/operation is gradually reduced in sync with reduction in  $V_{dd}$  while BER is at 0%. Another important observation is that the effect of body-biasing is helping to keep the BER at 0% in this region of the plot. Both the 8-bit RCA and BKA adders operated at 0.5V  $V_{dd}$  with forward body bias of 2V  $V_{bb}$ , achieve maximum energy efficiency of 76% and 75% respectively at 0% BER. Similarly, 16-bit RCA and BKA achieve maximum energy efficiency of 60% and 59% respectively at 0% BER, while operating at 0.6V for  $V_{dd}$  with forward body bias  $V_{bb}$  of 2V. This set of operating triads provides high energy efficiency without any loss in accuracy of the computation by taking advantage of near-threshold computing and body-biasing technique.

On the right half of the BER vs Energy/Operation plots, where the BER is greater than 0%, energy curve starts in three branches and tapers down when the BER reaches 40% and above. In those three branches, operating triads with body-biasing are most energy efficient followed by triads without

body-biasing and finally triads with overclocking. In 8-bit BKA adder, 28 out of 43 operating triads operate within 0% to 25% BER. Similarly, 36 triads operate within 0% to 25% BER in 8-bit RCA adder. In 16-bit BKA and RCA adders, correspondingly 30 and 38 operating triads operate within BER range of 0% to 25%. For an application with acceptable error margin of 25%, 8-bit RCA, 8-bit BKA, 16-bit RCA, and 16-bit BKA can be operated at 0.4V  $V_{dd}$  with forward body bias  $V_{bb}$  2V to achieve maximum energy efficiency of 92%, 89%, 90.8%, and 84%, respectively.

Approximation in arithmetic operators based on voltage over-scaling, provides *dynamic approximation*, which makes the user to control the energy-accuracy trade-off efficiently by changing the operating triad of the design at runtime. In this method, *dynamic approximation* can be achieved without any design-level changes or addition of extra logic in the arithmetic operators unlike accuracy configurable adder proposed in [16]. Dynamic speculation techniques like in [17] can be used to estimate the BER at runtime to switch between different triads to achieve high energy efficiency with respect to user defined error margin. 8-bit RCA and BKA can be dynamically switched from accurate to approximate mode by merely scaling down  $V_{dd}$  from 0.5V to 0.4V at the cost of 8% BER to increase energy efficiency from 76% to 87%. Similarly in 8-bit RCA, switching from accurate to approximate mode is possible by reducing  $V_{dd}$  from 0.5V to 0.4V at the cost of 16% BER to increase energy efficiency from 75% to 89%. BKA adder configuration records more BER compared to RCA because of more logic paths of same length due to parallel prefix structure. Likewise in 16-bit RCA, accurate to approximate mode can be switched by scaling  $V_{dd}$  from 0.6V to 0.4V at the cost of 6% BER to increase energy efficiency from 60% to 84%. In 16-bit BKA, accurate to approximate mode can be switched by scaling  $V_{dd}$  from 0.6V to 0.4V at the cost of 9% BER to increase energy efficiency from 59% to 84%. Both 16-bit adders provide leap of 24% increase in energy efficiency at the maximum cost of 9% BER compared to accurate mode.

## VI. CONCLUSION

In this paper, we have proposed to use voltage over-scaling to highlight possible trade-off between energy efficiency and approximation in arithmetic operators that can be used for error-resilient applications. In this work, we characterized different configurations of adders using different operating triads to generate statistical model for approximate adder. We have laid down the framework to construct statistical model by characterizing approximate operators based on voltage over-scaling. We have achieved maximum energy efficiency of 76% in 8-bit RCA while operating at 0.5V  $V_{dd}$  without any accuracy loss. By increasing the effect of voltage over-scaling from 0.5V to 0.4V, energy efficiency is increased to 87% at the cost of 8% BER. All the adder configurations have shown maximum energy gains of up to 89% within 16% of BER and 92% within 22% of BER. Dynamic approximation can be used in these adders by employing dynamic speculation methods to monitor the errors at runtime.

## REFERENCES

- [1] T. Kuroda, K. Suzuki, *et al.*, "Variable supply-voltage scheme for low-power high-speed cmos digital design." *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 454–462, 1998.

TABLE IV: Energy Efficiency and BER in 8-bit and 16-bit Ripple Carry and Brent-Kung Adders

BER Range	Number of Triads				Max. Energy Efficiency (%)				BER at Max. Energy Efficiency (%)			
	8-RCA	8-BKA	16-RCA	16-BKA	8-RCA	8-BKA	16-RCA	16-BKA	8-RCA	8-BKA	16-RCA	16-BKA
0%	16	14	15	18	76	75.3	60.5	73.3	0	0	0	0
1% to 10%	15	7	15	9	87	65.3	83.6	84	8	8.8	6	9.1
11% to 20%	2	5	6	3	74	89	86.2	73.3	11	16.1	17.5	18.1
21% to 25%	3	2	2	–	92	82.8	90.8	–	22	25	22.1	–

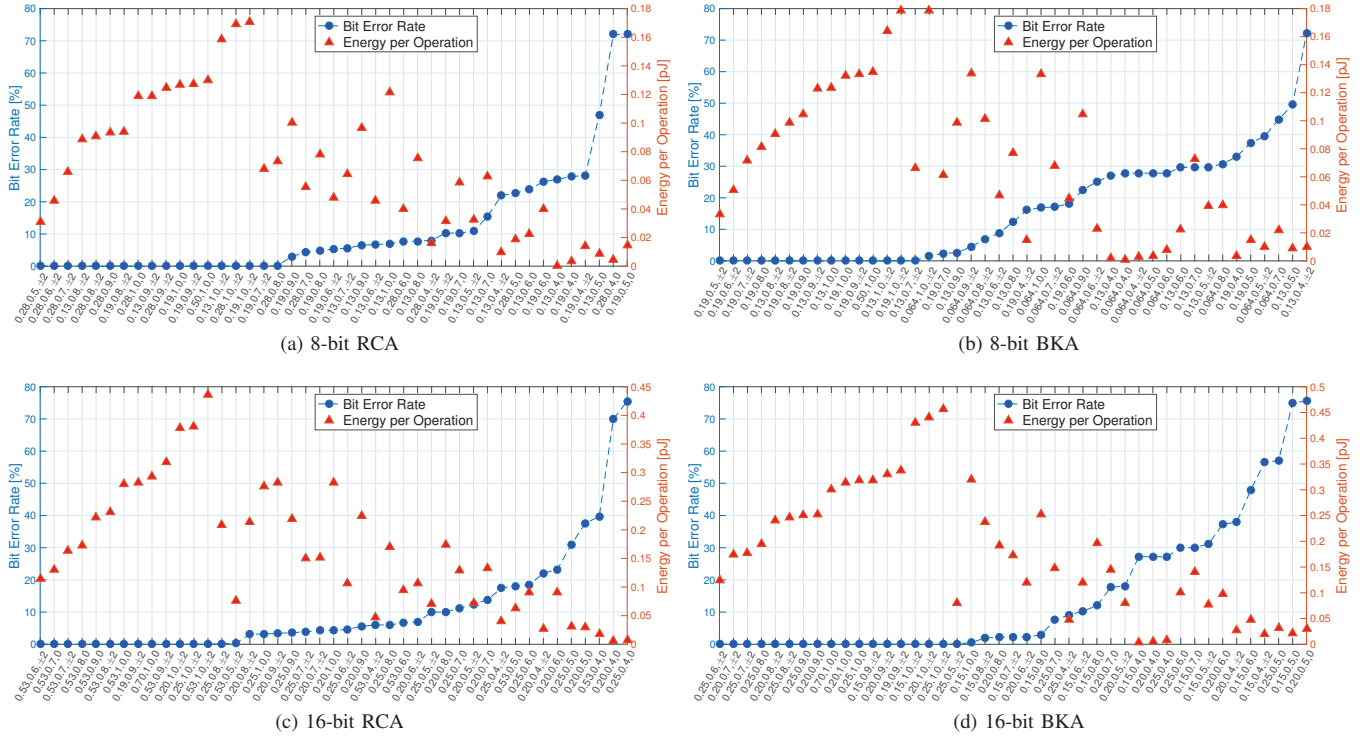


Fig. 8: Bit-Error Rate vs. Energy/Operation for 8-bit and 16-bit adders

[2] E. Le Sueur and G. Heiser, “Dynamic voltage and frequency scaling: The laws of diminishing returns,” in *Proc. of the 2010 intl. conf. on Power aware computing and systems*, pp. 1–8, 2010.

[3] M. Nicolaidis, “Double-sampling design paradigm compendium of architectures,” *IEEE Trans. on Device and Materials Reliability*, vol. 15, no. 1, pp. 10–23, 2015.

[4] R. Hegde and N. R. Shanbhag, “Soft digital signal processing,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 6, pp. 813–823, 2001.

[5] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *18th IEEE European Test Symposium (ETS)*, pp. 1–6, 2013.

[6] G. Tziantzioulis, A. Gok, et al., “Lazy pipelines: Enhancing quality in approximate computing,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1381–1386, 2016.

[7] L. Soares, S. Bampi, et al., “Near-threshold computing for very wide frequency scaling: Approximate adders to rescue performance,” in *IEEE New Circuits and Systems Conference (NEWCAS)*, pp. 1–4, 2015.

[8] Y. Kim, S. Venkataramani, K. Roy, and A. Raghunathan, “Designing approximate circuits using clock overgating,” in *Proc. of the 53rd Annual Design Automation Conference*, p. 15, 2016.

[9] I. Akturk, N. S. Kim, and U. R. Karpuzcu, “Decoupled control and data processing for approximate near-threshold voltage computing,” *IEEE Micro*, vol. 35, no. 4, pp. 70–78, 2015.

[10] C. Li, W. Luo, S. S. Sapatnekar, and J. Hu, “Joint precision optimization and high level synthesis for approximate computing,” in *Proc. of the 52nd Annual Design Automation Conference*, p. 104, 2015.

[11] A. Lingamneni, C. Enz, et al., “Energy parsimonious circuit design through probabilistic pruning,” in *Design, Automation & Test in Europe*, pp. 1–6, 2011.

[12] J. Kim and S. Tiwari, “Inexact computing for ultra low-power nanometer digital circuit design,” in *IEEE/ACM International Symposium on Nanoscale Architectures*, pp. 24–31, 2011.

[13] J. Schlachter, V. Camus, and C. Enz, “Near/sub-threshold circuits and approximate computing: The perfect combination for ultra-low-power systems,” in *IEEE Computer Society Annual Symp. on VLSI*, pp. 476–480, 2015.

[14] A. Lingamneni, C. Enz, K. Palem, and C. Piguet, “Designing energy-efficient arithmetic operators using inexact computing,” *Journal of Low Power Electronics*, vol. 9, no. 1, pp. 141–153, 2013.

[15] W. N. HE et al., *CMOS VLSI Design: A Circuits And Systems Perspective, 4/E*. Pearson Education India, 2006.

[16] A. B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in *Proc. of the 49th Annual Design Automation Conference*, pp. 820–825, 2012.

[17] R. Ragavan, C. Killian, and O. Sentieys, “Adaptive overclocking and error correction based on dynamic speculation window,” in *IEEE Computer Society Annual Symp. on VLSI (ISVLSI)*, pp. 325–330, 2016.