Optimizing Energy to Minimize Errors in Dataflow Graphs Using Approximate Adders

Zvi Kedem
New York University
kedem@nyu.edu

Vincent Mooney
Georgia Institute of Technology & Nanyang Technological University
mooney@gatech.edu, vjmooney@ntu.edu.sg

Kirthi Krishna Muntimadugu
Rice University
kirthi.krishna@rice.edu

Krishna V. Palem
Rice University
palem@rice.edu

Avani Devarasetty
International Institute of Information Technology, Hyderabad
avani@students.iiit.ac.in

Phani Deepak Parasuramuni
International Institute of Information Technology, Hyderabad
phani_dp@students.iiit.ac.in

ABSTRACT

Approximate arithmetic is a promising, new approach to low-energy designs while tackling reliability issues. We present a method to optimally distribute a given energy budget among adders in a dataflow graph so as to minimize expected errors. The method is based on new formal mathematical models and algorithms, which quantitatively characterize the relative importance of the adders in a circuit. We demonstrate this method on a *finite impulse response filter* and a *Fast Fourier Transform*. The optimized energy distribution yields 2.05X lower error in a 16-point FFT and images with SNR 1.42X higher than those achieved by the best previous approach.

Categories and Subject Descriptors

B.2 [**Arithmetic and Logic Structures**]: Miscellaneous; B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms

Design, Reliability

Keywords

Approximate Computation, Voltage Scaling, Energy Consumption Minimization, DSP Circuits

1. INTRODUCTION

The world of computing now faces two important challenges: reliability and energy consumption. First, the miniaturization of computing devices through technology scaling referred to as Moore's Law is a hindrance to reliable computing. Second, portability is hobbled by the energy consumption of mobile electronics. In [7, 4] it is shown that in the context of multimedia audio and video signal processing, both of the challenges can be met: error can be tolerated while energy is saved. This is possible because the quality of the output is evaluated primarily by human perception which can interpret useful information from (slightly) erroneous data. This leads to a new design methodology in which the computations are not deterministic but probabilistic and approximate.

Approximately correct arithmetic, which we address in this paper, was introduced in [4]. In conventional design methodology the supply voltage of a circuit is determined by the frequency of operation. In approximate arithmetic circuits, the supply voltages are lowered below the threshold determined by the frequency of operation, thereby lowering the energy consumption. As a result, the circuit is clocked at a cycle time shorter than its worst-case critical path delay. Therefore, some computations might only be partially completed which results in an "approximate" value at the outputs of the circuit.

To improve the accuracy of the output of the circuit for the same energy consumption, as opposed to uniform voltage scaling, a novel *biased* voltage scaling approach or BIVOS was proposed in [4], where *more important* data is computed more accurately and accuracy is less for *less important* data. For a single adder, this is done by having a higher supply voltage for the most significant bits and a lower supply voltage for the less significant bits.

First, though energy savings were obtained in [7, 4, 12] by biased investment at the level of an adder, there was no definitive methodology to *optimally* supply the voltage across the circuit. Second, there was no previous attempt to optimize a circuit that consisted of multiple components where each of these components could be an n-bit adder. In such circuits, analogous to the case of the computed bits in an adder [7, 4], the relative importance of these components has to be taken into account in order to optimize energy consumption. For example, it is the case that in some circuits,

CASES'10, October 24–29, 2010, Scottsdale, Arizona, USA.

modelled in this paper, the data produced by a particular adder is more important than other adders.

In this paper, we address the latter problem of optimizing a circuit with multiple adders by showing a very efficient method to relatively invest energy across different adders in a circuit based on their *importance*. For supplying voltages for components inside an adder, we use the previous BIVOS approach. Our method is general enough to model any adder design or combination of different adder designs, though we present our specific results for the special case of the ripple carry adder. Our primary contributions in this paper are as follows:

- We provide a strong mathematical foundation capable of modelling the propagation of errors in a circuit with multiple shifters and approximate adders.
- We present a very fast algorithm to *quantitatively* compute the relative importance of each adder in any graph as defined in the model.
- We present a theorem that *optimally* distributes energy based on the relative importance of each adder, applicable to any directed acyclic graph structure.
- We demonstrate this approach on two example circuits, a *finite impulse response filter* (FIR) and a *Fast Fourier Transform* (FFT), and through HSPICE simulations we show that dramatic savings in energy consumption can be achieved when using our approach even when compared to the best existing prior art, BIVOS.

Keeping in mind the domain of DSP, we developed our approach to encompass popularly used circuits such as an FIR or an FFT which can be implemented using only adders and constant-number multipliers. Also, the standard implementation of a constant-number multiplier uses a set of adders and implicit shifting. Hence, in this paper we will consider circuits which consist only of adders and shifters. We only consider optimization of energy and errors in dataflow graphs and do not model memory and feedback elements.

In this section we have motivated our approach and have discussed related techniques. In Section 2, we present our target circuit model and state the associated optimization problem. In Section 3, we develop our solution to minimizing error for a given energy budget. We summarize the method and describe the extension to other adders in Section 4. In Sections 5 and 6, we show the impact of the solution on two applications of interest, an FIR and an FFT. Section 7 discusses the impact our method has on a conventional circuit design framework. We outline future work and present conclusions in Section 8.

1.1 Related work

The fundamentally novel design methodology where the computations are not deterministic but approximate and probabilistic in nature was introduced by Chakrapani et al. [4] and George et al. [7] . In this paper we model approximate arithmetic circuits where accuracy is compromised because of overclocking, which is operating the circuit at a frequency higher than that strictly permitted by the critical path. In contrast, probabilistic arithmetic circuits take into account the inherent thermal noise that is present in all devices as well as any parameter variations. Though thermal noise and parameter variations in current transistor technologies are not very prominent, it is predicted by the International Technology Roadmap for Semiconductors (ITRS) [8] that in future technologies their effect will be drastic. The roadmap also forecasts that relaxing the accuracy constraint on circuits will be necessary to improve the efficiency of manufacturing, verification and testing of circuits.

In contrast to probabilistic or approximate circuits, there are other techniques which use multiple voltages and aggressive voltage scaling. Martin et al. [11] use aggressive voltage scaling, multiple voltage levels, and an adaptive circuit which adjusts its throughput but guarantees that errors do not occur even in the computation. Manzak and Chaktrabarti [10] and Yeh et al. [15] present techniques that are non-adaptive which operate the critical paths of the circuit at higher voltages than the non-critical paths and also use transistor sizing. This is similar to a biased voltage scaling but the bias is because of the time criticality of the output rather than the importance of the data that we use. Ernst et al. [6] present a method in which they use circuit-level timing speculation thus allowing incorrect operation of circuit elements, which are detected and then corrected. A recent announcement in Technology Review (Published by MIT) [3] describes recent advances in error resilient circuit including a prototype chip designed by Tschanz et al. [14] at Intel that lets errors happen and then corrects them using less power overall. Shim et al. [13] show a design in which circuit level timing errors are not corrected at the circuit level, rather techniques borrowed from signal processing are used to correct such errors. But the distinguishing feature of these techniques from our approach is that these circuits might compute incorrectly but then they employ a variety of error correction mechanisms which assure that the output is always correct.

An exception to these techniques is by Banerjee et al. [1] where in the specific case of a 2-dimensional discrete cosine transform they modify the circuit topology such that computations that are more important to output quality take shorter time than the ones that do not affect the output quality as much. So when they overclock, the computations that take shorter time would be computed correctly but the other ones might have errors in them.

A straightforward technique to reduce the energy consumption is power gating [9], which is simply cutting off power to *less important* circuitry. But in power gating we are neglecting the switched part of information completely, whereas in our approach we can relatively invest based on the significance of the data.

2. THE MODEL

In this section we define the model we use and specify the problem we address. The concept of trading error for energy savings in circuits is entirely at an early stage and thus, we felt a need for a foundational model and formal mathematical results, leaving detailed simulation of large systems for later.

As discussed, we will consider circuits of adders only. Some of these circuits have been obtained by reducing constant-number multipliers to a set of adders and shifters. This is advantageous because in many circuits used in DSP, values are multiplied by constants and hence general multipliers are not needed. They can be replaced with a set of adders and shifters using a variety of methods such as those used in [2], thereby reducing total area, energy consumed, and delay. In the design of such a circuit, shifting of a number is

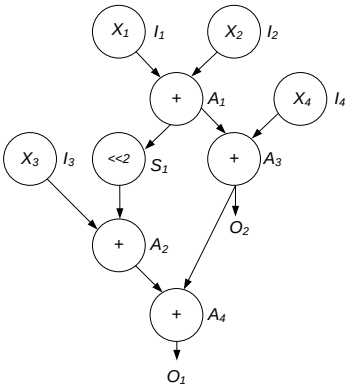


Figure 1: Example of a graph-theoretical representation of a circuit

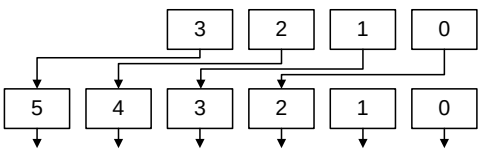


Figure 2: An example of an implicit shifter

done implicitly by routing the interconnects to shift the bits appropriately to another position.

But in modeling we will explicitly consider and show shifters as they influence the size of the errors. The implicit implementation of the shifter S_1 shown in Fig. 1 which shifts a number to the left by 2 positions is shown in Fig. 2, where a rectangle represents a full adder and the number inside the rectangle denotes its position in the adder.

2.1 The graph based model

In the context of modeling, it is useful to consider shifters explicitly though they are not actually implemented. A circuit we target consists of the following components: inputs, shifters and adders. Some adders or shifters are also labeled as outputs. It is convenient to model such a circuit using a directed acyclic graph (DAG).

The DAG will have some $N_{\rm I} + N_{\rm A} + N_{\rm S}$ vertices: *inputs* I_1, I_2, ..., and $I_{N_{\rm I}}$; *adders* A_1, A_2, ..., and $A_{N_{\rm A}}$; and *shifters* S_1, S_2, ..., and $S_{N_{\rm S}}$. Some of the adders or shifters are also labeled as *outputs* O_1, O_2, ..., $O_{N_{\rm O}}$. Thus each A_i or S_j is either not an output or a unique O_k, so O_k is just an alias for A_i or S_j. Each O_k may have any number of bits but this will typically be a power of two. We will write n for N_I. For a simple example, see Fig. 1.

A vertex I_i has in-degree 0, and an input x_i to the circuit is supplied at I_i. An input x_i may have any number of bits but will typically be a power of two. A vertex A_j has in-degree 2 and is an adder, adding the two numbers on its incoming arcs. A vertex S_j has in-degree 1 and shifts the number on its incoming arc either left or right by the specified amount. We will use the term s-shifter for a shifter vertex that shifts the input left with magnitude s (where s is a positive or negative integer).

At each adder or shifter, a linear function of $x_1, x_2, \ldots, x_n$ is computed. Thus the value of the output at O_k (corresponding to some adder or shifter) is a function $F_k(\mathbf{x}) = \sum_{i=1}^{n} w_{k,i}x_i$ for some $w_{k,1}, w_{k,2}, \ldots, w_{k,n}$, which can also be written as $\mathbf{w}_k \cdot \mathbf{x}^{\rm T}$ where $\mathbf{w}_k = (w_{k,1}, w_{k,2}, \ldots, w_{k,n})$, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and "$\cdot$" is the scalar product. $\mathbf{x}$ is drawn from a set $\mathbf{X}$ of allowable inputs. For example, each x_i could be an integer in the range $[0, 2^{10})$.

For Fig. 1, we see that $F_1(\mathbf{x}) = (5, 5, 1, 1) \cdot \mathbf{x}^{\rm T}$.

2.2 The energy optimization problem for our target dataflow graph of adders

Let E be the *total energy budget* to be invested in the circuit, and let $E_{\rm Add}$ be the *energy required to run an adder correctly* for all possible inputs. We assume that $E < N_A E_{\rm Add}$, and therefore we cannot assure that all adders run correctly for all inputs, and for different inputs $\mathbf{x}$, different error values may appear. Let E_j be the energy actually supplied to adder A_j. Given some choice of E_1, E_2, ..., and $E_{N_{\rm A}}$, such that $\sum_{j=1}^{N_{\rm A}} E_j = E$, and some input $\mathbf{x}$, the resulting expected error for $F_k(\mathbf{x})$ is denoted by $\mathrm{Er}(O_k, \mathbf{x})$. We use the term expected error here to consider the effect of temperature and process variations in the parameters of the circuit, but we will refer to it just as error in the later sections. We rely on [4] for calculation of expected error of a single adder. Our task in this paper is to distribute E among the adders so as to

$$\text{minimize} \underset{\mathbf{x}\in\mathbf{X}}{\text{avg}} \sum_{k=1}^{N_O} \mathrm{Er}(O_k, \mathbf{x}). \quad (1)$$

We refer to this problem as the *single resource dataflow energy-error optimization* problem, and our methodology to solve this problem is presented in Section 3.

3. THE SINGLE RESOURCE DATAFLOW ENERGY-ERROR OPTIMIZATION METHOD

To model the interactions and effect of multiple adders producing errors in the outputs of a circuit, we first consider just a single adder producing error in the circuit to define some metrics that we use later in the optimization.

3.1 A single approximate adder

We consider the implications of a single adder that produces an error (for at least some inputs). We will refer to such an adder as an *approximate adder*. We will see that the same errors in different adders may have different implications for errors at an output. Consider some input $\mathbf{x}$ and assume that only one adder, $A_{\rm Er}$, *produces* an "approximate" result since the inputs were supplied. This error propagates, and can cause an error of at an output O_k for $k \in \{1, 2, \ldots, N_{\rm O}\}$. We look at the value of the outputs after time $t_{\rm O}$ units since the inputs were supplied. Let $t(A_{\rm Er}, k)$ be the time it takes for the output of $A_{\rm Er}$ to propagate to O_k. Let $\mathrm{Er}(A_{\rm Er}, \mathbf{x}, t)$ and $\mathrm{Er}(O_k, \mathbf{x}, t)$ be the errors at the output of $A_{\rm Er}$ and output O_k after t units of time since the inputs were supplied with $\mathrm{Er}(A_{\rm Er}, \mathbf{x}, t) \neq 0$. Then we define the *significance* of $A_{\rm Er}$, $\sigma(A_{\rm Er}, \mathbf{x}, t_{\rm O}, t(A_{\rm Er}, k))$ as follows:

$$\sigma(A_{\rm Er}, \mathbf{x}, t_O, t(A_{\rm Er}, k)) = \frac{\sum_{k=1}^{N_{\rm O}} \mathrm{Er}(O_k, \mathbf{x}, t_{\rm O})}{\mathrm{Er}(A_{\rm Er}, \mathbf{x}, t_{\rm O} - t(A_{\rm Er}, k))}. \quad (2)$$

Let us refer to Fig. 1 again. Let the adders A_1, A_2, A_3 and A_4 be 8-bit ripple carry adders (RCAs) and

the inputs be $x_1 = 15$, $x_2 = 1$, $x_3 = 0$, and $x_4 = 0$. Let the worst case delay of a full adder be 10 units of time, and therefore an 8-bit RCA has a critical path delay of 80 units. With this specific input, A_1 would take 50 units to compute the correct output in the worst case because the carry has to propagate across 5 full adders. For this output to propagate till O_1, it would take about 70 units to propagate through the other adders as Fig. 1 describes a combinational circuit. But if we overclock the circuit such that we sample O_1 after 50 units, the answer would only be *approximate*.

To compute the error at the output after 50 units of time, consider A_1. After 30 units, the output of adding $00001111 = 15$ and $00000001 = 1$, assuming exact worst-case delays, is $00000100 = 8$. The output value of A_1 propagates to the output of A_2 by 40 units and it has a value of $8 \times 2^2 = 32$ when it reaches the left input to A_4. Also the output value of A_1 propagates to the output of A_3 by 40 units and has a value of 8 when it reaches on the other input of A_4. Thus, the value at the output of O_1 would be $32 + 8 = 40$ after 50 units of time whereas the correct output is $16 \times 2^2 + 16 = 80$. As a result, there is an error of 40 and the significance of A_1 to O_1 is equal to $40/8 = 5$ (where 8 is the error of the approximate adder after 30 units). This example shows that though the first adder had enough time to compute the correct value (by the end of 50 units of time A_1 would have computed correctly), this value did not have enough time to propagate through the rest of the circuit and thus to impact the final output.

Note that the exact magnitude of error is dependent on the time at which the outputs are sampled and the propagation time from the approximate adder to the particular output. But the quantity that we are interested is the significance of the approximate adder which is the amount by which the error at the approximate adder is amplified (or reduced) when it has propagated to the output. As is evident from the above example, this significance value is dependent only on the circuit topology and not the exact magnitude of errors. We strengthen this concept in Section 3.2 where we evaluate the significance of an adder based solely on the circuit topology independent of the individual errors. So we will omit the time parameter in the following discussions.

3.2 Computing the significance of an adder

Consider some circuit C, some energy budget E, and some input $\mathbf{x}$, such that there is exactly one approximate adder (all other adders compute correctly), and denote that adder by A_{Er}. This adder may cause errors in various vertices, and for vertex v and input $\mathbf{x}$, the error will be denoted by $\mathrm{Er}(v, \mathbf{x})$. Then, the *significance* of A_{Er} to v under $\mathbf{x}$ is defined by

$$\sigma(A_{\mathrm{Er}}, v, \mathbf{x}) = \frac{\mathrm{Er}(v, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})}. \tag{3}$$

Of course, if there is no path from A_{Er} to v, then $\sigma(A_{\mathrm{Er}}, v, \mathbf{x}) = 0$. We will now prove some useful relations. First we note, that,

$$\sigma(A_{\mathrm{Er}}, A_{\mathrm{Er}}, \mathbf{x}) = 1. \tag{4}$$

Let $v \neq A_{\mathrm{Er}}$. No errors can propagate to a circuit's inputs, so we will consider only shifters and adders. Assume that v is an s-shifter with an immediate predecessor u. Then as a shifter does not introduce errors but may *amplify* (or reduce) them, $\mathrm{Er}(v, \mathbf{x}) = 2^s \mathrm{Er}(u, \mathbf{x})$, and therefore

$$\begin{aligned}\sigma(A_{\mathrm{Er}}, v, \mathbf{x}) = \frac{\mathrm{Er}(v, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} &= 2^s \frac{\mathrm{Er}(u, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} \\ &= 2^s \sigma(A_{\mathrm{Er}}, u, \mathbf{x}).\end{aligned} \tag{5}$$

and if $\sigma(A_{\mathrm{Er}}, u, \mathbf{x})$ does not depend on $\mathbf{x}$, neither does $\sigma(A_{\mathrm{Er}}, v, \mathbf{x})$.

Assume that v is an adder with immediate predecessors u and w. Adder v is not A_{Er} and therefore does not introduce errors. Then,

$$\begin{aligned}\sigma(A_{\mathrm{Er}}, v, \mathbf{x}) = \frac{\mathrm{Er}(v, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} &= \frac{\mathrm{Er}(u, \mathbf{x}) + \mathrm{Er}(w, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} \\ &= \frac{\mathrm{Er}(u, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} + \frac{\mathrm{Er}(w, \mathbf{x})}{\mathrm{Er}(A_{\mathrm{Er}}, \mathbf{x})} \\ &= \sigma(A_{\mathrm{Er}}, u, \mathbf{x}) + \sigma(A_{\mathrm{Er}}, w, \mathbf{x}).\end{aligned} \tag{6}$$

and if $\sigma(A_{\mathrm{Er}}, u, \mathbf{x})$ and $\sigma(A_{\mathrm{Er}}, w, \mathbf{x})$ do not depend on $\mathbf{x}$, neither does $\sigma(A_{\mathrm{Er}}, v, \mathbf{x})$.

From Eqs. 4–6, by simple inductive argument, it follows also that the significance of a vertex is always greater or equal to 0 and *it does not depend on the value of the input* $\mathbf{x}$ *that caused the error at* A_{Er}! It is purely a property of the circuit's stucture. Therefore we can write just $\sigma(A_{\mathrm{Er}}, v)$ for the significance of A_{Er} to v no matter what $\mathbf{x}$ is (though sometimes it may be convenient to write it explicitly).

Similarly, it is easy to see, that the $\sigma(A_{\mathrm{Er}}, \mathbf{x})$ does not depend on $\mathbf{x}$, so we can just write $\sigma(A_{\mathrm{Er}})$.

By referring to Fig. 1, we can provide intuition for significance and its properties. Assume that for some energy budget E_1 and input $\mathbf{x}_1$, A_1 produced an error of δ_1, and adders A_2, A_3, and A_4 processed their summands correctly. Then A_2 produced an error of $4\delta_1$, A_3 of δ_1, and A_4 of $5\delta_1$. So, e.g., $\sigma(A_1, A_4, \mathbf{x}_1) = 5\delta_1/\delta_1 = 5$. But similarly, if for E_2 and input $\mathbf{x}_2$, A_1 produced an error of δ_2, and adders A_2, A_3, and A_4 processed their summands correctly, still $\sigma(A_1, A_4, \mathbf{x}_2) = 5\delta_2/\delta_2 = 5$.

We can also discuss the relative importance of the correctness of adders. If for some energy budget A_3 produced an error of δ, and adders A_1, A_2, and A_4 processed their summands correctly, $\sigma(A_3, A_4, \mathbf{x}) = \delta/\delta = 1$, so for A_4 correctness of A_1 is more important than that of A_3.

For each v we define an *amplification factor*, $\mathrm{AF}(v)$, to help produce a simple algorithm for computing significances, as

$$\mathrm{AF}(v) = \begin{cases} 1, & \text{if } v \text{ is an input or an adder} \\ 2^s, & \text{if it is an } s\text{-shifter.} \end{cases} \tag{7}$$

We extend the definition to paths of vertices, by

$$\mathrm{AF}(v_{j_1}, v_{j_2}, \ldots, v_{j_k}) = \prod_{i=1}^{k} \mathrm{AF}(v_{j_i}). \tag{8}$$

For vertices u and v, we will denote by $P(u, v)$ the set of all the paths from u to v.

We next give a very easily computable, explicit formula for computing $\sigma(A_{\mathrm{Er}}, v)$.

LEMMA 3.1. *Assume that under some energy budget E and input $\mathbf{x}$ a circuit has exactly one approximate adder, A_{Er}. Let v be any vertex. Then*

$$\sigma(A_{\mathrm{Er}}, v) = \sum_{p \in P(A_{\mathrm{Er}}, v)} AF(p).$$

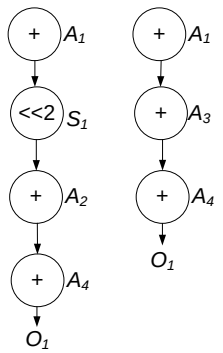


Figure 3: The two paths from A_1 to A_4 ($= O_1$). Error of δ at A_1, while propagating through the left path contributes an error of 4δ to error at O_1 and an error of 1δ while propagating through the right path. Thus the total error at O_1 is 5δ and the significance of $A_1 = 5\delta/\delta$.

EXAMPLE. Before starting the formal proof, we look at an example. Let us consider the graph in Fig. 1 with $A_{\mathrm{Er}} = A_1$ and $v = O_1$. There are two paths from A_1 to O_1, the "left" path $p_{\mathrm{L}} = (A_1, S_1, A_2, O_1)$ and the "right" path $p_{\mathrm{R}} =(A_1, A_3, O_1)$ as shown in Fig. 3.

Assume that an error of δ is generated by A_1. Consider p_{R} first. On this path δ is passed through A_3 and the incoming error at O_1 is δ. Consider p_{L} now. On this path δ is converted to $2^2\delta$ by S_1, the error of $2^2\delta$ is passed through A_2 and the incoming error at O_1 is $2^2\delta$. The total error at O_1 is $(2^2 + 1)\delta = 5\delta$.

By the definition in Eq. 8, we have $\mathrm{AF}(p_{\mathrm{L}}) = 1{\cdot}2^2{\cdot}1{\cdot}1 = 2^2$ and $\mathrm{AF}(p_{\mathrm{R}}) = 1{\cdot}1{\cdot}1 = 1$ and therefore $\sum_{p\in P(A_1,O_1)} \mathrm{AF}(p) = 4 + 1 = 5$.

But by Eq. 3 (recall that $\mathbf{x}$ can be omitted), $\sigma(A_1, O_1) = \frac{5\delta}{\delta} = 5$. Therefore the Lemma holds for this example.

PROOF. To shorten the proof, we skip over simple "pathological" cases, such as the case of one vertex connected by two outgoing arcs to a single adder.

As already noted, if there is no path from A_{Er} to v, i.e., $P(A_{\mathrm{Er}}, v) = \emptyset$ then $\sigma(A_{\mathrm{Er}}, v) = 0$. Therefore the claim holds in such cases.

We prove the lemma by induction on N, the number of vertices in the circuit.

The smallest circuit of interest has $N = 3$ vertices, two inputs feeding one adder. So this will be our base case. Here $v = A_{\mathrm{Er}}$ and $P(A_{\mathrm{Er}}, A_{\mathrm{Er}})$ consist of only one path of length 1, namely (A_{Er}). From Eq. 4, $\sigma(A_{\mathrm{Er}}, A_{\mathrm{Er}}) = 1$ and as by Eq. 7, $\mathrm{AF}(A_{\mathrm{Er}}) = 1$, the claim holds.

Let now $N > 3$ and assume that the lemma holds for all circuits with at most $N - 1$ vertices. Consider any circuit C of N vertices and remove from it any vertex v of out-degree 0 together the arcs incoming to it, resulting in a new circuit D. We will also use C and D as subscripts to indicate to which of the two circuits we are referring.

Note that, in general, if A_{Er} has out-degree 0, then again any v of interest is just A_{Er} itself, and similarly to the base case, $\sigma(A_{\mathrm{Er}}, A_{\mathrm{Er}}) = \mathrm{AF}(A_{\mathrm{Er}}) = 1$ and the claim holds. Therefore, if the removed vertex was A_{Er}, we already know that the claim holds for C, so consider the case where A_{Er} was not the removed vertex, and therefore it is also in D.

Table 1: Significance values of the adders in the graph shown in Fig. 1

A_{Er}	$\sigma(A_{\mathrm{Er}}, O_1)$	$\sigma(A_{\mathrm{Er}}, O_2)$	$\sigma(A_{\mathrm{Er}})$
A_1	5	1	6
A_2	1	0	1
A_3	1	1	2
A_4	1	0	1

There are two cases for v. If v is an s-shifter, it has one predecessor, say u. There is a one-to-one correspondence between paths in $P_D(A_{\mathrm{Er}}, u)$ and those in $P_c(A_{\mathrm{Er}}, v)$. Every path p in the latter set is obtained by extending exactly one path q in the former set by v. By Eqs. 7–8, $\mathrm{AF}(p) = 2^s\mathrm{AF}(q)$ and therefore $\sum_{p\in P_C(A_{\mathrm{Er}},v)} \mathrm{AF}(p) = 2^s \sum_{q\in P_D(A_{\mathrm{Er}},u)} \mathrm{AF}(q)$. From Eq. 5, $\sigma_C(A_{\mathrm{Er}}, v) = 2^s\sigma_D(A_{\mathrm{Er}}, u)$, and the claim follows.

If v is an adder, it has two predecessors, say u and w. Note that as a path cannot end both with u and w, $P_D(A_{\mathrm{Er}}, u) \cap P_D(A_{\mathrm{Er}}, w) = \emptyset$. There is a one-to one correspondence between paths in $P_D(A_{\mathrm{Er}}, u) \cup P_D(A_{\mathrm{Er}}, w)$ and those in $P_C(A_{\mathrm{Er}}, v)$. Every path p in the latter set is obtained by extending exactly one path q in the former set by v. By Eqs. 7–8, $\mathrm{AF}(p) = \mathrm{AF}(q)$, and therefore $\sum_{p\in P_C(A_{\mathrm{Er}},v)} \mathrm{AF}(p) = \sum_{q\in P_D(A_{\mathrm{Er}},u)} \mathrm{AF}(q) + \sum_{q\in P_D(A_{\mathrm{Er}},w)} \mathrm{AF}(q)$. From Eq. 6, $\sigma_C(A_{\mathrm{Er}}, v) = \sigma_D(A_{\mathrm{Er}}, u) + \sigma_D(A_{\mathrm{Er}}, w)$, and the claim follows.

By induction, for any vertex z in D the claim holds. Since v was of out-degree 0, no path from A_{Er} to z in C can pass through v, v cannot "impact" any other vertex in C, and therefore, $\mathrm{Er}_C(z, \mathbf{x}) = \mathrm{Er}_D(z, \mathbf{x})$ As, of course $\mathrm{Er}_C(A_{\mathrm{Er}}, \mathbf{x}) = \mathrm{Er}_D(A_{\mathrm{Er}}, \mathbf{x})$, it follows that $\sigma_C(z) = \sigma_D(z)$. By induction $\sigma_D(A_{\mathrm{Er}}, z) = \sum_{p\in P_D(A_{\mathrm{Er}},z)} \mathrm{AF}(p)$, and as $P_C(A_{\mathrm{Er}}, z) = P_D(A_{\mathrm{Er}}, z)$ the claim holds for all such vertices z. □

THEOREM 1. *Assume that under some energy budget E and input $\mathbf{x}$ a circuit has exactly one approximate adder, A_{Er}. Then,*

$$\sigma(A_{\mathrm{Er}}) = \sum_{k=1}^{N_O} \sum_{p\in P(A_{\mathrm{Er}},O_k)} \mathrm{AF}(p). \tag{9}$$

PROOF. Immediate from Eq. 2 and Lemma 3.1. □

To explicitly demonstrate the application of Theorem 1 to compute the significance (relative importance) of each adder we use the graph in Fig. 1. The relative significance values from Eq. 9 for each output from each vertex is shown in Table 1.

The above method to compute the significance of an adder, $\sigma(A_{\mathrm{Er}})$, by Eq. 9, is very fast. A simple method of computing $\mathrm{AF}(p)$ is to do a breadth-first search [5] from each vertex and count all paths from the vertex to the outputs. This would be a $O\big((V + E)V\big)$ operation where V is the number of vertices and E is the number of arcs in the graph.

COROLLARY 1. *Assume that under some energy budget E and input $\mathbf{x}$ a circuit that contains no shifters (explicit or implicit) has exactly one approximate adder, A_{Er}. Then,*

using | | to denote cardinality,

$$\sigma(A_{\mathrm{Er}}) = \sum_{k=1}^{N_O} |P(A_{\mathrm{Er}}, O_k)|.$$

PROOF. If there are no shifters, $\mathrm{AF}(p) = 1$ for any path p. □

3.3 Multiple approximate adders

Until now, we have considered only the case when one adder produces an error while adding its summands. In general, a set of adders can produce errors. The cumulative effect of this set of adders produces an error in an output whose absolute value is between 0 and the sum of the absolute values of the individual errors, as they may partially (or fully) cancel each other out.

Modeling this phenomenon of errors canceling out is complex and also partially depends on the particular input. Hence to simplify the analysis we will target to minimize the *the worst possible case*, in which the errors do not cancel each other to any extent. Thus, we want to minimize the sum of the absolute values of the errors.

3.4 Case study: Ripple carry adder

The discussion in Sections 2 and 3 holds for any type of adder. To proceed, however we will need to choose specific designs of adders, whose error production under various energy investment has been studied. This leads to considering the Ripple Carry Adder (RCA) for which such results exist.

It has been shown in [4], through simulations over a large number of input cases, that the average expected error $\mathrm{Er}(A_i)$ for a RCA A_i with BIVOS, is roughly proportional to its delay D, i.e. $\mathrm{Er}(A_i) \propto D$. In a conventional CMOS transistor, the delay of a transistor D_{T} is inversely proportional to its supply voltage VDD, i.e. $D_{\mathrm{T}} \propto \mathrm{VDD}^{-1}$.

The dynamic energy consumed during a transition of a transistor switch E_{T} is proportional to the square of its supply voltage, i.e., $E_{\mathrm{T}} \propto \mathrm{VDD}^2$. Also, the delay of adder A_i, D, is proportional to the delay of a transistor D_{T}, and the dynamic energy E_i consumed by adder A_i is proportional to the energy of a transistor E_{T}. Thus $E_i \propto E_{\mathrm{T}} \propto \mathrm{VDD}^2 \propto {D_{\mathrm{T}}}^{-2} \propto D^{-2} \propto \mathrm{Er}(A_i)^{-2}$ and for some constant r, we can write

$$E_A \cong r \frac{1}{\mathrm{Er}^2(A_i)}. \tag{10}$$

3.5 Optimizing energy distribution

We will make use of a solution to an optimization problem which we present next.

LEMMA 3.2. *Let integer $n > 0$ and $c, a_1, a_2, \ldots, a_n > 0$. Then, the function $\sum_{j=1}^{n} a_j x_j$ subject to constraints $x_j > 0$ for $j = 1, \ldots, n$ and $\sum_{j=1}^{n} x_j^{-2} = c$ is minimized at*

$$x_i = \frac{\left(\sum_{j=1}^{n} a_j^{2/3}\right)^{1/2}}{c^{1/2}\, a_i^{1/3}} \quad \textit{for } i = 1, \ldots, n$$

PROOF. By using Lagrange multipliers. □

It will actually be more useful to write the solution as

$$x_i^2 = \frac{\sum_{j=1}^{n} a_j^{2/3}}{c a_i^{2/3}} \quad \text{for } i = 1, \ldots, n \tag{11}$$

We now have

THEOREM 2. *Given a circuit with significance σ_i computed for each adder A_i, the optimal distribution of a given energy budget E to minimize the average sum of worst case errors is given by*

$$E_i = E \frac{\sigma_i^{2/3}}{\sum_{j=1}^{N_{\mathrm{A}}} \sigma_j^{2/3}} \tag{12}$$

where E_i is the energy devoted to A_i.

PROOF. Let $\mathrm{Er}(A_i, \mathbf{x})$ be the error produced at approximate adder A_i in the given circuit for input $\mathbf{x}$. From Eq. 2 and Theorem 1 the sum of errors at the outputs due to A_i is $\sigma(A_i)\mathrm{Er}(A_i, \mathbf{x})$. Thus, the worst case error that we want to minimize is the average over all $\mathbf{x}$'s of $\sum_{i=1}^{N_{\mathrm{A}}} |\sigma(A_i)\mathrm{Er}(\mathrm{A}_i, \mathbf{x})|$. Since we have a fixed energy budget E, $\sum_{i=1}^{N_{\mathrm{A}}} \mathrm{Er}^{-2}(\mathrm{A}_i, \mathbf{x})$ is constant, see Eq. 10. Applying Eq. 11, we obtain Eq. 12. □

It is to be noted that we only determine through Theorem 2 the energy of each adder in the graph. The method through which the supply voltages for the components inside a single adder are allocated is similar to the method in [4]. We pick a set of voltages and then using simulations of the single adder for different binning schemes (binning [4] is a technique in which each component is assigned a particular voltage bin) we pick the best.

4. SUMMARY OF THE METHOD AND EXTENSION TO OTHER ADDERS

Our method of minimizing energy usage across multiple adders in a circuit depends on the following two key elements.

1. Quantitative characterization of the relative significance of each adder in the circuit

   We model the circuit as a DAG consisting of adders and shifters, and compute the significance of each adder based on the topology of the circuit as formalized in Eq. 9

2. The computation of the best distribution of the given energy budget, based on the relative importance of all the adders, which have been computed above. To do this, we rely on the relationship between the voltage supplied to an adder (and thus the energy devoted to it) and the average error produced by the adder

   This computation is done for the case of ripple carry adder, for which the relationship between the energy and the error is known, and we obtain the energy to be devoted to each adder in Eq. 12

Our method to implement the first element is adder-independent. It does not matter which adder circuit is used, as the relative importance of an adder *depends only on the graph topology.* The second element, though, depends on the way error produced at the output of an adder is affected with respect to the energy invested on that adder, which we did for ripple carry adders.

For adders with different designs (such as a carry lookahead adder or a carry skip adder) the error-energy relationship might be different. In this case, the result that will be different is of Lemma 3.2 (which will influence Theorem 2) where the constraint varies based on the error-energy relationship. For example, if for some adder design the energy of the adder was proportional to the cube of the inverse of the error

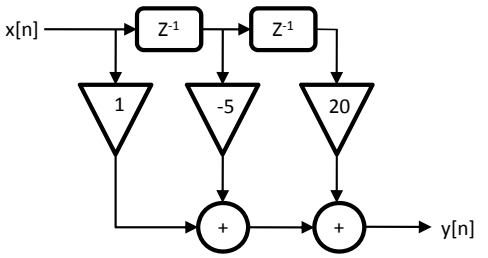


Figure 4: A finite impulse response filter

(instead of the inverse of a square in the case of dynamic energy usage of a ripple carry adder), then the constraint would be $\sum_{i=1}^{n} \frac{1}{x_i^3} - c$. In a general case, let us consider that the error-energy relationship is $E_A \cong f(\mathrm{Er}(A_i))$, then the constraint would be $\sum_{i=1}^{n} f(x_i) - c$.

5. OPTIMIZING DESIGNS OF DSP PRIMITIVES

We apply the *single resource dataflow energy-error optimization* method of Theorem 2 to two specific cases, a *finite impulse response filter* and the *Fast Fourier Transform* which are ubiquitous in embedded signal processing. These signal processing elements are used in a variety of applications such as hearing aids or media players where the output of the devices is evaluated by human perception (which can tolerate errors).

5.1 Converting constant-number multipliers to adders and shifters

As discussed, every constant-number multiplier can be converted to a set of adders and shifters. For example, $x \times 5 = (x << 2) + x$ and $x \times 15 = (x << 4) - x$. Also, if we want both $x \times 21$ and $x \times 13$, then we can compute $x \times 5 = (x << 2) + x$ just once and use it to compute both $x \times 21 = (x << 4) + x \times 5$ and $x \times 13 = (x << 3) + x \times 5$.

To summarize, we try to use canonical signed digit (CSD) coding and sub-expression sharing techniques [2] to transform constant-number multiplications to additions and shifting most efficiently.

5.2 Approximate finite impulse response filter

We will consider digital filters with a finite-duration impulse response (FIR). The output is

$$y[n] = \sum_{m=0}^{N-1} h[m]\, x[n-m].$$

where n and m are integers representing samples in time, x is the input sequence, y the output sequence and h is the impulse response of length N.

Any FIR can be represented in the graph-theoretic framework of Section 2. Consider the FIR shown in Fig. 4 which computes $y[n] = x[n] - 5x[n-1] + 20x[n-2]$. Fig. 5 shows it in the form of a DAG where we use a 16-bit 2's complement RCA for each adder.

According to Theorem 2, the optimal distribution of energy across adders depends on the significance of each adder. As there are no shifters and there is exactly one path from each adder to the output, $\sigma(\mathrm{A}_i) = 1$ for $1 \le i \le N_\mathrm{A}$. Thus, every

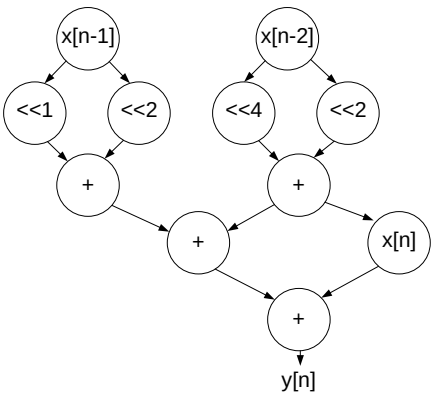


Figure 5: Graph theoretical representation of a finite impulse response filter

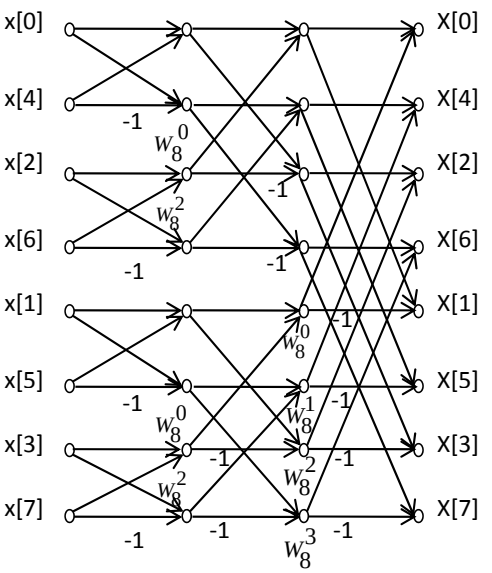


Figure 6: Flow graph of a complete decimation-in-time decomposition of a 8-point FFT

adder in this circuit is "equally important" and distribution of energy investment equally across all adders produces the minimum error magnitude in the output. This holds for all FIRs.

5.3 Approximate fast Fourier transform

For a finite duration sequence of complex numbers, the *Discrete Fourier Transform* (DFT) is used to transform a sequence from its original representation (often in the time domain) to the frequency domain representation. It is

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n] W_N^{kn} & \text{if } 0 \le k \le N-1 \\ 0 & \text{otherwise.} \end{cases}$$

where x is the input sequence, X is the DFT and $W_N = \mathrm{e}^{\frac{-2\pi i}{N}}$.

We use the *Fast Fourier Transform* (FFT) to compute the DFT. A flow graph of a complete decimation-in-time decom-

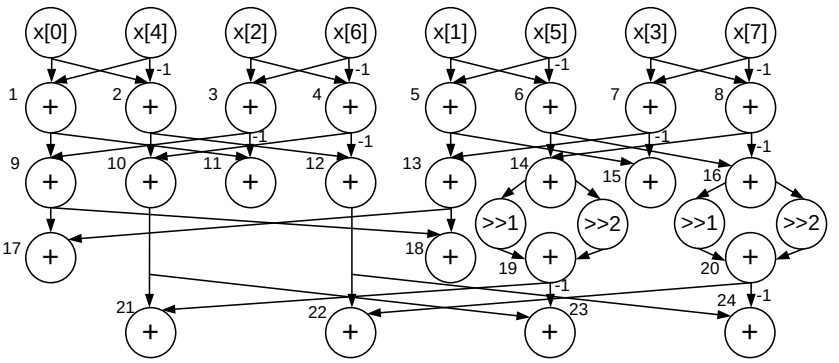


Figure 7: Graph-theoretic representation of an 8-point FFT

position of an 8-point FFT computation is shown in Fig. 6, with its graph-theoretic description shown in Fig. 7.

To distribute the energy budget, we use Eq. 12, after computing σ_i's for all adders A_i's. For this, we use Eq. 9, where each A_i in turn plays the role of A_{Er}. From Fig. 7, $N_{\mathrm{A}} = 24$. We do not show all the significance values for all the adders, but for example, $\sigma_1 = 2$ and $\sigma_6 = 3$. Given all σ_i's, we apply Eq. 12.

This method is easily applicable to higher order FFTs and we applied it to a 16-point FFT, whose diagram we do not show, as it is quite large. We use the 16-point FFT design presented in [16], which is a hardware-efficient architecture based on the phase-amplitude splitting technique which converts a DFT to cyclic convolutions and additions. In the design, all the multipliers are converted into a set of shifters and adders.

We convert the original systolic implementation into a parallel implementation by replicating the circuit after removing memory elements and feedback loops. This results in a circuit consisting of 207 adders, which processes all 16 data words simultaneously. We used 16-bit 2's complement adders to design both the 8-point and 16-point FFTs. We show the results of applying the energy optimization technique on these circuits in the next section.

6. EXPERIMENTAL FRAMEWORK AND RESULTS

In this section we present the framework for validation of this global optimization scheme. We will also present the simulation results which show the savings in energy consumption that can be obtained by applying this methodology.

6.1 Simulation framework

To validate our claims we use Synopsys HSPICE Version B-2008.09 and explore across different supply voltage configurations. The number of configurations is too large for all of them to be explored in HSPICE. So we developed a very fast, C++ based simulator framework for approximate circuits, which we use as first step in narrowing down the number of candidate configurations of interest. We feed this simulator the energy consumption and transition delay values of basic gates simulated in HSPICE. The simulator uses this data to simulate the behavior of approximate circuits. From this simulation we obtain the average error at the output and the average dynamic energy consumption for the circuit over a large set of input data.

Once candidate configurations are obtained by this simulator, we simulate the entire circuit with these configurations in HSPICE. All the simulations are performed in Synopsys 90 nm technology. The technology chosen has approximately 0.1% static leakage, so we have not yet needed to model static energy consumption. The range of voltages in which the circuit components are operated is 0.7 V to 1.2 V. To limit the overhead of supplying and transmitting these voltages, we picked only four specific ones for all our circuits, namely 0.7 V, 0.9 V, 1.0 V, and 1.2 V. We looked at all possible combinations of supply voltages considering 0.7 V, 0.8 V, 0.9 V, 1.0 V, 1.1 V and 1.2 V, but we picked these four specific voltages to present the results as they seemed to perform the best given that we wanted at most four distinct voltages (and hence four voltage domains/islands).

Although our custom simulator is only used to propose candidates to HSPICE for the 8-point FFT, we validated the simulator with respect to the energy consumption and average error to be within a margin of 12% by complete HSPICE simulations of an 8-point FFT for the four supply voltage levels.

These circuits are built using 16-bit 2's complement ripple carry adders. We picked the 16-bit ripple carry adder for the sake of simplicity of implementation and ease of understanding. The circuit that has been simulated is a combinational circuit with no pipelining and sampling only at the final outputs. Similar to the technique used in [4], we overclock the circuits so that the frequency of operation is higher than that permitted by the critical path of the circuit.

6.2 Results and comparisons

We start by briefly summarizing previous results, which are directly relevant to our work. Let E_{con} denote the the energy consumption of a conventional implementation of a circuit, in which the circuit is being operated at a frequency that is equal to what is permitted by the critical path delay, and hence has no *approximate* answers.

Now we consider circuits which are overclocked, that is, they are operated at frequencies higher than that permitted by conventional implementation (limited by critical path delay). Therefore, these circuits are approximate in their computation, as not enough energy (energy less than E_{con})

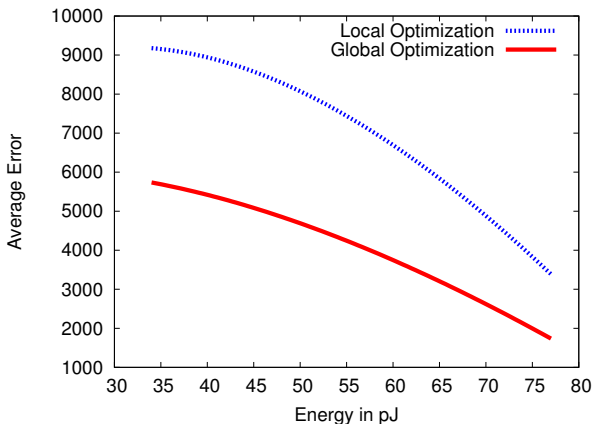


Figure 8: Energy consumption vs. Average error for a 8-point FFT with local and global optimization

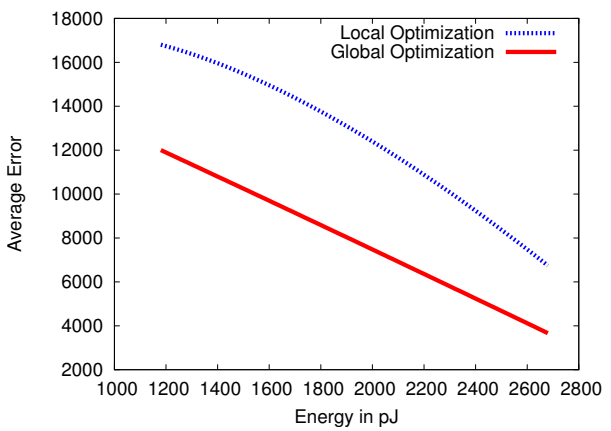
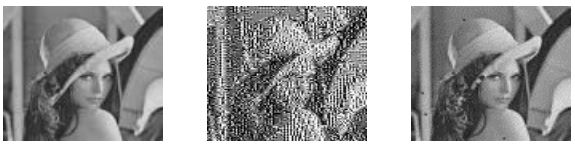


Figure 9: Energy consumption vs. Average error for a 16-point FFT with local and global optimization

is supplied to them, so that they cannot run correctly at the higher frequencies.

Considering a circuit of an adder, the question studied in [4] was how to choose voltages so as to minimize the energy for some acceptable level of the average error. In the first method, all the full adders within the adder were given the same voltage level (uniformly voltage scaled), with the resulting required energy denoted by E_{UVOS}. In the second method, various full adders were given one of four voltage levels (biased voltage scaled), with resulting required energy denoted by E_{BIVOS}. It was shown that for the same level of overclocking and average error, $E_{\text{BIVOS}} < E_{\text{UVOS}}$. This shows that in an adder, some bit positions are more important than others. But this previous work only addresses *local optimization*. Using this approach, design of circuits with more than one n-bit adder implies using BIVOS in each adder without considering the relative significance of the various adders. We will refer to this as *locally optimized*.

We propose, in this paper, to optimize the energy consumption of an approximate circuit with the optimization done using the *single resource dataflow energy-error optimization* method presented in this paper, with the resulting energy denoted by E_{global}. Thus the relative significance of an entire



(a) (b) (c)

Figure 10: Reconstructed images obtained after processing through a (a) conventional correct 8-point FFT (b) locally optimized approximate 8-point FFT (c) globally optimized approximate 8-point FFT

adder with other adders in the circuit are taken into account to invest energy in an efficient way by finding the solution to Eqn. 1. But for supply voltages internal to a single adder we use the BIVOS scheme presented in [4]. Thus, supply voltages are assigned to minimize energy considering both entire n-bit adders and components inside a single adder. We will refer to this as *globally optimized*. The results we present show that globally optimized yields better energy savings for the same expected error than locally optimized.

We do not present results for the FIR because by using the global optimization scheme we conclude that all the adders are equally important and thus energy has to be distributed uniformly across all the adders. Hence the conventional locally optimized BIVOS scheme is the best we can do.

For the case of an 8-point FFT (shown in Fig. 7), we assume a given energy budget for the entire circuit and also find the critical path delay. For each energy budget that we pick, we obtain a point that has been used to interpolate the curve in Fig. 7. The 8-point FFT has a critical path delay of ≈ 10 ns when operated at 1.2 V but it is overclocked to a frequency of 0.2 GHz (essentially the inputs and outputs are provided with an interval of 5 ns) which is faster than permitted by a conventional design methodology.

Applying Theorem 2 (Eq. 12), we find an energy budget per adder from the total energy budget of the circuit. Based on this individual energy budget, we apply the previously published BIVOS scheme for each of the 24 adders using the four supply voltages (0.7 V, 0.9 V, 1.0 V, and 1.2 V) assumed. The result for the overall 8-point FFT in HSPICE with uniformly distributed random data as input is shown in Fig. 8. For the same energy investment of 77 pJ, the globally optimized 8-point FFT has 1.95X lower error than the "locally" optimized FFT operating at the same speed. Also, global optimization gives the designer at the least 1.44X lower energy investment for the same amount of quality trade off in the FFT. Overall, the globally optimized FFT has 2.8X lower *energy-delay product* (EDP), when compared to a conventional FFT for the same error.

A similar comparison using results from our custom simulator, which has been validated with HSPICE, is presented for a 16-point FFT in Fig. 9 to show that the analysis is scalable. In this case, we achieved 2.05X lower error for the same investment of 2700 pJ in both the globally optimized and only locally optimized FFT when they are overclocked to a frequency of 0.1 Ghz (the critical path delay of the 16-point FFT at 1.2 V is ≈ 20 ns).

This phenomenon is also demonstrated in Fig. 10 which consists of reconstructed images after processing them through an approximate 8-point FFT and an inverse FFT.

The three images in Fig. 10 are the original image, the image processed through FFT with only local optimization, and the image processed through FFT with global optimization, respectively from left to right. Fig. 10(c) has an SNR which is 1.42X times higher than the Fig. 10(b) for a similar energy of 62 pJ and operating frequency of 0.125 GHz. Also Fig. 10(c) has 1.7X lower energy consumption when compared to Fig. 10(a).

7. IMPACT ON CIRCUIT DESIGN

The generality of our approach allows a circuit designer who is attempting to optimize the design of any circuit that can be represented as a graph of adders to automatically compute the *optimal* investment of energy. The design automation tool will invest energy such that the quality is computed as the "best".

Moreover, the algorithm to compute the relative importance of the adders is extremely fast. So the approach scales very easily with the size and complexity of the circuit.

8. CONCLUSIONS AND FUTURE WORK

In this paper we show that considering the relative importance of different components of a circuit is vital to realizing energy savings while improving the quality of the output. We present a general method to apply this approach to any circuit built using adders and shifters. Two instances of the savings in energy that can be expected are shown for the FFT. The present paper models only adders and considers the implicit shifting operations while computing the *amplification factor*. We are working to extend the current optimization scheme to include general multipliers.

We are also planning to extend this methodology to control-flow graphs with memory elements and feedback loops.

We plan to validate this analysis on a wide variety of popularly used circuits in embedded systems, where low-energy consumption is a primary criterion and accuracy can be traded off.

9. ACKNOWLEDGMENTS

This work was supported in part by the US Defense Advanced Research Projects Agency (DARPA) under seedling contract number F30602-02-2-0124. It was also supported by a grant from the Nanyang Technological University in Singapore through the Institute of Sustainable Nanoelectronics under Contract #09121901. We also wish to thank the anonymous referees for their reviews which helped us improve the paper.

10. REFERENCES

[1] N. Banerjee, G. Karakonstantis, and K. Roy. Process variation tolerant low power dct architecture. In *Proc. of the Conf. on Design, Automation and Test in Europe*, pages 630–635, 2007.

[2] N. Boullis and A. Tisserand. Some optimizations of hardware multiplication by constant matrices. *IEEE Transactions on Computers*, 54:1271–1282, 2005.

[3] K. Bourzac. Intel prototypes low-power circuits. *Technology Review, Published by MIT*, 2010. `http://www.technologyreview.com/computing/24843/`.

[4] L. Chakrapani, K. Muntimadugu, A. Lingamneni, J. George, and K. Palem. Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation. In *Proc. of the 2008 Intl. Conf. on Compilers, architectures and synthesis for embedded systems*, pages 187–196, 2008.

[5] T. Cormen, C. Stein, R. Rivest, and C. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, third edition, 2009.

[6] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proc. of the 36th Annual IEEE/ACM Intl. Symp. on Microarchitecture (MICRO)*, pages 7–18, 2003.

[7] J. George, B. Marr, B. Akgul, and K. Palem. Probabilistic arithmetic and energy efficient embedded signal processing. In *Proc. of the The IEEE/ACM Intl. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, pages 158–168, 2006.

[8] ITRS. International technology roadmap for semiconductors. `http://www.itrs.net/Links/2007ITRS/ExecSum2007.pdf`, 2007.

[9] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi. *Low power methodology manual: for system-on-chip design*. Springer Verlag, 2007.

[10] A. Manzak and C. Chaktrabarti. Variable voltage task scheduling algorithms for minimizing energy/power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(2):270–276, 2003.

[11] S. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Proc. of the Intl. Conf. on Computer Aided Design*, pages 721–725, 2002.

[12] K. Palem, L. Chakrapani, Z. Kedem, A. Lingamneni, and K. Muntimadugu. Sustaining Moore's law in embedded computing through probabilistic and approximate design: Retrospects and prospects. In *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, pages 1–10, 2009.

[13] B. Shim, S. Sridhara, and N. Shanbhag. Reliable low-power digital signal processing via reduced precision redundancy. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(5):497–510, 2004.

[14] J. Tschanz, K. Bowman, S. Lu, P. Aseron, M. Khellah, A. Raychowdhury, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik, and V. De. A 45nm resilient and adaptive microprocessor core for dynamic variation tolerance. In *Proc. of IEEE Intl. Solid-State Circuits Conf.*, pages 282–283, 2010.

[15] Y. Yeh, S. Kuo, and J. Jou. Converter-free multiple-voltage scaling techniques for low-power CMOS digital design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(1):172–176, 2001.

[16] Y. Zhou, J. Noras, and S. J. Shepherd. Novel design of multiplier-less FFT processors. *Signal Process.*, 87(6):1402–1407, 2007.