MULTIPLE CONSTANT MULTIPLICATION IMPLEMENTATIONS IN NEAR-THRESHOLD COMPUTING SYSTEMS

James B. Wendt, Nathaniel A. Conos, Miodrag Potkonjak

Computer Science Department
University of California, Los Angeles

ABSTRACT

Modern applications, such as video and audio processing, employ many linear transforms and filters, such as the discrete cosine transform (DCT) and fast Fourier transform (FFT). There is a need to drastically reduce the energy consumption of these applications in order to enable their ultralow energy realization on mobile devices. It is well known that all of these applications can be implemented using multiple constant multiplication (MCM). As of now, there has been no effort to synthesize ultralow energy implementations of these systems, in particular, through the use of near-threshold computing (NTC). In order to synthesize MCM in NTC systems, we propose techniques that include minimizing the critical path through load reduction along with the creation of deep combinational logic in order to reduce the impact of process variation. We have demonstrated the effectiveness of our new techniques on both FFT and DCT applications and show up to an order of magnitude reduction in energy for the target delay.

1. INTRODUCTION

Linear transforms and filters are prevalent in many types of applications in wireless and embedded systems. In particular, the discrete cosine transform (DCT) and fast Fourier transform (FFT), are important for video, image, and audio applications, as well as for many communication schemes, including orthogonal frequency division multiplexing.

It is well known that many linear transforms and filters, including FFT, DCT, FIR and IIR filters, can be implemented using multiple constant multiplication (MCM). MCM enables low power and low latency parallel multiplications of several hard-coded constants with a single input variable using shifts, additions, and subtractions. Currently, a large body of literature on MCM focuses on minimizing their number of operations, implementation size, and depth. However, as of now, there has been relatively no effort to synthesize ultralow energy implementations, in particular, through the use of near-threshold computing (NTC) [1].

NTC is a relatively new paradigm in which circuit supply voltages are operated at or near the threshold voltage. This provides as much as $10\times$ energy efficiency gains over traditional super-threshold operation while also increasing delay by the same factor. A key obstacle in designing circuits for NTC operation is process variation, which manifests as deviations in gate characteristics, such as delay and energy, from their nominal values. These deviations become more pronounced when operating at near-threshold. For example, small fluctuations in threshold voltages have a much higher impact on delay in near-threshold than in super-threshold. Output capacitive load also becomes of concern at near-threshold due to its ability to compound NTC performance inefficiencies.

In order to synthesize MCM in NTC systems, we propose several techniques which exploit several degrees of freedom. Among them, common sub-expression exploration is combined with simultaneous speed and energy organization through operation replication for load reduction to improve latency. Another degree of freedom is the creation of deep combinational logic which not only eliminates energy expensive read/write operations to storage elements, such as flip-flops, but also reduces the impact of process variation and improves yield.

2. PRELIMINARIES

2.1. Near-Threshold Computing

In modern systems, low energy is often a premiere design requirement [2] [3]. At the integrated circuit level, energy consumption and delay are often managed by altering the supply voltage relative to ground. Since the relationship between energy and voltage is quadratic, voltage scaling has become one of the most effective and researched methods for integrated circuit power reduction [4].

Today, the vast majority of circuits operate in the super-threshold region where $V_{dd} \gg V_{th}$. Techniques for sub-threshold operation of circuits in which $V_{dd} < V_{th}$ have also been proposed [5]. However, the most balanced tradeoffs between energy reduction and performance degradation are found in the near-threshold region where $V_{dd} \sim V_{th}$ [1].

The most notable design challenge of NTC is that circuits are highly susceptible to variability. Inherent variations in manufacturing processes affect threshold voltage distribu-

tions, thus imposing design constraints at near-threshold. Existing solutions attempt to mitigate the effects of process variation in the context of NTC using device aging but do not address performance degradation [6]. The major challenge in using NTC memories is high susceptibility to faults.

In NTC data path systems, the two key challenges are process variation and delay. We address these two issues by introducing deep combinational logic chains for process variation and minimizing the number of operations under a specified maximal logic delay. In particular, our delay model is novel in that it considers the loads that a particular gate drives while loads are traditionally only considered in gate-level systems. To the best of our knowledge, this is the first time that they are included in high-level (operation-level) synthesis.

2.2. Chaining

Chaining is a powerful architectural structure in which complex arithmetic or logic units are connected in succession without intermediate storage units, thus creating a chain of deep logic. Traditionally, chaining is used for reducing the required number of clock cycles. Our key novelty is that we use chaining for reducing the impact of process variation. At super-threshold voltages, delay is predominantly affected by the magnitude of the supply voltage. At near-threshold voltages, delay becomes exponentially affected by the difference between the supply and threshold voltages.

Traditional design techniques place registers between operations in order to capitalize on rapid clock rates and high throughput. However, at near-threshold operation where process variation has an exponential impact on delay, this multicycle architecture can be detrimental. Due to inherent delay variations, the clock rate is constrained to be no faster than the maximum delay of any set of operations positioned between registers. The overall circuit delay will be approximately the maximum delay times the required number of clock cycles to finish the operation to the end. By implementing the same circuit using deep logic we reduce the impact of the few extremely slow components affected by process variation through delay averaging over long chains.

2.3. Multiple Constant Multiplication

MCMs were first studied in the context of minimizing the number of operations in FIR filters [8]. Cappello and Steiglitz were the first to prove that optimal MCM generation is NP-complete [9]. Consequently, several MCM generation techniques have been identified and addressed using three different types of approaches [10] [11] [12].

To the best of our knowledge, ours is the first effort to address ultralow power implementations of MCM computations in the general case. However, several groups have addressed minimizing power in FIR filters using common subexpression elimination [13] [14].

3. ITERATIVE NODE REPLICATION FOR TARGET DELAY YIELD IN NTC

Our iterative node replication (INR) technique for delay minimization of MCM structures operating in near-threshold improves latency, yield, and reduces susceptibility to process variation through deep logic chaining and load reduction. Previous MCM design techniques that focus on operation cardinality reduction also inherently increase average component fan-out. Higher loads induce longer delays and impose design-time assignment of larger sized components, ultimately increasing energy consumption. At near-threshold voltages these consequences can outweigh the benefits of instantiating a design with a minimal number of operations. In order to improve overall circuit delay at near-threshold, load must be a preeminent design consideration.

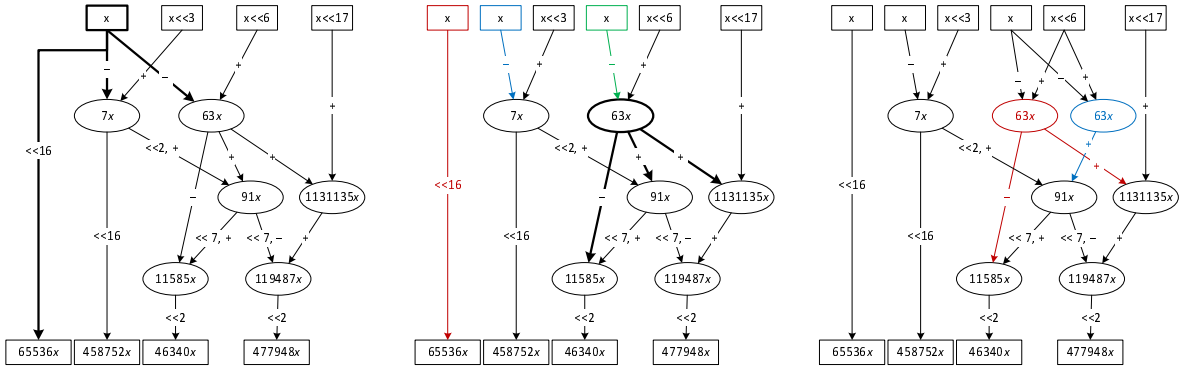
There exists a large body of literature on algorithmic solutions for MCM optimization, especially since the problem is NP-complete. We build our solution on top of the best heuristic tools currently available, specifically, the Spiral suite developed by Voronenko, Puschel, and their collaborators [7]. We begin with minimal depth MCM implementations for the set of constants corresponding to the pertinent benchmark because when operating in NTC, minimizing circuit depth is crucial for minimizing delay.

Figure 1 depicts an example of the starting, first, and second iterations of our INR technique for a single MCM tree corresponding to an FFT with 8 inputs. We describe the algorithm in more detail in the following section. All three implementations are functionally equivalent, but differ in terms of energy and delay when operated at near-threshold. In this example, we highlight the reduction in critical path delay that our replication technique achieves. After replicating input x and the $63x$ operation we observe a reduction in delay by about 300 ps. In this example we use approximate values for clarity in presentation. In simulation we apply the appropriate load-delay models for the appropriately sized adder modules.

3.1. Algorithm

Load reduction and delay minimization is accomplished by iteratively replicating operations on the critical path that have maximal load. When considering multiple nodes that have zero slack and an equal maximal load, nodes whose transitive fan-out affect the largest set of epsilon critical paths are replicated first.

Load is distributed to the new replica by prioritizing output paths with the least slack. Zero slack output paths are swapped from the old node to the newly replicated node until no more zero slack paths can be assigned or the load of the new replica is half the load of the original. In the best case, there will exist only a single path with zero slack (i.e. a single critical path), and thus only a single output path with zero slack will be assigned to the replica node. In singling out this



(a) critical path delay = 13253 ps (b) critical path delay = 12651 ps (c) critical path delay = 12350 ps

Constant (`float`)	Fixed-point Representation (`uint`)
1.0	65536
0.707107	46340
−0.707107	477948
−1.0	458752

Load	Delay (ps)
1	2937
2	3238
3	3539
4	3840

Fig. 1: Functionally equivalent MCM structures for a single variable of an 8 point FFT. The left table contains the multiplier constants for a single input in fixed-point representation using 16 fraction bits and 3 integer bits. The right table contains approximate delay values for a carry-lookahead adder (cell size 2) operating at near-threshold. (a) A minimal depth, minimal operation MCM tree created by the Spiral MCM synthesis tool [7]. Bolded nodes are those selected to be replicated in the following iteration. (b) A reconstructed MCM tree created by replicating inputs for x from the previous iteration. (c) The next iteration of the MCM tree created by replicating the $63x$ operation and balancing output load.

Benchmark	Solution	Target Delay (ns)	Energy (mJ)	Energy Savings	# Operations	Area (μm^2)
FFT–64	Spiral-MC	55.31	60.51	1.00×	2347	8.01×10^6
	Spiral-D		14.91	4.06×	2347	7.00×10^6
	INR		10.58	5.72×	2612	7.55×10^6
FFT–128	Spiral-MC	55.31	367.56	1.00×	8555	3.19×10^7
	Spiral-D		79.78	4.61×	8555	2.80×10^7
	INR		46.49	7.91×	9204	2.82×10^7
DCT–8x8	Spiral-MC	61.86	305.28	1.00×	5835	2.12×10^7
	Spiral-D		50.52	6.04×	5835	1.84×10^7
	INR		33.38	9.14×	6121	1.88×10^7
DCT–16x16	Spiral-MC	58.29	10822.23	1.00×	75234	3.12×10^8
	Spiral-D		1539.11	7.03×	75234	2.71×10^8
	INR		976.43	11.08×	76011	2.66×10^8

Table 1: Energy and area results for FFT and DCT applications synthesized using multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic synthesis tool and our iterative node replication techniques.

and only this path we are able to maximally reduce its internal delay constraints through load reduction by capitalizing on the positive slack of the node's remaining output paths.

In Figure 1b, node $63x$ is chosen for replication because it has the maximum load among all nodes on the critical path. Once node $63x$ is replicated in Figure 1c, we assign only the path to $91x$ to the newly created (blue) replica since it is the only node that is on the critical path. In this way, the replicated $63x$ node (blue) that is now a part of the path with the highest constraint will have the least load between both the red and blue $63x$ nodes, thereby reducing the internal constraints as maximally as possible. For the case that more than half the paths from the replication node have zero slack, we distribute the load evenly between the original and the replica node.

Final MCM selection is accomplished by choosing a single MCM from our generated pool of iterations that has minimum energy when satisfying the target delay.

3.2. Results

We synthesize MCM implementations of DCT and FFT benchmarks in the presence of process variation and compare our results to multi-cycle and deep logic implementations of

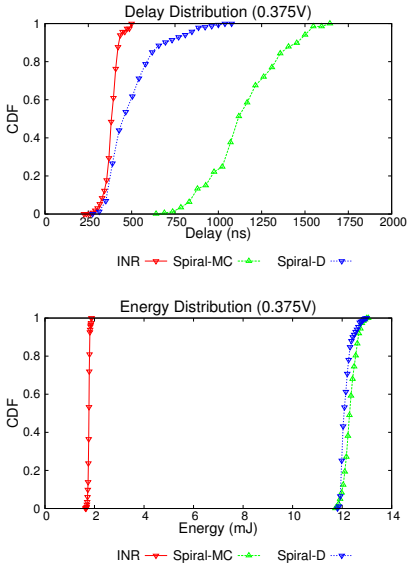


Fig. 2: DCT–16x16 circuit yield with respect to (a) delay and (b) energy in the presence of process variation when operating in near-threshold with a nominal V_{th} of 0.33V. We compare our technique with the multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic solutions.

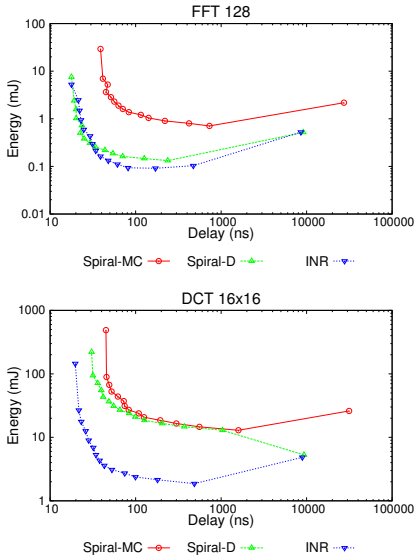


Fig. 3: Circuit energy consumption for target delays for FFT and DCT benchmark applications using multi-cycle (MC) and deep logic (D) implementations of Spiral's heuristic solutions and implementations generated using our iterative node replication technique.

Spiral solutions. A fixed width size of 16 fractional bits and 3 integer bits is used. We employ Markovic's gate level models [15] after fitting them to an industrial standard cell library [16]. Each cell is sized per capacitive load requirements and input transition slew. Our nominal threshold voltage is 0.33 V.

In Table 1 we compare nominal solutions corresponding to the delay of the multi-cycle Spiral solution when applying a near-threshold supply voltage. We scale the supply voltages for the deep logic and INR cases to achieve the same target delay and record the resultant energy consumption values and area requirements. Our techniques have an energy savings improvement ranging from 10% to 70% beyond even the Spiral deep logic solution. In some cases, this is achieved even when imposing additional area overhead.

Figure 2 depicts circuit yield with respect to energy and delay in the presence of process variation for the 16x16 DCT application operated at near-threshold. Our techniques generate simultaneously lower energy and lower delay implementations of the required MCMs for this application. We find that not only does our load reduction technique reduce circuit delay, but also simultaneously reduces energy by eliminating the need for larger cell sizes that the original Spiral solution requires due to its ultra compact and high fan-out structure.

Figure 3 depicts the energy usage for operation of the pertinent synthesized benchmark application at a desired target delay. For smaller circuits we observe similar improvements to the Spiral deep logic implementation. In the case of the 4x4 DCT, the circuit is too small for replication to have a meaningful impact before the increase in the number of operations consumes too much energy without sufficient delay improvements. However, for larger applications in which MCM synthesis becomes very complex, we observe that our techniques substantially reduce energy consumption rates. This is expected since larger complex minimum depth MCM trees will contain many operations along many epsilon critical paths, thus harboring many opportunities for load reduction exploitation.

4. CONCLUSION

We have presented new techniques for ultralow energy implementations of MCM circuits through the use of near-threshold computing. Our techniques reduce energy and delay through node replication, which directly reduces load in order to reduce the critical path and ultimately reduce energy by enabling smaller cell size assignments. Furthermore, we reduce the impact of process variation through the use of deep combinational logic. We have explored our techniques for MCM optimization on the popular FFT and DCT linear transforms in the presence of process variation using accurate gate-level models and cell sizing techniques. We find that for larger applications requiring larger and more complex MCMs, our techniques show substantial improvements in energy reduction for a range of target delays.

5. REFERENCES

[1] Ronald G Dreslinski, Michael Wieckowski, David Blaauw, Dennis Sylvester, and Trevor Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[2] James B Wendt, Saro Meguerdichian, Hyduke Noshadi, and Miodrag Potkonjak, "Semantics-driven sensor configuration for energy reduction in medical sensor networks," in *International Symposium on Low Power Electronics and Design*, 2012, pp. 303–308.

[3] Teng Xu, James B Wendt, and Miodrag Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *International Conference on Computer-Aided Design*, 2014, pp. 417–423.

[4] Padmanabhan Pillai and Kang G Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Symposium on Operating System Principles*, 2001, vol. 35, pp. 89–102.

[5] Benton H Calhoun, Alice Wang, and Anantha Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, 2005.

[6] James B Wendt and Miodrag Potkonjak, "Improving energy efficiency in sensing subsystems via near-threshold computing and device aging," in *IEEE Sensors*, 2013, pp. 555–558.

[7] Yevgen Voronenko and Markus Püschel, "Multiplierless multiple constant multiplication," *ACM Transactions on Algorithms*, vol. 3, no. 2, pp. 11–50, 2007.

[8] Henry Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, pp. 1044–1047, 1989.

[9] Peter R Cappello and Kenneth Steiglitz, "Some complexity issues in digital signal processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 5, pp. 1037–1041, 1984.

[10] Miguel R Corazao et al., "Performance optimization using template mapping for datapath-intensive high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 8, pp. 877–888, 1996.

[11] Oscar Gustafsson, Henrik Ohlsson, and Lars Wanhammar, "Improved multiple constant multiplication using a minimum spanning tree," in *IEEE Conference on Signals, Systems and Computers*, 2004, vol. 1, pp. 63–66.

[12] Markus Püschel and José MF Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1280–1316, 2003.

[13] Mahesh Mehendale, Sunil D Sherlekar, and G Venkatesh, "Algorithmic and architectural transformations for low power realization of FIR filters," in *International Conference on VLSI Design*, 1998, pp. 12–17.

[14] Ahmet T Erdogan, Mohammad Hasan, and Tughrul Arslan, "Algorithmic low power FIR cores," *IEEE Proceedings on Circuits, Devices and Systems*, vol. 150, no. 3, pp. 155–160, 2003.

[15] Dejan Markovic, Cheng C Wang, Louis P Alarcon, Tsung-Te Liu, and Jan M Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, 2010.

[16] Muhammet M Ozdal, Chirayu Amin, Andrey Ayupov, Steven M Burns, Gustavo R Wilke, and Cheng Zhuo, "An improved benchmark suite for the ISPD-2013 discrete cell sizing contest," in *International Symposium on Physical Design*, 2013, pp. 168–170.