Resilient Low Voltage Accelerators for High Energy Efficiency

Nandhini Chandramoorthy, Karthik Swaminathan, Martin Cochet, Arun Paidimarri, Schuyler Eldridge,
Rajiv V. Joshi, Matthew M. Ziegler, Alper Buyuktosunoglu, Pradip Bose

IBM T. J. Watson Research Center

ABSTRACT

Low voltage architecture and design are key enablers of high throughput per watt in heterogeneous, accelerator-rich many-core designs. However, such low voltage operation poses significant challenges because of difficulties in achieving reliable functionality of on-chip memories, particularly SRAMs at these design points. In this paper, we present a technique of low-voltage neural network acceleration, where the embedded SRAM architecture is equipped with a novel application-aware supply voltage boosting capability. This technique mitigates low-voltage induced failures, while enabling *Very low voltage (VLV)*[1] operation during most of the application run, resulting in substantial improvement in net energy efficiency. We present a framework to evaluate the impact of low-voltage SRAM errors on machine learning applications and characterize trade-offs between output inference accuracy and energy efficiency in our application-programmable supply boosted SRAM architecture. Using the proposed technique we push the limits on the minimum operable voltage (V_{min}) for the desired output quality. As a proof of concept, we demonstrate these techniques on *Dante*, a Deep Neural Network (DNN) accelerator chip taped out in state-of-the art 14nm technology.

1. INTRODUCTION

. Domain-specific accelerators that operate under very tight power constraints form a key component of architectures, particularly at the edge. Consequently, Very Low Voltage (VLV) functionality is highly desirable in such a context where operations per second per watt is a critical consideration. However, the minimum possible operating voltage, V_{min}, is limited by the operation of the on-chip SRAM [1]. At such low voltages, SRAMs do not function reliably due to bit cell variability and yield challenges and are vulnerable to failures at voltages much higher than those at which logic starts to fail.

A significant aspect of obtaining reliable *Very Low Voltage (VLV) operation* is ensuring correct SRAM functionality, particularly for custom accelerator designs. For example, accelerators for Deep Neural Networks (DNNs) generally feature a large amount of on-chip SRAM to store weights and inputs, and bit failures in these SRAM at very low operating voltages can adversely affect the output accuracy.

[1] In this paper, we adopt the term *Very Low Voltage (VLV)* instead of *Near Threshold Voltage (NTV)*, so that this may include a wider range of voltage values, instead of just those that are adjacent to the threshold voltage V_t.

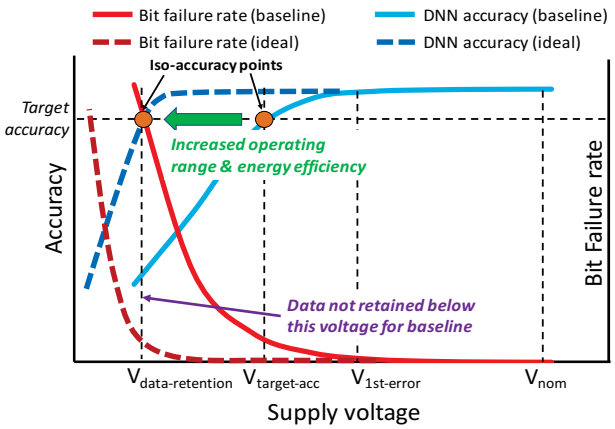

Figure 1: Effect of reducing supply voltage on SRAM bit failure rate in a DNN accelerator and the corresponding output accuracy. Operating at voltages below $V_{target-acc}$ *and* meeting the target accuracy is not possible for the baseline configuration. In contrast, for the ideal scenario, the target accuracy is met even when voltage is dropped down to $V_{data-retention}$, the minimum voltage at which the SRAM retains its stored data

Figure 1 shows a sharp increase in bit failure rates (in red), when voltage is reduced below the nominal value (V_{nom}). The first bit failures are observed at $V_{1st-error}$ and the output DNN inference accuracy (in blue) falls short of the target at voltages below $V_{target-acc}$. For the baseline configuration, it is not possible to reduce the voltage below $V_{target-acc}$ on account of the increasing bit failure rate and still meet the target accuracy, thus forcing the accelerator to operate at a voltage much higher than $V_{data-retention}$, the minimum voltage at which the SRAM retains stored data. Consequently, the significant gap between $V_{data-retention}$ and $V_{target-acc}$ prevents designers from exploiting the advantages in energy efficiency that can be gained by lowering the supply voltage. In contrast, in an ideal scenario, the accelerator can meet the target accuracy even when operating at $V_{data-retention}$. Reducing this gap between $V_{data-retention}$ and $V_{target-acc}$ through algorithmic, architecture and circuit-level techniques would result in superior levels of energy efficiency without compromising on output accuracy.

In accelerator designs with a single shared supply for logic and memories, this shortcoming in energy efficiency is exacerbated. SRAM reliability issues in single supply designs

2378-203X/19/$31.00 ©2019 IEEE
DOI 10.1109/HPCA.2019.00034

dominated by logic power preclude the possibility of lowering supply voltage to improve energy efficiency. To provide distinct SRAM and logic voltages, different types of on-chip or off-chip regulators can be used. The latter increases packaging costs, has a large regulation time constant, and limits system-level integration, which is critical for devices with constrained form factors. Buck-boost converters [2] can provide up to 90% efficiency, but require off-chip inductors. Fully on-chip switched-capacitor regulators are limited to $<$80% efficiency [3], without advanced processes such as deep-trench capacitors [4], or complex circuit techniques to accommodate supply ripple [5]. Further, charge pumps [4] have high response times and cannot provide fine-grained control of SRAM voltage. Linear regulators such as Low Dropout Regulators (LDOs) provide on-chip fine-grained voltage control but suffer from decreasing efficiency when the difference between SRAM and logic voltage increases [6].

Supply voltage boosting is a potential method of extending the operating voltage range of SRAMs [7]. Here, logic and memories share a single power supply and the SRAM power supply is *boosted* to a higher voltage dynamically to enable reliable read or write operation at low voltages. Voltage boosting helps to effectively lower the V_{min} of SRAMs leading to lower operating voltage for the entire chip while still ensuring reliable memory operation. However, this technique has only been implemented on standalone specialized 8T SRAM arrays, as opposed to the universally adopted 6T SRAM arrays. Furthermore, the adoption of this technique has not yet been examined in an end-to-end architectural solution such as DNN accelerators. Based on our evaluations of on-chip memories in DNN accelerators, we observe that DNN accuracies drop sharply at VLVs. Mitigating these errors requires much higher margins of boost voltage than is possible with the existing techniques described in [7]. We also observe that the output accuracy of neural networks exhibits varying degrees of sensitivity to errors in different types of data, such as inputs and network weights of different layers. Hence there is a need to selectively vary the effective supply voltage across different banks or groups of SRAM, depending on how sensitive output accuracy is to the data stored in a given SRAM bank. However the capability for such fine-grained control does not exist.

In order to address the above concerns, we propose a novel programmable architecture that *dynamically boosts the supply voltage of SRAMs* to different target voltages. This comprises of a programmable booster circuit for SRAMs, capable of boosting the SRAM supply to a higher value and at a lower cost as compared to prior works. The proposed design is capable of achieving up to 50% peak boost in supply voltage and can be deployed to work with a wide range of vendor IP-compilable commodity 6T-SRAMs as well as specialized 8T and 10T SRAMs. In contrast, the designs implemented in [8, 7] are only capable of a fixed, smaller level of boost across the entire SRAM array, and cannot be configured depending on application requirements or the region of memory being accessed.

In our design, boosting is carried out only during SRAM read/write accesses, and the SRAM remains un-boosted during inactive cycles. In addition, different regions/banks of the SRAM can be boosted to target voltages independent of the other. The voltage to which the SRAM is to be boosted is determined by the application requirement, effectively handing the application control of memory power consumption as well as accuracy. With the help of the proposed fine-grained programmable boosting architecture, it is possible to reach the target accuracy at the lowest possible energy overhead at each supply voltage, while also widening the operating range of the accelerator. Note that such fine-grained temporal and spatial voltage adjustment is difficult in state-of-the-art Dynamic Voltage Frequency Scaling (DVFS) schemes using voltage regulators, due to significant latency, area and power overheads.

To validate these concepts, we designed *Dante*, a chip consisting of a DNN accelerator with voltage-boosted SRAMs taped out in 14nm bulk-FinFET technology. This paper elucidates the design considerations involved in architecting the programmable voltage-boosted memory implemented in this chip. The chief contributions of this paper are as follows:

- A resilient SRAM-driven architecture with programmable voltage-boosting capability, to enable fine-grained temporal and spatial control of application accuracy and memory power consumption at the granularity of individual SRAM banks or macros. This includes a low overhead runtime-configurable architecture for boosting, with dedicated boost circuitry per SRAM bank and a novel boost configuration module to enable such fine-grained control.
- A detailed accuracy model for studying the effects of SRAM bit flips in neural network applications across supply voltages. The energy and accuracy trade-offs of dynamic bank-level programmable boosting are evaluated for 6T commodity SRAMs across a wide range of supply voltages, based on real hardware measurements.
- A taped-out chip consisting of a DNN accelerator with voltage-boosted SRAMs. This is the first time boosting techniques have been demonstrated as part of a full accelerator architecture as opposed to standalone SRAM arrays.

For supply voltages ranging from 0.34V to 0.46V, boosting results in up to 26% and on average 17% energy savings compared to having dual supplies for logic and memory to achieve within 2% of the maximum accuracy for the convolution layers of *AlexNet* [9]. Boosting results in 30% energy savings compared to having a single supply for logic and memories that achieves the same accuracy and a 32% savings in leakage energy per cycle on average, for the taped-out DNN accelerator.

2. SRAM BIT ERRORS IN DNNS

Deep learning workloads display a tolerance to bit errors due to incorrect behavior of memory cells, up to a certain level. Factors such as the type and depth of the neural network, the layer in which the fault occurs, determine the degree to which the quality of inference is affected by the fault. We built a Tensorflow-based [10] fault-injection framework to examine the effect of bit-flips on output accuracy, using a methodology described in detail in Section 5. A fully connected DNN (FC-DNN) with 4 layers of size $784 \times 256 \times 256 \times 256 \times 32$, similar to the one evaluated in [11], is chosen as a illustrative example. Inference accuracy is computed from running 5000 test images of the MNIST dataset [12]

on TensorFlow. On account of the inter-cell variation in parameters such as the threshold voltage (V_t), there is variation in the vulnerability across the bitcells when the supply voltage (V_{dd}) is dropped. This bit-cell variation amounts to an average bit error rate measurable at each voltage for a given SRAM macro. Figure 2 shows the measured SRAM bit error rate and the result of injecting bit flips in weights and inputs of the FC-DNN. The variation of output accuracy with bit-errors at each voltage provides us with insights into the design and requirements of a memory architecture aimed at maximizing accuracy.

• We observe that for the voltage range shown, injecting faults into weights of all layers causes a drastic drop in accuracy. A steep slope can be observed in the voltage range 0.4V to 0.46V.

• For the same bit error rate, say at 0.014 at 0.44V, bit flips in an input image used for inference cause only a 1% drop in accuracy as opposed to 44% decrease due to faulty weights. This shows that the FC-DNN is more tolerant to bit flips in inputs compared to weights.

• It is also interesting to observe the effect of injecting faults in each layer weights individually, as shown from results of injecting selectively into the first and last weight layers. Injecting faults into the first layer seems to have a higher impact on output accuracy than the last layer due to the larger number of weights in the first layer and accumulation of faulty computations from the first hidden layer to the last.

We observe similar trends in the behavior of other types of neural networks that we evaluated, such as Convolution Neural Networks (CNNs). These observations lead us to the following design specifications for the voltage-boosted memory architecture.

• **How much to boost** From the graph it can be seen that supply voltage should be at least 0.46V for maximum accuracy for a target accuracy within 2% of the accuracy at nominal voltage for the given application. This means that the booster architecture should be capable of achieving the target voltage $Vddv$ from any supply voltage, $Vdd < 0.46V$ such that the desired accuracy level is reached at the lowest energy cost. This work presents a novel SRAM booster architecture to achieve such high values of boosted voltage ($Vddv$ - Vdd) for very low supply voltages, resulting in increased energy efficiency.

• **When to boost** The proposed solution adjusts SRAM voltage dynamically during each read and write, without incurring additional latency. Thus there will be substantial savings in leakage energy, particularly in memory dominated architectures. The approach presented in this paper involves boosting the SRAM power grid only, while having a lowered supply voltage for logic in the accelerator. An equivalent comparison point would be that of linear voltage regulation with the higher supplied voltage set for memories and a lowered voltage for accelerator logic using Low Drop-Out (LDO) regulators. However LDOs are inefficient due to energy loss in resistors, with efficiency proportional to the ratio between output and input voltages, and have an operational latency. We shall prove that our solution is significantly more efficient in terms of dynamic energy compared to using state-of-the art LDOs.

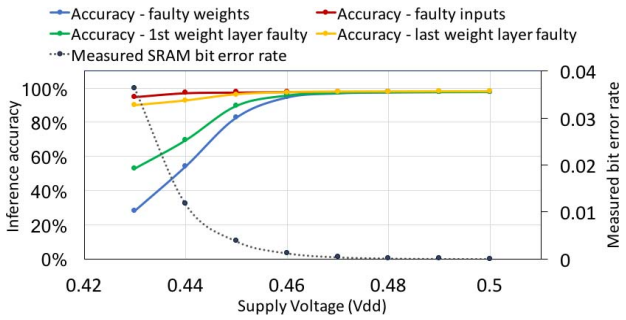

Figure 2: Effect of fault injection in inputs and weights of all layers and selective injection into individual layers for the MNIST network, on output inference accuracy, as modeled using the fault modeling framework described in Section 5. The hardware measured SRAM bit error rate used for fault injection is also shown in the figure

• **What to boost** Since the inference accuracy of the neural network is sensitive to weights and inputs as seen in Figure 2, there is immense potential in an architecture that allows for application-controlled voltage adjustment. With different types/classes of computation data such as inputs and weights stored in an accelerator's local memory boosted to different voltages, as dictated by the accelerator, desired accuracy levels can be reached at the lowest energy cost. Our solution offers both *spatial* and *temporal* programmability. The accelerator controls the amount of boosted voltage, setting it to one of many possible adjustment levels at *each bank* dynamically. The sensitivity of output inference accuracy to weights of different layers is translated to one of the possible boost levels and programmed using a separate instruction as described in detail in Section 3. Thus, application-aware voltage adjustment in our solution enables different data bits to be accessed with a higher degree of reliability, translating to a higher level of output accuracy. Such dynamic *spatial* and *temporal* voltage control is highly complex and inefficient using conventional voltage regulation methods.

3. BOOSTED SRAM ARCHITECTURE

This section describes the boost-enabled memory architecture and illustrates the features that make dynamic, bank-level, accelerator-programmable voltage adjustment possible. We widen the operating range by boosting to the target voltage or to the configuration with the lowest energy overhead that achieves the desired output accuracy. In addition, our architecture also has the capability to boost each SRAM bank differently, as well as adjust the amount of boost for a given bank, potentially for each read and write. This is beneficial to accelerators since memory voltage can be varied with application phase or the type and sensitivity of data being processed in different stages of computation. From the application's viewpoint, the output sensitivity to data within an address range is translated to one of the P levels of boost for the SRAM bank corresponding to that address range.

Figure 3 shows the overall architecture of a dynamic boost-enabled SRAM. The design consists of the SRAM with no modifications to any internal circuitry, augmented with a *Booster Circuit*. The booster circuit consists of a column of *Booster Cells* and the *Boost Input Control* block for pro-

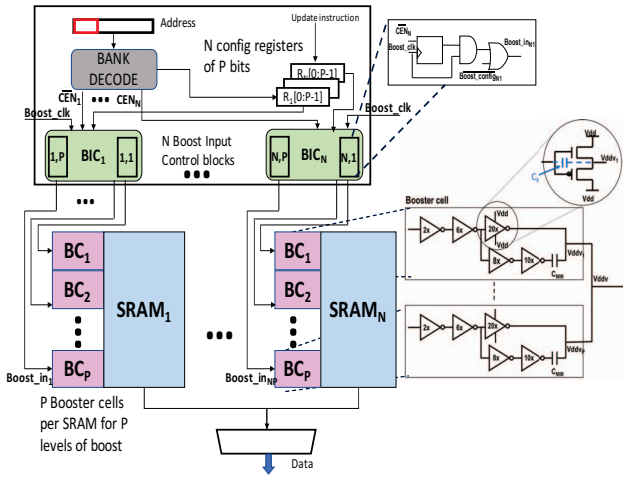

Figure 3: Architecture of SRAM array with boost controller, programmable boost circuitry, and MIM capacitor design. The drive strengths of the inverter buffers in the booster cells are also indicated in the figure

grammable, fine grain voltage adjustment.

3.1 Booster cell operation

The basic boost device unit, which we refer to as a *boost inverter* (circled in Figure 3), is identical to a standard cell inverter, but with both transistor sources connected to Vdd. The drains are connected together forming $Vddv$, the boosted output supply voltage, which is connected to the SRAM power supply grid. During un-boosted operation, the SRAM power is supplied through the turned on pFET device of the boost inverter. As the boost input swings from low to high, a boost in $Vddv$ is generated due to capacitive coupling between the gate and drain of the nFET and pFET. An estimate of the additional boost voltage is obtained using the equation:

$$V_b = Vdd * C_b/(C_b + C_{mem} + C_p) \quad (1)$$

where, V_b is the amount by which voltage is boosted and is equal to $Vddv - Vdd$, C_b is the equivalent boost capacitance of the boost inverter, and C_{mem} is the capacitance of the memory macro and C_p is a parasitic capacitance on the $Vddv$ node from the boost inverter. In this work, we have included a *Metal-Insulator-Metal (MIM) capacitor* C_{MIM} parallel to the booster cells. This increases the total boost capacitance to $C_b + C_{MIM}$ and thus the total boost voltage, and at a much lower cost quantified in Section 3.2.3. The overall circuit is referred to as the *Booster Cell* (BC) in Figure 3.

3.2 Boost-enabled Memory Architecture

Dynamically boosting supply voltage to different levels is achieved by means of increasing or decreasing the number of boosters or the amount of boost capacitance connected to the voltage supply. For P levels of voltage boost at a given supply voltage Vdd, P booster cells are used, with outputs shorted and connected to the SRAM power supply grid. A booster cell can be turned on or off individually using the Boost Input Control block (*BIC*) shown in Figure 3. Turning a booster cell on or off increases or decreases the boost capacitance connected to the SRAM power supply, thereby adjusting the amount of boosted voltage. Each SRAM bank's power supply is connected to a separate set of booster cells, the inputs of which are controlled individually using *BIC* blocks, to enable or disable boosting at the bank level.

3.2.1 Boost Input Control block

The Boost Input Control (*BIC*) block turns a booster cell on or off, adjusting the boost voltage in small steps in the process. For N banks and P booster cells per banks, $BIC_{n,p}$ generates a $Boost\_in_{n,p}$ for controlling the p-th *BC* for the n-th bank. The *BIC* block for a given bank generates *Boost_in* from the application-programmable input configuration bits *Boost_config*, active-low bank read or write enable signal *CEN*, and the *Boost_clk* signal. A booster cell P_i is enabled only when the corresponding configuration bit $Boost\_config_i$ is set to 1. The input to the booster cell $Boost\_in_i$ is low when enabled, turning the pFET on. When there is no memory activity (high *CEN*), the booster cell output is not boosted and fixed at Vdd. During a read or write access, the output of the enabled booster cells is boosted to $Vddv_i > Vdd$ during the high phase of *Boost_clk*. The amount of boosted voltage depends on the number of enabled booster cells. When P_i is disabled by setting the corresponding configuration bit to 0, $Boost\_in_i$ stays high, turning the nFET on and the output stays at a voltage slightly lower than Vdd.

Accelerator Control: Programmability is achieved by setting the configuration bits shown in Figure 3. There are *P* configuration bits corresponding to *P* booster cells for each SRAM bank. For example, with 4 boost levels, '1111' corresponds to all 4 booster cells of a given SRAM bank enabled, and '0000' corresponds to all 4 booster cells disabled. The target boost level is written into the configuration registers dynamically via a special *set_boost_config* instruction by the accelerator. This instruction sets the target boost level for all subsequent accesses to a given bank until it is re-written.

Data layout: The spatial and temporal tunability of the boost architecture makes the design flexible and independent of data layout, banking or interleaving schemes. This means that using the *set_boost_config* instruction, a) a given SRAM can be boosted to different target voltages according to computation stage b) multiple SRAMs can be boosted to different target voltages as long as they are independently controlled by separate *BIC* blocks. Accelerators typically fetch each layer of weights from the DRAM and store them in the local SRAM. Our architecture provides the capability to set different boost levels while processing weights of different layers of a DNN. Similarly, different regions of local memory accessible using dedicated output ports can be subject to fine-grain voltage adjustments to different target voltages. Thus weights of different layers, inputs or intermediate computations such as partial products, as well as control/configuration bits can be accessed with varying degrees of accuracy, as dictated by the application. If data with different levels of sensitivity reside in the same SRAM bank, the *set_boost_config* instruction is issued to modify the bank's configuration bits before access to each type of data. In order limit the overhead, the *set_boost_config* instruction must be issued at relatively large intervals. Storing data of different sensitivity (such as inputs and weights) in different regions of memory controlled by a *BIC* block does not increase the number of memory accesses overall,

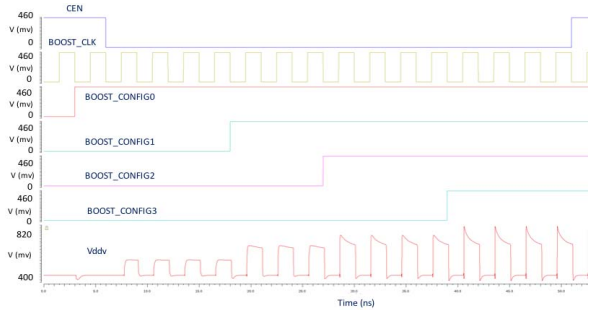

Figure 4: Simulation of the booster circuit depicting 4 programmable boost levels in output waveform Vddv, as configuration bits are changed dynamically

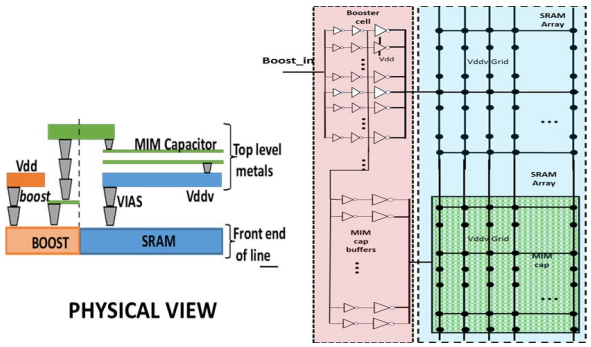

Figure 5: Layout view of SRAM array along with the boost circuit and the MIM capacitor

but keeps the number of *set_boost_config* instructions small.

Figure 4 illustrates 4 levels of boosted supply $Vddv$ waveforms that are obtained as boost configuration bits are dynamically set. The simulation was carried out in Cadence Spectre [13]. The circuit consists of 4 booster cells and their input controllers and each booster cell consists of a 10pF Metal-Insulator-Metal (MIM)-capacitor, 64 boost inverters and their buffers. This shall be referred to as the *standard* configuration in the rest of the paper. As shown in the waveforms, supply voltage adjustment happens within a cycle, unlike voltage regulators that incur high latency. This ensures that boosting does not increase the SRAM access latency. The simulation shows 4 levels of boosted voltage with increments of the order of 50mV, but much finer granularity with more boost levels can be implemented as well.

3.2.2 Design considerations

As shown in Figure 5, each booster cell is sized with the number of boost transistors and buffers to harness the required capacitance, forming a column of boosters laid out alongside the SRAM macro. *No internal modifications to the SRAM are required*. The booster inverter design itself can be easily derived from a standard inverter layout by connecting the nFET to Vdd instead of ground. This allows for a simple and repeatable design using standard cells present in all technology libraries. In addition, the high drive-strength and fin-count (≈80 fins) of the boost logic mitigate process mismatches and other effects responsible for PVT variations.

A *Metal-Insulator-Metal (MIM) capacitor* C_{MIM} comprises of an insulator (oxide) between two metal plates. Its main advantage is that it provides a larger capacitance, while leveraging the higher metal layers of the process technology, without contributing to routing congestion in the lower layers. The MIM capacitor can thus be designed entirely in higher metal layers and placed on top of the SRAM macro, as shown in Figure 5, with no additional area overheads.

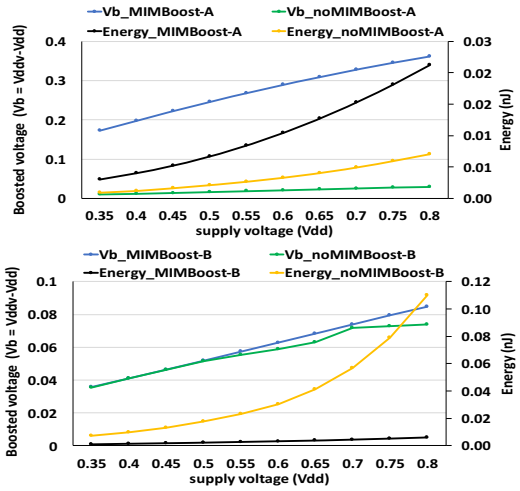

Figure 6: Boosted voltage and power for boost circuits with and without MIM capacitors. MIMBoost-A and noMIMBoost-A have equal area and generate roughly the same boosted voltage at lower voltages. However noMIMBoost-B is 8× the area of MIMBoost-B

3.2.3 Advantages over existing boost designs

To quantify the benefits of using a MIM capacitor in the boosted circuit over the boost-inverter only circuits in [7, 8], we created boost circuits with and without MIM capacitors with equal area (*MIMBoost-A*, *noMIMBoost-A*) that generate roughly the same boosted voltage (*MIMBoost-B*, *noMIMBoost-B*). *noMIMBoost-B* consisting of 8192 boost inverters with buffers, is 8× the area of *MIMBoost-B* which consists of 256 boost inverters with buffers and a MIM capacitor of 4.2pF with buffers. *MIMBoost-A* is the *standard* configuration described in Section 3.2 consisting of 256 boost inverters with buffers, and a MIM capacitor of 40pF with buffers, while *noMIMBoost-A* consists of 1024 boost inverters with buffers. In case of MIM-capacitor based circuits, the computed area includes the boost inverters, buffers for both the MIM capacitor and boost inverters, as well as the MIM capacitor itself. Figure 6 shows that *MIMBoost-A* generates 14× the boosted voltage for the same area compared to *noMIMBoost-A*. *noMIMBoost-B* is also similarly disadvantaged, expending 10× the energy as *MIMBoost-B* generating roughly the same boosted voltage. Summarily, MIM-capacitor booster circuits generate a larger boost at lower energy and area.

3.3 Evaluation of boosted SRAM

Figure 7 (top) shows bitcell failure rates at very low voltages, obtained from hardware measurements on a test chip comprising of a standalone 4Mbit server-class 6T SRAM in 14nm technology. Bit error rates were measured on this test site in the absence of any boosting mechanisms. The SRAMs chosen for this measurement at low voltages are

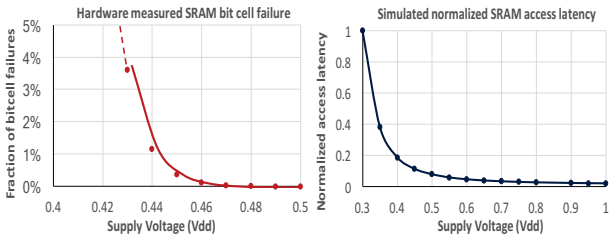

Figure 7: Variation in bit failure rate with supply voltage, measured on a standalone 4 Mbit SRAM test chip (left), and the normalized SRAM access latency with supply voltage, simulated in Cadence Spectre (right)

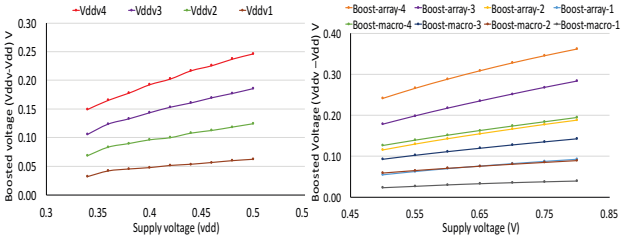

Figure 8: Peak boosted voltage for low (left) and high (left) V_{dd} for a 32Kbit SRAM macro for 4 boost levels

those that have zero bit fails at 0.6V. It can be seen that the failure rate increases exponentially with decreasing voltage. Figure 7 (bottom) also shows the access latency of a 32Kbit dual port SRAM. The memory access latencies at each voltage point was obtained using the Cadence Spectre simulator. The programmable booster circuit described in Section 3.2 which dynamically boosts SRAM supply voltage only during reads and writes, can be used to improve failure rates at very low voltages, as well as decrease access latency at high/nominal voltages without significant power overheads.

3.3.1 Very low voltages

Increased SRAM bit cell error rates at low voltages directly translate to a drop in application accuracy. Since voltage is scaled only during reads and writes, supply voltage boosting at very low voltages will improve bit cell failure rates at a low energy cost. Figure 8 (left) shows the boosted voltage at very low voltages for 32Kbit dual ported SRAM for each level of programmable boost using the *standard* configuration described in Section 3. $Vddv4$ refers to the highest boost level and $Vddv1$ refers to the lowest.

Figure 8 shows that the peak boosted voltage increases monotonically with increasing supply voltage. Hence one would have to use different boost levels at different supply voltages in order to achieve different target voltages (and hence target accuracy) required by different applications. It would thus be impossible to attain the target accuracies required by the application using static boosting schemes with a single boost level for every voltage.

3.3.2 High voltages

At supply voltages higher than the vendor-specified minimum voltage for reliable operation, SRAM boosting offers advantages in the way of improving performance by reducing access latency at lower power costs. At higher supply voltages, logic in a chip can be pipelined to drive up the operating frequency. However, SRAM access latencies do not scale proportionally. Obtaining single-cycle access to SRAMs arrays at very high frequencies becomes challenging without a custom re-design of the array, since the array access cannot be broken down further into pipeline stages. Supply voltage boosting only during SRAM access can reduce the access latency for such deeper pipeline designs.

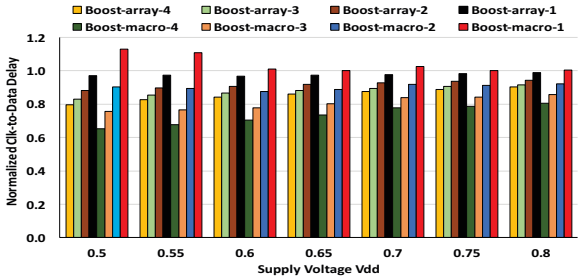

Figure 9: Normalized SRAM access latency when only the array is boosted, and when the macro (including peripherals) is boosted to different programmable levels

Figure 8 (right) shows the boost obtained at supply voltages greater than 0.5V. The amount of boost $V_b = Vddv - Vdd$ and SRAM access latency at each level of programmable boost are shown in the figure. The SRAM consists of the data array and peripheral logic such as address decoders. We explored the following classes- a) macro level boosting, where both the data array and the peripheral logic are connected to Vddv b) array-level boosting where only the array power supply is connected to Vddv and the peripheral circuits are powered at constant supply voltage Vdd. The configurations *Boost-macro-p* refer to p^{th} level of programmable boost obtained when the entire macro is boosted and *Boost-array-p* configurations refer to boosting only the array. The macro access latency values in Figure 9 are expressed as a fraction of the access latency at each unboosted voltage Vdd. Boosting the peripheral circuit reduces the access latency further as seen in the Figure 9, however V_b is reduced, as seen in Figure 8, due to additional load capacitance offered by peripheral logic. For example, boosting peripheral logic and the array leads to a maximum of 35% reduction in overall macro access latency at 0.5V. However, it is only beneficial to boost the peripheral logic at the cost of incurring additional power overheads and reduced boosted voltage, if the objective is overall reduction in access latency.

4. DNN ACCELERATORS WITH BOOSTED SRAMS

In order to validate the concept of programmable supply boosting for an accelerator in hardware, and highlight the design challenges, we designed and taped out *Dante*, a chip consisting of a Deep Neural Network (DNN) accelerator and SRAMs with the supply voltage boost circuit described in Section 3.2. The 2.3 mm^2 chip was designed using the 14nm technology node with vendor compiler-generated SRAMs IPs. The booster technology is agnostic to the process technology as well as the standard cell libraries and can be effec-

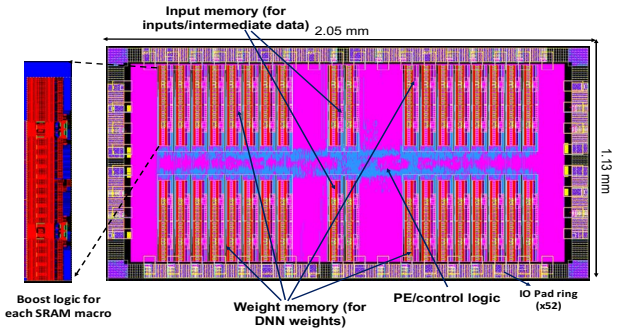

Figure 10: Layout of *Dante*, a DNN accelerator chip showing SRAMs, boost logic, Processing Element (PE) and control logic, and I/O pad ring

Table 1: Configuration parameters for DNN chip

Chip dimensions	2.05 mm x 1.13 mm in 14 nm tech
Size of memories	Weight memory: 128 KB Input memory: 16 KB
Target frequency	330 MHz for $V_{dd} = 0.8V$ 50 MHz for $V_{dd} \leq 0.5V$
Target voltage range	0.34V to 0.8V
Booster Configuration	Programmable circuit with 4 programmable levels
Booster area	0.0039 mm^2 per SRAM macro
MIM capacitance	40 pF per SRAM macro

tively implemented on other technology nodes using most commercially available vendor memory IPs. The choice of a Deep Neural Network accelerator was to demonstrate accuracy vs energy trade-offs involved in programmable supply voltage boosting. The architecture is an enhanced version of an existing design, a dynamically allocated, multi-context neural network accelerator architecture, *DANA* [14].

The processing elements (PEs) in the taped out design consist of logic for multiply-and-accumulate and activation function computation. The memory consists of 144 KB of on chip SRAM, constructed using 36 4KB (512 $\times$ 64 bit) SRAM macros. This includes a weight memory of 128KB and a 16KB input memory. Table 1 details the various configuration parameters used in the taped out design.

The MIM capacitor-based programmable boost circuit providing 4 levels of supply voltages described in Section 3.2, boosts each SRAM bank of size 64Kbit (2 macros) to a different supply voltage using its corresponding configuration bits. The Boost Input Control block described in Section 3.2 operates on the configuration bits and the *CEN* signal for each bank to generate an input signal for its booster cells, which in turn modify the supply voltage dynamically for each read and write access of the SRAM bank. The overall chip layout is shown in Figure 10. The chip consists of a continuous Vdd power grid across higher metal layers extending uniformly all over the chip, and a partitioned, isolated Vddv power grid on each SRAM macro across the lower metal layers. In order to control the voltage at the granularity of each individual bank of the weight memory, and input memory independently, a separate set of booster cells controlled by *BIC* blocks are added to each macro. The output of the booster cells are routed to the power supply of each SRAM.

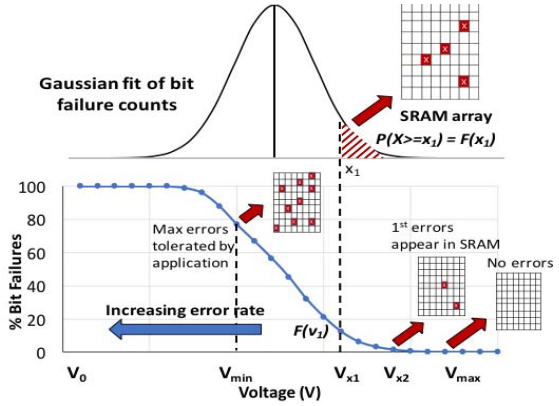

Figure 11: Illustration of fault model used to determine the errors that occur in the SRAM array at different V_{dd}s

Dedicated booster circuits and isolated power grids lead to a huge reduction in leakage energy. This can be attributed to only boosting the active SRAMs during a read or write, and all other idle SRAMs remaining at unboosted supply Vdd.

In the taped-out chip, memory accesses within the same layer are boosted uniformly, and to different voltages across layers, depending on the target accuracy. The weight and input memories are also boosted to different target voltages independently. Having such a configurable boost architecture also allows us to boost inputs to a lower voltage, just high enough to ensure reliable operation. This is described in detail in Section 6. The total on-chip memory of the taped out chip is restricted to 144KB. However, boosting techniques can be scaled and designed for much larger memory sizes of the order of several MB, which would enable an examination of larger, more complex deep neural networks as well. By adopting a banked architecture with per-bank boost circuitry similar to the one described in this paper, one can ensure reliable boosting functionality within a single clock cycle.

5. ACCURACY AND ENERGY MODELING METHODOLOGY

In this section, we describe the methodology used for determining the energy consumption and the fault model used to characterize SRAM behavior at very low voltages.

5.1 Quantifying low voltage error effects

The following methodology is used in order to quantify the resilience of the neural network configurations, when run on an accelerator operating at very low voltages. We characterize low voltage SRAM errors based on measurements across multiple die for a server-class memory test chip, as shown in Figure 7.

The objective of our modeling methodology, shown in Figure 11, is to translate the measured bitcell failure rate at each voltage into a fault map of bit positions. The bitcell failure rate $F(v)$ at various voltage levels from (V_{min} to V_{max}) is obtained by fitting failure data measured across different memory banks to an exponential distribution. Variation of V_t across the array can be assumed to be normally distributed, hence we model vulnerability of each bitcell by representing it as a random number from a normal distribu-

tion. For a given supply voltage v_{x1}, a bit cell is *'faulty'* if the generated random value x, is greater than x_1 such that $P(X \geq x_1) = F(x_1)$, where $F(x_1)$ is the probability of bitcell failure at voltage v_{x1}. This ensures the fault model is *inclusive*, that is, failures present in a fault map at voltage V_1 will also include failures present at voltage V_2, where $V_1 < V_2$.

When the faulty bitcell is read, the output is non-deterministic, since not every bitcell fault would manifest as a bit flip error during read. Hence, the probability of a bit flip, in a *faulty* bitcell is p, assumed to be 0.5 by default. The fault map thus generated, is overlaid with the SRAM array to obtain a new corrupted set of weights and activations used for inference. Since, in addition to the fault probability, the overall accuracy is also dependent on the exact location where the faults occur in the SRAM array, we carry out Monte Carlo simulations by generating 100 different fault maps and determine the average accuracy when each of these fault maps is applied to the SRAM array. We imnplement this fault modeling methodology in TensorFlow [10] to flip bits based on measured SRAM bit error rates at each supply voltage, for inputs and weights of different layers for DNNs.

5.2 Energy Modeling Methodology

The energy numbers for processing element logic were obtained post route and extraction on 14nm technology using Cadence Joules [15]. The PEs consist of logic for multiply-and-accumulate and activation function computation. The SRAM energy was obtained from detailed Cadence ADE Spectre simulations [13]. There are 3 different power supply configurations for which we estimate energy consumption.

• **Single supply configuration:** For a single power supply and on-chip memory of a given size, the dynamic energy cost is computed as:

$$DE = SRAMAcc \times E(SRAM, Vdd) + NC \times E(PE, Vdd) \quad (2)$$

where $E(SRAM, Vdd)$ refers to the energy per access of the SRAM at supply voltage Vdd as measured from Spectre simulations, $E(PE, Vdd)$ is the energy of a processing element obtained from Cadence Joules post route, *NC* and *SRAMAcc* denote the number of compute (multiply and accumulate) operations and SRAM accesses respectively. The on-chip memory is composed of banks of size 64 Kbit and the energy cost of banked SRAM access also includes the multiplexer cost. The number of accesses refers to the actual number of unique accesses to a given region of SRAM, excluding the re-use of data within the processing element once it is fetched from the SRAM. Leakage energy is estimated as the sum total of leakage energies of the two components.

• **Boosted configuration:** Using Spectre-based circuit simulations, the following quantities are estimated a) Energy consumed by the programmable booster circuit, b) boost input control circuit and c) SRAM energy with the array supply driven by the booster circuit output $Vddv$.

$$DE = \sum \Big(SRAMAcc_i \times \big(E_{boost}(SRAM, Vddv_i) + E(BC, Vdd) \big) \Big) + \big(NC \times E(PE, Vdd) \big). \quad (3)$$

where $E_{boost}(SRAM, Vddv_i)$ refers to the energy cost of a single SRAM access boosted to level i of target voltage $Vddv_i$, *E(BC,Vdd)* refers to energy cost of the booster circuit and the input control block, at supply voltage Vdd and generating target voltage $Vddv_i$. $SRAMAcc_i$ refers to the number of accesses to a region of SRAM that is boosted to target voltage $Vddv_i$, one of the possible voltage levels using the programmable boost circuit described previously. Leakage energy per cycle (LE) is estimated as:

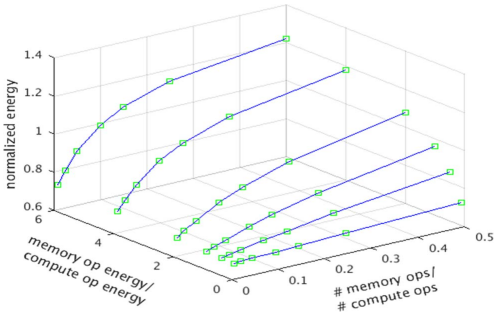

Figure 12: Comparison of boosted and dual-Vdd configurations for a range of architectural parameters. The z-axis shows the ratio of energy of an accelerator with boosted memories to energy of an accelerator with dual supplies. The x-axis shows the ratio of memory accesses to compute operations and the y-axis shows the ratio of energy of a single memory access to a compute operation.

$$LE = LE(SRAM, Vdd) + LE(BC, Vdd) + LE(PE, Vdd) \quad (4)$$

It needs to be noted that for the results shown in Section 6, the energy cost of off-chip memory access (when the DNN is loaded into the on-chip SRAM) is not included. All experiments were simulated at a constant frequency of 50 MHz.

• **Dual-supply configuration:** Separate voltage supplies for logic and memory is an alternative to boosting the supply voltage to create higher supply for SRAM,. We examine a scenario with two different power supplies - V_h powering memories and V_l, the power supply for logic obtained using a Low Drop-Out voltage regulator (LDO). The overall efficiency η of the LDO is given as:

$$\eta = (V_l / V_h) \times \eta_i \quad (5)$$

where η_i is the current efficiency and is ≈99% for state-of-the-art LDOs [6]. The dynamic energy is modeled as:

$$DE = SRAMAcc \times E(SRAM, V_h) + NC \times E(PE, V_l)/\eta \quad (6)$$

where $E(SRAM, V_h)$ is the energy of a single SRAM access at voltage V_h, $E(PE, V_l)$ the energy of the processing element logic at voltage V_l which is the output of the LDO and η the overall LDO efficiency. Since the current efficiency is close to 100%, the overall efficiency becomes a function of (V_l/V_h). The leakage energy per cycle is computed as:

$$LE = LE(SRAM, V_h) + LE(PE, V_l)/\eta \quad (7)$$

6. RESULTS

6.1 Understanding the boost-enabled accelerator design space

Supply-voltage boosting can be used to improve the energy efficiency of *any accelerator with on-chip SRAM*. Most common accelerator designs can be modeled in terms of the following architectural parameters. a) Ratio of number of

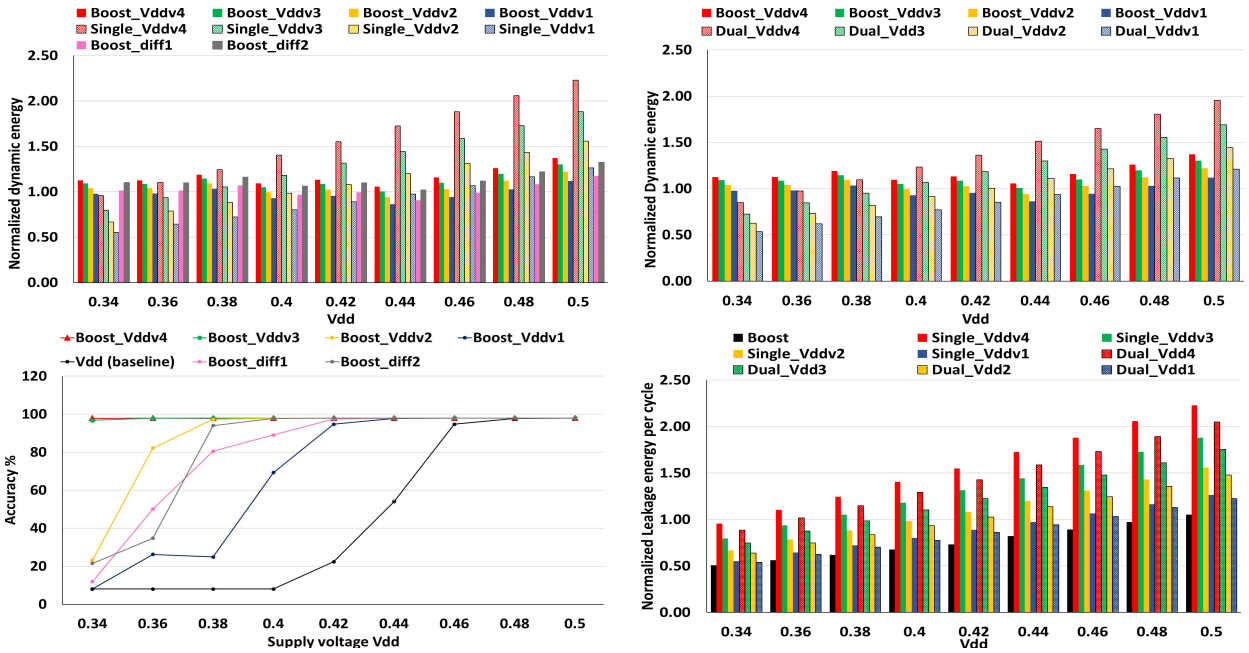

Figure 13: Analysis of fully connected DNNs (Clockwise from top left): (a) Dynamic energy comparison of boost vs single supply (b) Dynamic energy comparison of boost vs dual supply (c) Inference accuracy vs voltage (d) Leakage energy per cycle for boost, single and dual supply configurations. Boosting configurations refer to a chip supply Vdd (powering logic) boosted to a target voltage $Vddv$ for SRAM. Single supply refers to logic and SRAM at voltage $Vddv$. Dual supply refers to $V_h = Vddv$ to power memories and $V_l = Vdd$ to power logic, where V_l is obtained from the output of an LDO. All data points in the graph are normalized with respect to energy of the chip (logic + SRAMs) at 0.5V.

memory accesses to compute logic operations (*Ops_ratio*), and b) Ratio of energy of a single memory access to a single compute operation (*Energy_ratio*). The energy of a single compute operation was varied as a fraction of energy per access of an SRAM bank obtained from hardware measurements. We establish the benefits of our proposed SRAM-boosting technique by sweeping across these parameters.

Figure 12 shows the energy impact when these parameters are varied, for an SRAM that is boosted from $Vdd = 0.4V$ to $Vddv = 0.6V$ (lowering the memory bit error rate to almost 0). Results are normalized with respect to an LDO-based dual-supply configuration with the SRAM at 0.6V. In both cases, the logic voltage is maintained at 0.4V. We observed that boosting memories is more energy efficient for designs with lower ratio of memory-to-compute operations, and memory-to-compute energy, compared to a corresponding dual-Vdd configuration. Accelerator designs tend to be optimized for memory accesses, with accessed data re-used wherever possible keeping *Ops_ratio* small. Further, for designs with small banks of memory that can be individually boosted, the energy of a memory access is not significantly higher than that of a compute operation. Hence, for accelerators with realistic values of *Ops_ratio* and *Energy_ratio*, it is possible to achieve energy savings of up to 32% using programmable boosting.

6.2 Analysis of Fully-connected networks

In this subsection, we analyze the accuracy and energy costs of programmable voltage boosting in the proposed DNN accelerator. Table 2 shows the different boost configurations

Table 2: Voltage boost level for each layer in each configuration of the MNIST DNN workload. Inputs are boosted to the minimum level such that $Vddv_i > 0.44$ which ensures reliable operation in this case

Config	Weights-L1	Weights-L3	Weights-L3	Weights-L4
Boost_Vddv1	Vddv1	Vddv1	Vddv1	Vddv1
Boost_Vddv2	Vddv2	Vddv2	Vddv2	Vddv2
Boost_Vddv3	Vddv3	Vddv3	Vddv3	Vddv3
Boost_Vddv4	Vddv4	Vddv4	Vddv4	Vddv4
Boost_diff1	Vddv1	Vddv2	Vddv3	Vddv4
Boost_diff2	Vddv4	Vddv3	Vddv2	Vddv1

explored. *Boost_Vddv1* and *Boost_Vddv4* refer to weights in all layers boosted to the lowest ($Vddv1$) and highest ($Vddv4$) voltage levels possible using the programmable boost circuit. In *Boost_diff1*, differential voltage boosts were applied to weights of increasing layer depth, with the weights in the deepest layer (closest to the output) boosted to the highest level ($Vddv4$). In *Boost_diff2*, weights of successive layers were subjected to decreasing levels of boost, with the first layer boosted to the highest level. SRAM accesses involving inputs and intermediate computations are boosted to the minimum level that achieves a target voltage of at least 0.44V, based on the analysis in Figure 2.

Figure 13 compares the energy costs of a) boosting a supply voltage Vdd (logic supply) to a target voltage, $Vddv$ (SRAM supply) b) single voltage supply for logic and memories that is raised to same target voltage $Vddv$ c) dual supply voltage, with $Vddv$ for memories supplied as input voltage, and Vdd for logic obtained using a Low Drop Out (LDO) voltage regulator. All the data points in the graph are normal-

Table 3: Workload characteristics

Workload	Dataflow	Type	SRAMAcc/MAC Ops
MNIST	DANA [14]	4 Fully Connected Layers	75%
AlexNet for CIFAR-10 [16]	Eyeriss Row Stationary [17]	5 Conv layers	1.67%

ized with respect to energy of the chip (logic + SRAMs) at 0.5V. Figure 8 shows boosted voltages $Vddv1 - Vddv4$ values at each supply voltage Vdd. Each configuration has been modeled using the methodology described in Section 5. The following can be inferred from the graphs:

• **Boost vs single supply**- At a given supply, boosting to higher target voltage levels (*Boost_Vddv4* and *Boost_Vddv3*) result in higher energy savings over corresponding single supply configurations, than lower target levels (*Boost_Vddv2* and *Boost_Vddv1*). Most of the energy savings are obtained from the logic in the chip being able to operate at a lower voltage with only the memory supply boosted.

• **Boost vs dual supply**- Figure 13 shows a dynamic energy comparison between boosting and dual supply. In the dual supply configurations, the lower logic voltage Vdd is obtained using an LDO from a higher supply voltage $Vddv$. The trends in the graph show an overall energy savings when boosting is used as compared to dual-Vdd configurations. The LDO efficiency is higher when the difference between input and output voltages is smaller. Therefore, dual supply can only be advantageous in cases where the level of boost is low and the memory activity is very high.

• **Accuracy vs Energy**- A configurable boost circuit makes it possible to boost each supply voltage to the level where the target accuracy is just met. To achieve a target accuracy of 98%, it is necessary to expend the energy cost of *Boost_Vddv3* at 0.38V, whereas *Boost_Vddv1* is sufficient when operating at 0.46V. Thus fine-grain control of boosted voltage is necessary for a wider operating range. Application-controlled programmable boost results in configurations such as *Boost_diff1* and *Boost_diff2* where weight accesses of different layers are boosted to varying levels.

• **Leakage energy**- Since both the logic and the SRAMs operate at lower voltage Vdd when not reading or writing, boosting results in a substantial leakage energy savings. In the voltage range of 0.34 to 0.5V, boosting results in 32% leakage energy savings compared to dual supplies. Compared to the leakage energy per cycle at supply voltage Vdd, on average, the booster circuit results in only 6% overhead resulting in large leakage energy savings for the memory-dominated architecture shown in Table I.

6.3 Analysis of Convolutional Networks

In general, fully connected networks tend to be memory intensive since they offer little scope for data-reuse. Therefore, to fully exploit the benefits of the proposed boost architecture, we examine the AlexNet workload with 5 convolutional layers[9]. The *Eyeriss* [18] chip for AlexNet enables efficient data re-use using a hierarchical memory architecture. In this evaluation, we used the memory and PE activity for the Row Stationary (RS) dataflow, assuming the same global buffer size of 128KB as in [17]. Dynamic energy is modeled using equations 3 and 6 for the boosted and dual-Vdd configurations respectively, with activity obtained

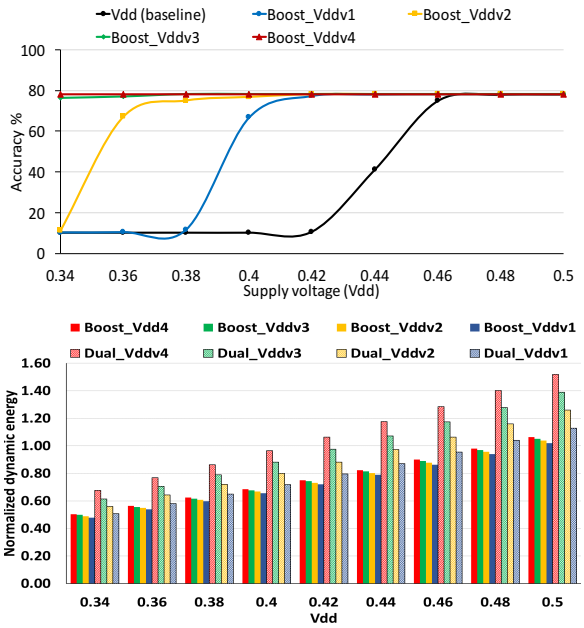

Figure 14: Accuracy (top) and dynamic energy (bottom) for processing 5 convolution layers of AlexNet using the Eyeriss row-stationary dataflow. The energy plot shows comparison of boost and dual supply configurations.

from [18]. The energy per access of the 128KB SRAM and the PEs is computed using the same methodology described in Section 5. The ratio of global buffer accesses to multiply-accumulate operations for the RS dataflow is very low, as shown in Table 3. Figure 14 shows the total dynamic energy of the AlexNet convolutional layers for the boost and dual-supply configurations. It can be seen that using a lower logic supply voltage Vdd and boosting the memory to $Vddv4$ is on average 26% more energy efficient as compared to using a higher voltage $Vddv4$ for the memories and lowering the logic supply to Vdd using LDOs. Boosted configurations across all 4 boost levels are 19% more energy efficient as compared to corresponding dual-supply configurations. The advantages of boosting are much higher as compared to the fully connected network because of lower memory activity and better reuse in the convolution layers.

• **Iso-accuracy comparison**: For a given target accuracy, our design enables us to opportunistically choose a boost configuration that increases Vdd so as to achieve the desired accuracy at the lowest energy overhead. Figure 15 shows a comparison of AlexNet dynamic energy between boost and dual supply techniques when the achieved target accuracy at each data point is the same. The required target accuracy (within 2% of the maximum) is reached when the supply voltage is 0.48V at a minimum. For each Vdd in the range of 0.34V to 0.46V, a different level of boost is necessary to achieve the same accuracy. The boosted voltage at each data point is shown. Compared to the dynamic energy at single supply of 0.48V, boosting results in 30% energy savings. Figure 15 also shows the dynamic energy of dual supply modes, where the logic voltage is in the range 0.34 to 0.46V, and chip input and memory voltage is assumed to be $Vdd + V_b$ at each data point. Overall, boosting results

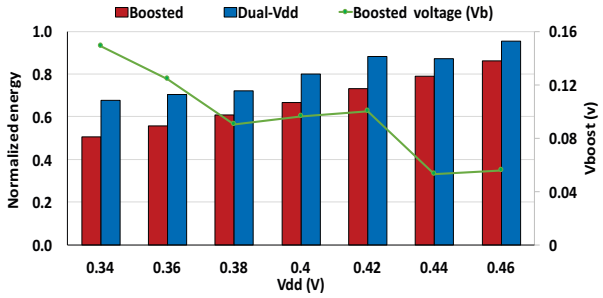

Figure 15: Comparison of dynamic energy for AlexNet when SRAM is boosted to achieve a target accuracy within 2% of peak. The chip reaches its target accuracy at $V_{dd} \geq 0.48V$ without need for boosting.

in 17% lower energy on average, across all voltage points, compared to dual supply operation.

With finer voltage adjustment (> 4 boost levels), one can obtain even greater energy savings with boosting. In addition, the proposed programmable boosting techniques can be combined with algorithmic techniques to obtain energy savings over and above that provided by software methods alone. For instance, *Deep Compression* [19] reduces the total size of AlexNet from 240MB to 6.9MB such that it can entirely fit in an on-chip SRAM. This makes our work indispensable to the application of Deep Compression at very low voltages.

7. RELATED WORK

7.1 DNN operation in the presence of errors

Several prior works have examined neural network operation in the presence of hardware faults. While [20] and [21] improve DNN model resilience by training them in the presence of faults, [22] also deploy in-situ canary circuits to determine SRAM failures at runtime. However, none of these techniques provide the fine-grained actuation allowing us to boost each bank separately at runtime. Further, the need for fault-aware training is also mitigated, since the nature of faults can be controlled by our boosting mechanisms. In [23], the authors examine an error propagation model for DNNs and its application to state-of-the-art DNN accelerators. However, the error mitigation techniques, such as selective latch hardening, come with high overhead and cannot adapt to runtime variations. In [24], the authors propose a fault injection framework for evaluating the resilience of DNNs and validate their model against real hardware measurements. However, the work does not consider voltage-dependent failures, or propose techniques to mitigate them.

7.2 Comparisons with cognitive accelerators

Several prior works such as [18, 25, 26, 27, 28, 29] showcase custom hardware designs for cognitive applications. However most are not geared for low voltage operation. In [28], the authors demonstrate a DNN accelerator prototype with techniques that enable low voltage operation by exploiting application-level resilience of the network. These techniques are complementary to those proposed in this paper and would reflect a similar improvement in V_{min} and power in our design as well. In [27], the authors demonstrate a cognitive IoT system with a CNN accelerator operating on harvested solar energy, while [29] propose a sub mW mixed-signal CNN accelerator. However, both designs operate at a fixed SRAM voltage, unlike our dynamic supply-boosting technique, which could achieve even more power-efficient operation. Works such as [30, 31, 32] demonstrate NVRAM-based analog neural accelerators. However, unlike SRAM-based designs, they are still at nascent stages of commercial adoption, due to NVRAM-based technologies facing limitations in latency, system integration and foundry availability.

7.3 Low Vmin SRAM design

8T and 10T bitcells are used to improve writeability and read stability of the SRAM cell [33, 34]. Some demonstrate 100mV read V_{min} reduction [35, 36] and others show functionality down to 0.35V [37] and 0.2V [38]. However, significant area overheads, (20% for 8T vs 6T [34] and 61% for 10T vs 8T SRAMs [39]), and increased complexity of characterization and manufacturing make our 6T-based boosting solution far more viable for widespread commercial adoption. [40] proposed a new SRAM design for sub-threshold operation. However, this requires a redesign of the SRAM array which comes at a very high cost in terms of layout and characterization unlike our work which is a drop-in addition to a standard SRAM. In works such as [37, 38, 40], the Vmin reduction translates to a constant penalty in terms of area and performance, including when operating at nominal or high voltage, where variability mitigation is not needed. On the contrary, our work targets a wide operating range with programmable levels of boost.

Supply voltage boosting for efficient SRAM operation was proposed in [41, 42], and was implemented in hardware by Joshi *et al* [7, 8, 43] on a stand-alone 8T SRAM array as as proof-of-concept. Boosting can be obtained either through parasitic coupling [8, 43], or enhanced with resonant coupling by adding an inductor [7]. However, these methods are limited by the limited amount of boost possible (10-15%), significant area/power costs, and complexity overheads of a resonant inductor, and have not been implemented in 6T SRAM. Further, none of these works consider an end-to-end system with application-aware voltage control.

8. CONCLUSION

In this paper, we presented an application-aware voltage supply-boosted SRAM architecture to improve reliable operation of accelerators at very low voltages. We demonstrated a DNN accelerator chip with boosted SRAMs taped out in 14nm technology. Finally, we demonstrate dynamic energy savings of upto 26% for AlexNet and savings of 32% in leakage power, when our design is compared to conventional voltage scaling schemes.

Acknowledgment

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

9. REFERENCES

[1] M. Qazi, M. Sinangil, and A. Chandrakasan, "Challenges and Directions for Low-Voltage SRAM," *IEEE Design Test of Computers*, vol. 28, Jan 2011.

[2] N. Kurd *et al.*, "5.9 haswell: A family of IA 22nm processors," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2014.

[3] D. Bol *et al.*, "Sleepwalker: A 25-MHz 0.4-V sub-mm^{2}7-μW/MHzmicrocontroller in 65-nm LP/GP CMOS for low-carbon wireless sensor nodes," *IEEE Journal of Solid-State Circuits*, vol. 48, Jan 2013.

[4] L. Chang *et al.*, "A fully-integrated switched-capacitor 2:1 voltage converter with regulation capability and 90% efficiency at 2.3a/mm2," in *2010 Symposium on VLSI Circuits*, June 2010.

[5] B. Zimmer *et al.*, "A RISC-V vector processor with simultaneous-switching switched-capacitor DC-DC converters in 28 nm FDSOI," *IEEE Journal of Solid-State Circuits*, vol. 51, April 2016.

[6] L. G. Salem and P. P. Mercier, "A sub-1.55mv-accuracy 36.9ps-fom digital-low-dropout regulator employing switched-capacitor resistance," in *IEEE International Solid - State Circuits Conference (ISSCC)*, 2018.

[7] R. V. Joshi, M. M. Ziegler, and H. Wetter, "A low voltage SRAM using resonant supply boosting," *Journal of Solid-State Circuits (JSSC)*, vol. 52, 2017.

[8] R. V. Joshi and M. M. Ziegler, "Programmable supply boosting techniques for near threshold and wide operating voltage SRAM," in *Custom Integrated Circuits Conference*, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012.

[10] "Tensorflow," www.tensorflow.org.

[11] B. Reagen *et al.*, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," in *International Symposium on Computer Architecture*, ser. ISCA, 2016.

[12] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/.

[13] "Cadence Spectre Circuit Simulator," www.cadence.com.

[14] S. Eldridge, A. Waterland, M. Seltzer, J. Appavoo, and A. Joshi, "Towards general-purpose neural network computing," in *International Conference on Parallel Architecture and Compilation (PACT)*, 2015.

[15] "Cadence Joules RTL Power Solution," www.cadence.com.

[16] "CIFAR-10 and CIFAR-100 datasets," www.cs.toronto.edu/~kriz/cifar.html.

[17] Y. H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *International Symposium on Computer Architecture (ISCA)*, 2016.

[18] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *International Solid-State Circuits Conference (ISSCC)*, 2016.

[19] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding," *International Conference on Learning Representations (ICLR)*, 2016.

[20] L. Yang and B. Murmann, "SRAM voltage scaling for energy-efficient convolutional neural networks," in *2017 18th International Symposium on Quality Electronic Design (ISQED)*, March 2017.

[21] J. Deng *et al.*, "Retraining-based timing error mitigation for hardware neural networks," in *Design, Automation and Test in Europe Conference*, ser. DATE, 2015.

[22] S. Kim *et al.*, "MATIC: Learning around errors for efficient low-voltage neural network accelerators." in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2017.

[23] G. Li *et al.*, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017.

[24] B. Reagen *et al.*, "Ares: A framework for quantifying the resilience of deep neural networks," in *Design Automation Conference (DAC)*, 2018.

[25] T. Chen *et al.*, "DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning," in *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014.

[26] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "EIE: efficient inference engine on compressed deep neural network," in *International Symposium on Computer Architecture (ISCA)*, 2016.

[27] T. Karnik *et al.*, "A cm-scale self-powered intelligent and secure IoT edge mote featuring an ultra-low-power SoC in 14nm tri-gate CMOS," in *International Solid-State Circuits Conference, ISSCC*, 2018.

[28] P. N. Whatmough *et al.*, "4.3 A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with > 0.1 timing error rate tolerance for IoT applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017.

[29] D. Bankman *et al.*, "An always-on 3.8 uJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28nm CMOS," in *International Solid - State Circuits Conference - (ISSCC)*, 2018.

[30] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *International Symposium on Computer Architecture (ISCA)*, 2016.

[31] M. N. Bojnordi and E. Ipek, "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," in *International Symposium on High Performance Computer Architecture (HPCA)*, 2016.

[32] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Frontiers in Neuroscience*, 2016.

[33] W. H. Henkels, W. Hwang, and R. V. Joshi, "A 500 mhz 32-word x 64-bit 8-port self-resetting cmos register file and associated dynamic-to-static latch," in *Symposium on VLSI Circuits*, June 1997.

[34] L. Chang *et al.*, "An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches," *IEEE Journal of Solid-State Circuits*, vol. 43, April 2008.

[35] B. Zimmer, S. O. Toh, H. Vo, Y. Lee, O. Thomas, K. Asanović, and B. Nikolić, "SRAM assist techniques for operation in a wide voltage range in 28-nm CMOS," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, Dec 2012.

[36] G. Shamanna *et al.*, "Using ECC and redundancy to minimize vmin induced yield loss in 6T SRAM arrays," in *2012 IEEE International Conference on IC Design Technology*, May 2012.

[37] F. Abouzeid *et al.*, "Scalable 0.35 V to 1.2 V SRAM bitcell design from 65 nm CMOS to 28 nm FDSOI," *IEEE Journal of Solid-State Circuits*, vol. 49, July 2014.

[38] T. H. Kim, J. Liu, J. Keane, and C. H. Kim, "A 0.2 V, 480 kb subthreshold SRAM with 1 k cells per bitline for ultra-low-voltage computing," *IEEE Journal of Solid-State Circuits*, vol. 43, Feb 2008.

[39] I. J. Chang, J. J. Kim, S. P. Park, and K. Roy, "A 32 kb 10T sub-threshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, Feb 2009.

[40] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200 mv 6-t subthreshold sram," *IEEE Journal of Solid-State Circuits*, vol. 43, Oct 2008.

[41] K. Itoh, A. R. Fridi, A. Bellaouar, and M. I. Elmasry, "A deep sub-V, single power-supply SRAM cell with multi-V/sub T/, boosted storage node and dynamic load," in *Symposium on VLSI Circuits*, 1996.

[42] M. Iijima, K. Seto, and M. Numa, "Low power SRAM with boost driver generating pulsed word line voltage for sub-1V operation," *Journal of Computers (JCP)*, 2008.

[43] R. V. Joshi *et al.*, "14nm FinFET based supply voltage boosting techniques for extreme low Vmin operation," in *Symposium on VLSI Circuits (VLSI Circuits)*, 2015.