Journal of Systems Architecture 157 (2024) 103292

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Shift-and-Safe: Addressing permanent faults in aggressively undervolted CNN accelerators

Yamilka Toca-Díaz, Rubén Gran Tejero, Alejandro Valero *

Department of Computer Science and Systems Engineering, Aragon Institute for Engineering Research - I3A, Universidad de Zaragoza, Zaragoza, Spain

ARTICLE INFO

Keywords:
CNN accuracy
Deep learning
Energy efficiency
Permanent faults
Ultra-low voltage

ABSTRACT

Underscaling the supply voltage (V_{dd}) to ultra-low levels below the safe-operation threshold voltage (V_{min}) holds promise for substantial power savings in digital CMOS circuits. However, these benefits come with pronounced challenges due to the heightened risk of bitcell permanent faults stemming from process variations in current technology node sizes.

This work delves into the repercussions of such faults on the accuracy of a 16-bit fixed-point Convolutional Neural Network (CNN) inference accelerator powering on-chip activation memories at ultra-low V_{dd} voltages. Through an in-depth examination of fault patterns, memory usage, and statistical analysis of activation values, this paper introduces Shift-and-Safe: two novel and cost-effective microarchitectural techniques exploiting the presence of outlier activation values and the underutilization of activation memories. Particularly, activation outliers enable a shift-based data representation that reduces the impact of faults on the activation values, whereas the memory underutilization is exploited to maintain a safe replica of affected activations in idle memory regions. Remarkably, these mechanisms do not add any burden to the programmer and are independent of application characteristics, rendering them easily deployable across real-world CNN accelerators.

Experimental results show that Shift-and-Safe maintains the CNN accuracy even in the presence of almost a quarter of the total activations with faults. In addition, average energy savings are by 5% and 11% compared to the state-of-the-art approach and a conventional accelerator supplied at V_{min}, respectively.

1. Introduction

Artificial Intelligence (AI) applications usually rely on specialized hardware accelerators to expedite their execution. Efforts to address power consumption in such accelerators are crucial for advancing energy efficiency, fostering environmental sustainability, and promoting responsible AI deployment. In this context, modern computing systems often grapple with compromised energy efficiency due to conservative operation guardbands motivated by variations in the manufacturing process of current CMOS technology nodes. For instance, the transistor's supply voltage (V_{dd}) is often set conservatively above the safe voltage limit (V_{min}) imposed by the worst-case transistor to mitigate the risk of sudden V_{dd} droops. However, these droops are infrequent events [1], leading to energy wasting with supply voltage overscaling, as energy consumption quadratically increases with V_{dd}.

In the realm of AI accelerators, particularly those employed in the inference of Convolutional Neural Networks (CNNs), the integration of large on-chip memories poses significant energy challenges. These memory structures typically employ 6-transistor SRAM bitcells, which are susceptible to process variations. To mitigate energy consumption, traditional techniques like Dynamic Voltage Scaling (DVS) have been widely adopted, involving the reduction of the voltage guardband by pushing V_{dd} toward V_{min} while maintaining a fixed frequency [1]. To achieve further energy savings, V_{dd} can be aggressively underscaled below V_{min}. However, this results in a potential risk due to the high occurrence of permanent faults in vulnerable bitcells, requiring advanced but energy-hungry Error-Correcting Codes (ECC) to ensure a reliable operation [2–4].

Prior work focusing on permanent faults as a consequence of supplying CNN accelerators at V_{dd} below V_{min} include dynamic adjustments of V_{dd} for individual neural network layers at runtime according to reliability demands [5], FPGA compilation process enhancements to bypass faulty cells [6], custom retraining of networks under faults [7,8], or weight transfer and reallocation techniques to bypass faulty processing elements and use reliable ones [9]. Unfortunately, these approaches depend on the programmer or costly application profiling efforts to adapt the mechanism.

Flip-and-Patch is a recent technique that does not rely on the programmer or profiling efforts [10]. This approach exploits the

* Corresponding author.
E-mail address: alvabre@unizar.es (A. Valero).

https://doi.org/10.1016/j.sysarc.2024.103292
Received 3 August 2024; Received in revised form 1 October 2024; Accepted 27 October 2024
Available online 5 November 2024

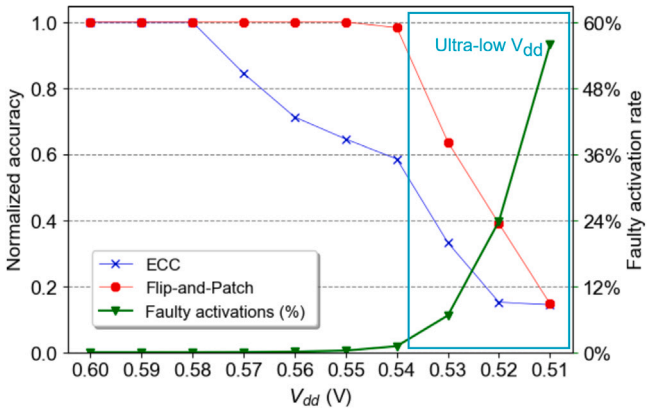

Fig. 1. Normalized accuracy of ECC and state-of-the-art Flip-and-Patch technique for different supply voltages with respect to the golden (fault-free) accuracy on the left Y-axis. The right Y-axis refers to the percentage of faulty activations in the on-chip memory of a CNN accelerator (green line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

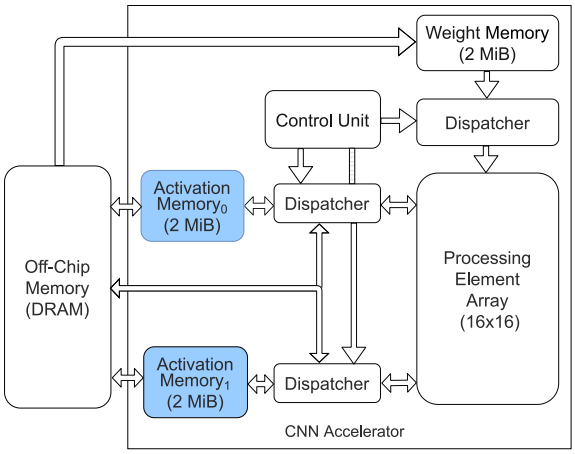

Fig. 2. Overview of the baseline CNN accelerator.

observation that CNN accuracy is highly sensitive to the memory bit position of faults. To minimize the impact of faults, this technique flips the representation of activations with faults located in most significant bit positions, situating faults in least significant bits. In addition, it employs a dedicated set-associative *patching* cache supplied above V_{min} to store a reliable replica of activations with faults in both most and least significant bits.

Fig. 1 illustrates the averaged top-1 accuracy of Single-Error Correction Double-Error Detection (SECDED) ECC at a granularity of 16-bit words and Flip-and-Patch for a number of widely used CNN applications as the V_{dd} of on-chip activation memories of an accelerator reduces below V_{min} (0.6 V).[1] See Section 3 for further details about the experimental environment. Results are normalized to the golden (fault-free) accuracy obtained at a reliable operation mode with V_{dd} = 0.6 V. In addition, the right Y-axis shows the percentage of faulty activations over the entire on-chip memory of the CNN accelerator. ECC does not hold the accuracy as soon as V_{dd} scales down to 0.57 V, leading to nearly 20% accuracy degradation. On the other hand, Flip-and-Patch maintains the golden accuracy within a faulty activation rate of 1% corresponding to V_{dd} = 0.54 V.

Entering in the range that we define as *ultra-low* voltages, the higher number of faulty activations (i.e., 6×, 21×, and 50× for 0.53 V, 0.52 V, and 0.51 V, respectively, compared to the number of faulty activations at 0.54 V) severely compromises the accuracy of both techniques, mostly performing a random guessing at 0.51 V.

This paper builds on the state-of-the-art Flip-and-Patch approach with the aim to address the impact on CNN accuracy caused by the huge number of faults appearing at ultra-low V_{dd}. To do so, the four main contributions of this work are listed as follows:

- We expose the vulnerabilities of Flip-and-Patch operating at ultra-low V_{dd} and perform an in-depth characterization study, identifying new opportunities to improve the fault tolerance of on-chip memories in CNN accelerators.
- Based on the large value range of CNN activation parameters and the presence of outliers, we propose a shift-based data representation that minimizes the value deviations caused by faults.
- Based on the underutilization of activation memories, we propose to maintain safe replica values in unused memory regions.
- We devise a new cost-effective microarchitectural design, namely Shift-and-Safe, supporting the shift-based representation and the management of safe replicas. The proposed design does not add any burden to the programmer neither relies on application profiling efforts.

Experimental results show that Shift-and-Safe ensures the golden accuracy at ultra-low V_{dd} while reducing the average energy consumption by 5% compared to Flip-and-Patch. Energy savings scale up to 11% and 40% with respect to a conventional accelerator supplied at V_{min} and nominal voltages, respectively.

The rest of this paper is organized as follows. Section 2 provides a background for this work. Section 3 presents a characterization study of CNN applications that enables our proposal. Section 4 introduces the proposed design approach. Section 5 evaluates experimental results. Section 6 discusses related work, and finally, Section 7 summarizes this work.

2. Background

This section surveys the CNN accelerator architecture and the framework employed to obtain the reliability models for this work. Then, the state-of-the-art Flip-and-Patch technique is described.

2.1. Baseline CNN accelerator architecture

Our modeled baseline accelerator for the inference of CNNs is based on state-of-the-art accelerators from both the industry [12] and the academia [13]. Fig. 2 shows the main components consisting of a 16 × 16 Processing Element (PE) array, on-chip memory storage, dispatchers for every memory, and a control unit. On-chip storage includes a couple of 2 MiB activation memories and a 2 MiB weight memory. These components are sized according to the domain of embedded systems [14], although our proposal could be easily adapted to larger accelerators.

The PE array forms a systolic array processor with PEs interconnected through a 2D mesh. Each PE computes 16-bit fixed-point dot-products through partial sums with an activation and a weight. The dataflow in the PE array follows the output stationary approach [15]. This array incorporates intermediate memory buffers to temporarily store and sequentially arrange output activations before forwarding them to the dispatchers.

Like the EIE accelerator [16], activation memories swap input and output roles after the computation of every network layer. That is, a given activation memory stores even layers and the counterpart memory stores odd layers. These memories are implemented as scratchpad

[1] This paper focuses on activations instead of weight parameters since prior work has shown that the latter are inherently more resilient to faults in CNN accelerators using fixed-point data-types [10,11].

memories, each one including a single read/write port and consisting of eight 256 KiB banks. For simplicity, every layer is stored from address 0×0 onwards and activations are sequentially arranged in memory, occupying one bank after another. For example, a 200 KiB layer occupies a single bank, whereas a 400 KiB layer occupies two banks. Layers exceeding the capacity of activation memories (2 MiB) are spilled to off-chip memory (see Section 3.1).

Similarly to previous CNN accelerator models [17,18], network parameters occupy 16 bits and are represented in fixed-point arithmetic, adjusting the number of integer and fractional bits to the requirements of each CNN application at run time (see Section 3.1). Finally, activations are sequentially retrieved from the on-chip memories when an input layer is read, providing 16 consecutive activations (32 bytes) to the dispatchers per memory access. Dispatchers are driven by the control unit, which exploits control information of the current layer to properly feed the PE array.

2.2. Reliability models

Our reliability models are extracted from the MoRS fault modeling framework [19]. This framework generates reliability models for permanent faults using publicly available undervolted fault map data from a real hardware platform. In particular, the platform corresponds to a VC707 Xilinx FPGA and fault maps are provided for supply voltage values from 0.6 V (V_{min}) to 0.54 V [6]. Setting V_{dd} below 0.54 V is not possible since the platform stops operating. To overcome this limitation, MoRS has been configured to obtain reliability models for ultra-low V_{dd} values below 0.54 V according to the fault maps of the real platform.

Note that this work focuses on permanent faults as a consequence of underscaling V_{dd} below V_{min}. These faults manifest during the entire period of time in which $V_{dd} < V_{min}$ and are detected during post-fabrication testing before deploying the device in the field [20,21]. Dealing with unpredictable faults appearing at specific execution cycles as a consequence of particle strikes, voltage noise, or aging effects are out of the scope of this work.

Finally, like previous academic work [22] and commercial devices [23], our baseline CNN accelerator has dedicated voltage domains for logic and memory arrays, which allows aggressive voltage underscaling in activation memories while maintaining the remaining hardware components at $V_{dd} \geq V_{min}$ to avoid faults.

2.3. State-of-the-art: Flip-and-Patch approach

Flip-and-Patch exploits the fact that the bit position of a fault in activations greatly impacts the accuracy of CNNs [10]. Particularly, this technique classifies 16-bit faulty activations as follows:

- Low-order (L) activation memory words with faults in the least significant byte.
- High-order (H) activation memory words with faults in the most significant byte.
- Low- & High-order (L&H) activation memory words with faults in both least and most significant bytes.

Authors make the observation that only H and L&H activations compromise the CNN accuracy. Consequently, in a write operation to the activation memory, H activations turn into L activations with a flip operation (i.e., logic values exchange specific bit positions 15 and 0, 14 and 1, 13 and 2, et cetera), whereas reliable replicas of all the L&H activations are stored in an additional 2.5 KiB 5-way set-associative patching cache powered at V_{dd} above V_{min}. In a read operation, H activations are flipped back and L&H activations are retrieved from the patching cache. Refer to [10] for further details.

Flip-and-Patch falls short when the faulty activation rate is greater than 1% (see Fig. 1). This is due to two main reasons: (i) the patching cache is not large enough to store the increasing number of L&H activations, and (ii) even after flipping, the greater volume of L and flipped H activations hurts the CNN accuracy (see Section 3.2).

Table 1
Main characteristics of the studied CNN benchmarks.

Benchmark	Accuracy	Largest layer size (in MiB)	Activation quantization
AlexNet [25]	0.89	0.55	$Q_{4,4}$
DenseNet [26]	0.92	1.53	$Q_{3,5}$
MobileNet [27]	0.88	1.55	$Q_{4,9}$
SqueezeNet [28]	0.93	0.76	$Q_{6,4}$
VGG16 [29]	0.81	1.56	$Q_{3,8}$
ZFNet [30]	0.83	0.53	$Q_{4,6}$

Table 2
Percentage of faulty activations for different V_{dd} values.

Type of activation	0.54 V	0.53 V	0.52 V	0.51 V
L	0.58%	3.5%	12.21%	26.1%
H	0.51%	3.13%	10.8%	22.45%
L&H	0.02%	0.24%	0.78%	7.35%
Total	1.11%	6.87%	23.79%	55.9%

3. Characterization study

This section introduces the CNN benchmarks and reliability models employed in this work. Then, the vulnerabilities of Flip-and-Patch at ultra-low V_{dd} are exposed. Finally, the section discusses the observations that enable our proposed mechanisms.

3.1. CNN benchmarks and reliability models

We have chosen six widely used CNN benchmarks with different accuracy, memory demands, and data representation. All the CNNs run a colorectal cancer histology dataset for image classification purposes [24]. All the presented results are averaged for the inference of the entire dataset consisting of 750 test images. More details about the experimental environment can be found in Section 5.1.

Table 1 shows the main characteristics of the studied CNNs. Accuracy ranges from 0.81 (VGG16) to 0.93 (SqueezeNet). In terms of storage requirements as largest layer size, results vary from 0.53 MiB (ZFNet) to 1.56 MiB (VGG16).[2] Such an underutilization of the activation memory provides an opportunity to exploit the self on-chip scratchpad memory of the accelerator to store reliable replicas of faulty activations with minimal overhead.

The rightmost column of the table shows the established quantization (Q) for activations represented with fixed-point data-types, distinguishing between integer bits (left number) and fractional bits (right number), avoiding accuracy losses compared to the top-1 accuracy with 32-bit floating-point (IEEE-754) representation. Benchmarks require between 8 and 13 bits, plus an additional bit for the sign. Since our baseline accelerator assumes 16-bit words, fractional bits are extended to cover up to 16 bits.

Table 2 shows the percentage of faulty activations in the on-chip memory of the accelerator according to the classification of faulty activations discussed in Section 2.3. Results are averaged for ten different fault maps per V_{dd} value. As observed, the total number of faulty activations exponentially grows with V_{dd} underscaling. The number of L and H activations is quite similar for a given V_{dd} value, whereas much less L&H activations with faults in both bytes can be appreciated. However, under the faultiest voltage level, the number of L&H activations is by 7.35%.

In fact, at ultra-low V_{dd} values, L&H activations sum up to 5 KiB, 16 KiB, and 151 KiB for 0.53 V, 0.52 V, and 0.51 V, respectively. These

[2] Results exclude those spilled layers to off-chip memory exceeding the 2 MiB capacity of the activation memory (see Section 2.1). This is the case of a single layer for SqueezeNet and ZFNet, as well as four layers for VGG16.

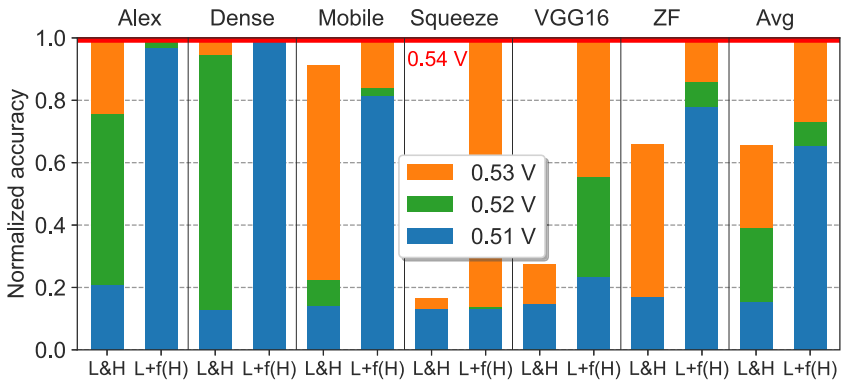

Fig. 3. Normalized accuracy of Flip-and-Patch for different ultra-low V_{dd} values with respect to the golden (fault-free) accuracy. Results distinguish between accuracy drops caused by L&H activations and L activations plus flipped H activations (*L+f(H)*). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Percentage of activations with specific number of logic '0' values in most significant bits excluding the sign (s) bit.

Benchmark	$s0*$	$s00*$	$s000*$
AlexNet	99.9788%	99.6193%	97.1099%
DenseNet	99.9999%	99.9996%	99.8060%
MobileNet	99.9999%	99.8647%	98.1782%
SqueezeNet	99.9973%	99.9812%	99.8888%
VGG16	99.9999%	99.9998%	99.9996%
ZFNet	99.9998%	99.9116%	98.8383%
Average	99.9959%	99.8960%	98.9701%

memory requirements largely exceed the size of the additional fault-free storage of Flip-and-Patch (i.e., 2.5 KiB). Moreover, for the two latter V_{dd} values, implementing such a dedicated patching storage in the accelerator could be prohibitive in terms of energy and area.

3.2. Limitations of Flip-and-Patch

Fig. 3 depicts the impact on CNN accuracy obtained by Flip-and-Patch as stacked bars when the number of faulty activations increases with V_{dd} reductions below 0.54 V. Faulty activations are classified as L&H activations that cannot be accommodated in the patching cache and L activations. The latter also include H activations turned into L activations after flipping them (label *L+f(H)*). For a given type of faulty activation, the counterpart is removed to isolate the impact on accuracy.

As expected, at V_{dd} = 0.54 V, the golden accuracy is preserved (i.e., horizontal red line at 1.0). However, the excess of L&H activations severely hurts the accuracy as V_{dd} underscales. This effect is highly remarkable in SqueezeNet, VGG16, and ZFNet, where the accuracy degradation surpasses 80%, 75%, and 35%, respectively, for every V_{dd} level. The impact of *L+f(H)* activations is generally less significant. However, applications like SqueezeNet and VGG16 show large accuracy losses below 0.53 V. On average, L&H activations degrade accuracy by 34%, 61%, and 85% for 0.53 V, 0.52 V, 0.51 V, respectively, whereas these percentages are by 1%, 27%, and 35% for *L+f(H)* activations.

3.3. New opportunities

This section describes the two main findings that underpin the foundation of our proposed Shift-and-Safe mechanism.

3.3.1. Shift opportunity

Flipping the representation of an activation reduces the deviation of the affected value, contributing to minimize the impact on CNN accuracy. However, as shown above, at ultra-low V_{dd}, flipping H activations

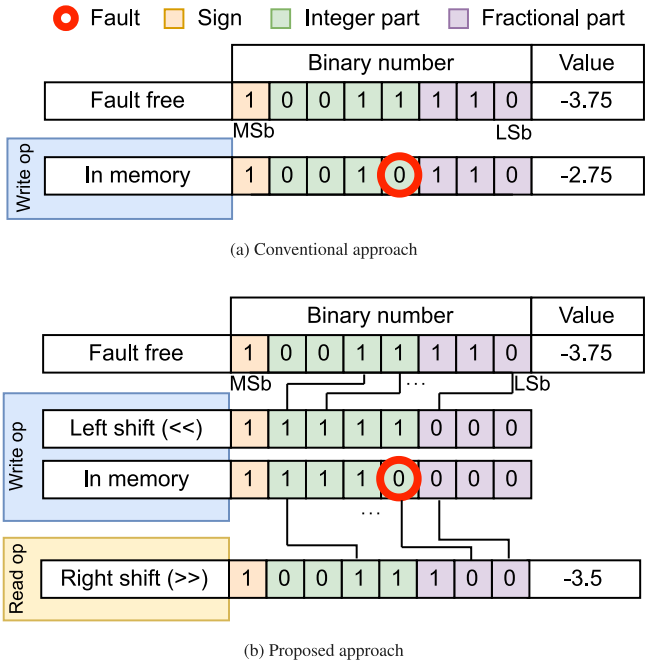

Fig. 4. Working examples of the conventional and proposed approaches considering an 8-bit L activation with a fault in the least significant half-byte. Labels *MSb* and *LSb* refer to the most and least significant bits, respectively.

is insufficient. Next, we analyze the activation values with the aim to further reduce the magnitude deviations caused by faults.

Table 3 classifies all the activations, represented as 16-bit fixed-point values, depending on the number of logic '0' in the three most significant bits, excluding the sign (s) bit (bit position 15). The pattern $s0*$ refers to activations with a logic '0' in bit 14 and the remaining bit positions (including the s bit) being either '0' or '1'. The pattern $s00*$ identifies activations with at least logic '0' in bits 14 and 13, and so on. Note that $s000*$ activations are a subset of $s00*$ activations, and in turn, the latter are a subset of $s0*$ activations. As observed, the presence of $s1*$ outliers prevents $s0*$ activations to be 100%. Moreover, $s01*$ activations are also outliers since they represent at most 0.3595% (99.9788 – 99.6193) in AlexNet.[3] On the other hand, the percentage of $s001*$ activations is more than 1% in ZFNet, and almost 2% and 3%

[3] The presence of outlier activations is essential to obtain the original accuracy of the networks.

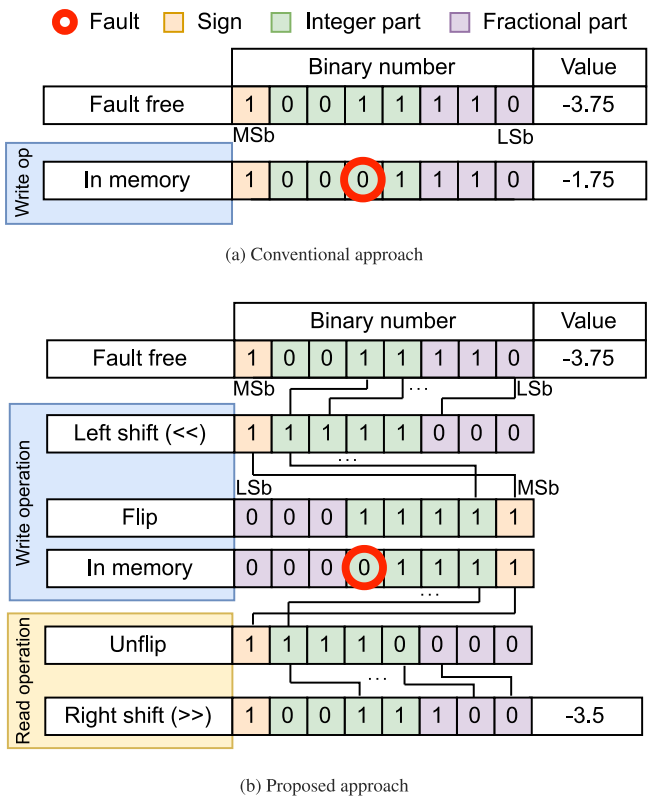

Fig. 5. Working examples of the conventional and proposed approaches considering an 8-bit H activation with a fault in the most significant half-byte. Labels *MSb* and *LSb* refer to the most and least significant bits, respectively.

in MobileNet and AlexNet, respectively. On average, $s0*$, $s00*$, and $s000*$ activations represent, respectively, 99.9959%, 99.8960%, and 98.9701% of the total activations.

Such value distributions present an opportunity to mitigate the influence of faulty bits in L activations and flipped H activations (*L+f(H)*). Fig. 4 illustrates with working examples how activation values are transformed in the case of an L activation with a stuck-at '0' fault in the fifth bit from left to right. For illustration purposes, the bit width is limited to 8. A conventional write operation (Fig. 4(a)) transforms the original fault-free value -3.75 into -2.75, resulting in a deviation of 1 unit. On the other hand, a write operation enhanced with the proposed approach (Fig. 4(b)) applies a 2-bit left shift preserving the s bit. As outlined in Table 3, the two dropped bits are likely to be '0'. Subsequently, when retrieving this value from the activation memory, the read operation reverses the shift by two bits to the right, padding the two leftmost magnitude bits with '0', resulting in the value -3.5 and just a deviation of 0.25 with respect to the fault-free value.

Fig. 5 depicts an example involving an H activation with a stuck-at '0' fault in the fourth bit. In this case, the fault-free value suffers a deviation of 2 units under a conventional write operation, resulting in the value -1.75. In contrast, the proposed technique applies a left-shift operation and then flips the value (*f(H)*) before storing the activation in memory. In a subsequent read operation, the value is *unflipped* and then shifted back to the right. As a result, the retrieved value is -3.5 with just a deviation of 0.25 compared to the fault-free value.

Consequently, bit values of the affected activations remain stored in memory cells that are two bits more significant with respect to a conventional write operation. More precisely, faulty bitcells of the target *L+f(H)* activation word affect less significant (two bit positions) activation bits, reducing the impact of these faults on the resulting magnitude. The sign bit is exempt from shifts to maintain the integrity of the non-negligible number of negative *L+f(H)* activations: switching the sign bit from '1' (negative) to '0' (positive) as a consequence of a shift doubles the magnitude of the value.[4]

The optimal shift operation in number of bits has been experimentally determined. Aggressive shifts involving more than two bits imply larger value losses, whereas a conservative 1-bit shift does not minimize sufficiently the impact of faults on accuracy. Particularly, experimental results show that 1-bit and 3-bit shifts degrade the average CNN accuracy by 0.6% and 2.7%, respectively, compared to 2-bit shifts. Notice too that performing fixed 2-bit shifts to all the affected activations largely simplifies the design of the shifter.

Finally, compared to the 16-bit fixed-point data representation used in this work, alternative representations with shorter bitwidths could significantly reduce the percentage of activations with logic '0' values in most significant bits and compromise the CNN accuracy achieved by the proposed Shift technique. In this sense, under an 8-bit fixed-point representation, the average percentage of $s0*$, $s00*$, and $s000*$ activations is by 99.5178%, 96.7214%, and 96.2392%, respectively, resulting in an average accuracy degradation of 3.2% using the Shift mechanism. Dealing with even shorter bitwidths could require the necessity to revisit the proposed design or explore alternative solutions, which is left for future work.

3.3.2. Safe-bank opportunity

The Shift approach deals with L and flipped H activations. However, it results ineffective for L&H activations with faults in both bytes. The Safe-Bank approach deals with these activations. This technique is based on two main observations. First, the majority of activation layers are smaller than the size of a typical activation memory. As described in Section 3.1, most layers demand only a fraction of the available activation memory (2 MiB). This surplus storage at the end of the memory addressing space presents an opportunity to patch L&H activations in such idle memory locations. Since the faultiest operation mode requires at most 151 KiB to patch all the potential L&H activations (see Section 3.1), a single 256 KiB bank of the activation memory powered at 0.6 V to avoid faults would largely accommodate all these activations.

The second observation relies on the sequential memory access pattern exhibited by activations (see Section 2.1). Since the order in which activations are consumed is the same in which they are produced, the safe bank is managed as a FIFO queue. When the activation memory acts as output buffer, a pointer keeps track of the next entry to be used in the safe bank. Every new L&H activation to be written is safely stored in that entry and the pointer advances to the subsequent entry. On the contrary, when the activation memory acts as input buffer, the pointer indicates the entry to be read to restore the following L&H activation. The pointer resets every time the activation memory changes its input/output role.

4. Proposed design: Shift-and-Safe

This section introduces the proposed design to support shift operations and exploit the underutilization of activation memories. Estimations of power, energy, area, and timing overhead of the proposed circuit are also discussed.

4.1. Shift technique

Fig. 6 illustrates the required components of our proposed Shift-and-Safe technique in the read port of an activation memory. A read operation requires to undone the transformations in the data representation of stored activations before forwarding them to the dispatcher.

[4] The sign bit is not covered by the proposed approach since the number of faults affecting this bit is rather low and we have experimentally evaluated that their impact on CNN accuracy is insignificant.

Table 4
Power and energy of DVS at 0.6 V (V_{min}), Flip-and-Patch (FaP) at 0.54 V, and a baseline (Base) and Shift-and-Safe (SaS) approaches at ultra-low V_{dd}.

	0.6 V	0.54 V	0.53 V		0.52 V		0.51 V	
	DVS	FaP	Base	SaS	Base	SaS	Base	SaS
Leakage power (mW)	315	296.3	263.8	290	255.9	276.4	247.9	263.6
Dynamic read energy (pJ)	83.8	71.3	62.8	69.5	59.5	66.2	56.3	63
Dynamic write energy (pJ)	66.4	53.7	46.5	52.1	43.4	49.1	40.5	46.1

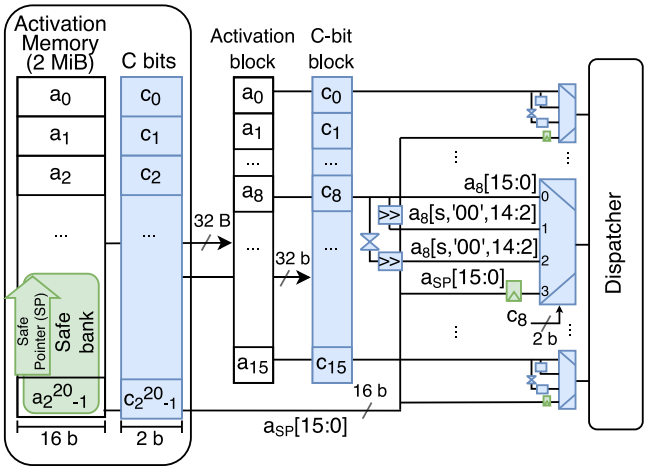

Fig. 6. Proposed technique in the read port of an activation memory. Required components of Shift and Safe-Bank approaches are highlighted in blue and green, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In this sense, remember that, when a layer is written, L activations are left-shifted, H activations are left-shifted and then flipped, and L&H activations are sequentially stored in the safe bank.

Every stored activation incorporates two control (C) bits to codify the different types of activations. Particularly, $C = 00$ identifies reliable activations that do not require any change in the representation before forwarding them to the dispatcher. $C = 01$ refers to L activations that need to be shifted back. $C = 10$ classifies H activations that necessitate to be flipped back and then shifted back. Finally, $C = 11$ determines L&H activations that have to be obtained from the safe bank.

Note that C bits are set during post-fabrication testing prior to deploying the accelerator, and they are independent of the applications to be run. This process is often employed in practice by traditional error detection/correction techniques [21]. Of course, C bits are supplied at 0.6 V to avoid faults.

After a 32-byte read operation of 16 consecutive activations (e.g., the initial 16 activations in Fig. 6) and the corresponding 32 C bits, 16 4-to-1 multiplexers driven by the C bits select among the four types of activations. For illustration purposes, the figure highlights the required components and subsequent activation bit rearrangements in the selectable inputs of the eighth multiplexer. Reliable activations (input 0) remain intact (a_i[15:0]). L activations (input 1) are 2-bit right shifted, preserving the sign (s) bit and introducing two logic '0' in bit positions 14 and 13, resulting in the bit rearrangement a_i[s,'00',14:2]. In the case of H activations (input 2), the flip operation is undone before performing the shift operation, resulting in the same representation as L activations. The management of the L&H activations is described in the next section.

4.2. Safe-bank technique

When a layer is written, activations classified as L&H ($C = 11$) are sequentially stored one after another in the safe (last) bank, starting at the last memory address and occupying ascending addresses. To do so, we employ a Safe Pointer (SP) stating the next address to be used for this purpose. On the other hand, when a layer is read, the SP pointer is set to the last memory address and L&H activations are read in the same order as they were stored.

For design simplicity, the same memory port is used to access both regular banks and the safe bank. In particular, L&H activations from the safe bank (a_{SP}[15:0]) are read cycle by cycle and temporarily stored in latches at input 3 of the corresponding multiplexers. Once all the L&H activations of a block are ready, the entire block is forwarded to the dispatcher. Section 5.3 quantifies the impact on system performance of the Safe-Bank technique.

Finally, note that a similar design is required to incorporate Shift-and-Safe in the write port of the activation memory to properly transform activations ahead of being stored.

4.3. Power, energy, area, and timing overhead

The proposed design necessitates two control (C) bits for every 16-bit activation word, resulting in a linear increase in storage overhead proportional to the activation memory size. For instance, in a 2 MiB activation memory, the C-bit overhead is 256 KiB. Although conventional memory designs already incorporate comparable control bits to discern between reliable and faulty contents [20,21], we conservatively consider their energy, area, and timing overhead. In fact, note that C bits would not be required for the safe bank. Nevertheless, we conservatively take into account their overhead.

Table 4 summarizes the leakage power and dynamic energy of activation memories under different operation modes. All the results were obtained with CACTI-P for a 32-nm technology node and ITRS low-power device type [31]. Conventional activation memories are labeled as Dynamic Voltage Scaling (DVS) or baseline (*Base*) when they are supplied at safe 0.6 V (V_{min}) or at ultra-low V_{dd}, respectively. Label *FaP* alludes to the Flip-and-Patch mechanism at 0.54 V, whereas *SaS* refers to the proposed Shift-and-Safe technique applied at ultra-low V_{dd}. For a given V_{dd} value, the overhead of SaS over Base consists of the required control bits, shifters, latches, and multiplexers. The overhead of supplying the control bits and safe bank at 0.6 V is also included.

As obtained with CACTI-P, the area of a conventional activation memory is 6.207 mm^2. Enhancing the memory with SaS imposes a total area overhead of 15.4%. Finally, the proposed changes in the read port of the memory increase the access time from 2.69 ns to 2.75 ns. We assume this small latency overhead does not compromise the cycle time of the accelerator.

5. Experimental evaluation

This section describes the simulation framework used to obtain experimental results. Then, CNN accuracy, system performance, and energy consumption of the proposed approach and two other recent techniques are evaluated under different supply voltages.

5.1. Simulation environment

Our fault-injection framework, like previous frameworks such as Ares [32], is built on top of the TensorFlow 2.5.0 library [33], which

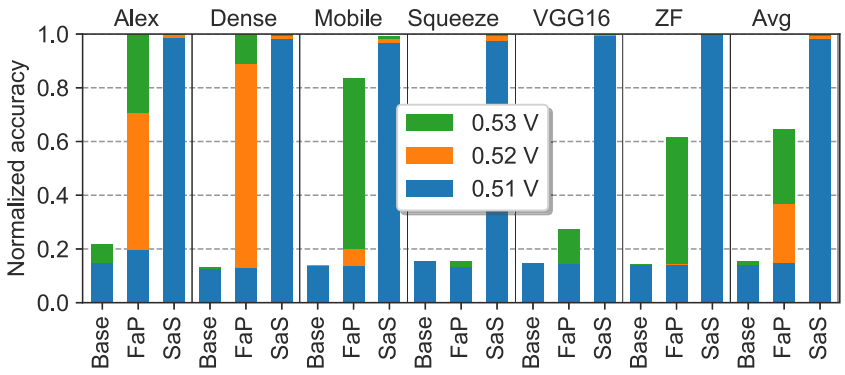

Fig. 7. Normalized accuracy of different approaches operating at ultra-low V_{dd} values with respect to a conventional fault-free operation mode ($V_{dd} \geq V_{min}$).

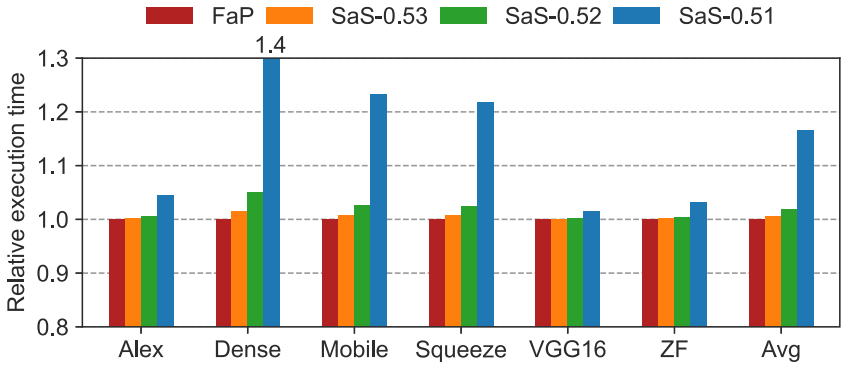

Fig. 8. Relative execution time of Flip-and-Patch (FaP) at 0.54 V and Shift-and-Safe (SaS) at different ultra-low V_{dd} values with respect to a conventional design.

runs high-level CNN descriptions in Python. Specifically, our framework monitors the output activation values, which are used as input for the next network layer, and alters them according to the faulty memory bitmaps provided by MoRS (see Section 2.2). This approach models activation memory behavior as it had been affected by permanent bitcell faults. During this monitoring stage, the framework also applies Shift-and-Safe operations to the activation values, as discussed in Section 4.

Additionally, the framework has been enhanced to simulate the dataflow of the baseline CNN accelerator architecture introduced in Section 2.1, incorporating the Shift-and-Safe technique within the memory ports, as well as previous Flip-and-Patch [10] and ThUnderVolt [5] approaches. In line with recent studies [34–36], our framework accurately measures execution time (in processor cycles), assuming an access latency of three cycles for the activation and weight memories and one cycle for the patching cache of the Flip-and-Patch technique. These latency values match the timing data provided by CACTI-P at a clock frequency of 1 GHz [31]. The systolic processing element (PE) array accounts a one-cycle delay for each partial sum and accumulation in the PE.

Apart from performance metrics like CNN accuracy and execution time, our framework also tracks the number of read/write memory accesses required to estimate energy expenses. These statistics are combined with per-access energy values obtained with CACTI-P (see Table 4) to calculate total energy consumption. Refer to Section 3.1 for a description of CNN benchmarks and input dataset. Finally, all the results presented in this work have been averaged for ten different fault maps per V_{dd} value according to Section 2.2.

5.2. Impact on accuracy

Fig. 7 plots the normalized accuracy of the proposed Shift-and-Safe (SaS) technique in activation memories supplied at different ultra-low V_{dd} values with respect to a fault-free operation mode with V_{dd} over V_{min}. For comparison purposes, results for the baseline (Base) scheme without any fault protection and Flip-and-Patch (FaP) are shown.

The baseline scheme severely affects the accuracy in all the benchmarks due to the high number of permanent faults, obtaining a random guessing output in most applications for all the V_{dd} values. FaP improves the accuracy, but it is still far from the golden value. This technique ensures the golden accuracy in AlexNet and DenseNet at 0.53 V, but fails to do so in the remaining benchmarks, leading to mostly a 40% accuracy degradation on average. Notice too that, at 0.51 V, FaP and Base behave the same way.

On the contrary, SaS outperforms FaP by applying shifts to *L+f(H)* activations and protecting all the L&H activations in the safe bank. At 0.51 V, where more than half of the total activations are faulty, the original accuracy is mostly recovered in all the applications.

5.3. Impact on system performance

Managing L&H activations in the safe bank of the proposed approach requires additional processor cycles. Fig. 8 plots the relative execution time (the lower the better) of SaS at different V_{dd} with respect to a conventional accelerator design. Performance of FaP at 0.54 V is also shown.

Exploiting the patching cache of FaP also requires additional cycles. However, this technique shows a negligible impact on performance due to the low number of L&H activations at 0.54 V. As expected, SaS incurs a higher performance penalty as V_{dd} underscales. Half of the benchmarks (DenseNet, MobileNet, and SqueezeNet) show a severe performance degradation at 0.51 V. This is mainly due to these networks are relatively deep in terms of number of layers and many of them are quite large, requiring an intensive use of the safe bank. Nevertheless, at 0.52 V, the performance loss does not exceed 5% (DenseNet) in any benchmark. On average, the performance penalty of SaS is by 0.6% and 1.8% at 0.53 V and 0.52 V, respectively.

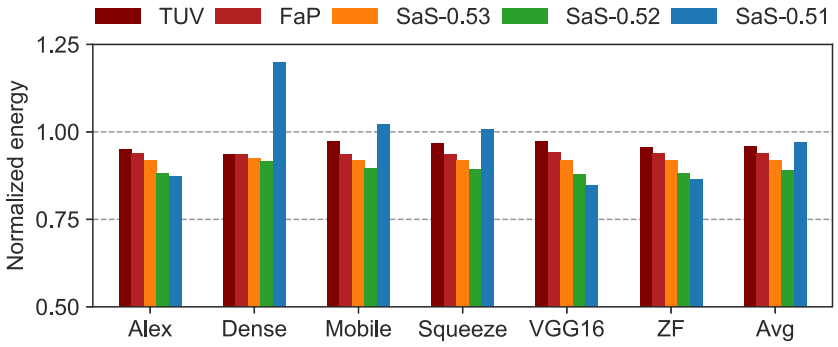

Fig. 9. Normalized energy consumption of an activation memory with ThUnderVolt (TUV) at a variable V_{dd}, Flip-and-Patch (FaP) at 0.54 V, and Shift-and-Safe (SaS) at ultra-low V_{dd} over conventional Dynamic Voltage Scaling (DVS) at 0.6 V.

5.4. Energy consumption

Fig. 9 shows the normalized energy consumption of an activation memory supplied at ultra-low V_{dd} values with SaS compared to a conventional memory powered at safe 0.6 V with DVS. For comparison purposes, energy consumption of FaP at 0.54 V and the ThUnderVolt (TUV) approach [5] at a variable V_{dd} is also analyzed. See Section 6.1 for details about the implementation of TUV. The reported energy comprises both leakage and dynamic expenses, including the overhead of the SaS components (see Section 4.3).

As expected, energy savings increase as V_{dd} underscales. For FaP and SaS techniques, energy savings are similar across most benchmarks for a given supply voltage. On the other hand, the impact on the execution time discussed in the previous section can be appreciated in the energy consumption of DenseNet, MobileNet, and SqueezeNet at 0.51 V. Notice too that, compared to TUV, energy reductions are greater for FaP and SaS approaches (except DenseNet where TUV and FaP consume a very similar energy). This is due to TUV requires most CNN layers to be powered with a $V_{dd} > 0.54$ V to maintain the golden accuracy.

Overall, at 0.52 V, SaS reduces the energy consumption on average by 5%, 7%, and 11% compared to FaP, TUV, and DVS, respectively. These energy savings might seem relatively low. However, it is worth noting that the studied supply voltage range below V_{min} is narrow (from 0.6 V to 0.51 V). Compared to a conventional activation memory supplied at nominal V_{dd} (0.9 V), the average energy savings are up to 40% at 0.52 V.

6. Related work

This section classifies related work into techniques addressing permanent and transient faults in CNN accelerators, patching solutions for general-purpose processors, and software efforts based on clipping algorithms.

6.1. Addressing permanent faults in CNN accelerators

Dealing with faults as a consequence of supply voltage underscaling in CNN accelerators has been explored from different angles, including the proposal of alternative representations of network parameters, additional fault-free memory storage, dynamic adjustments of V_{dd} at runtime according to reliability demands, FPGA compilation process enhancements, or network retraining methods.

Flip-and-Patch addresses permanent faults in on-chip activation memories with the proposal of an alternative data representation and the use of a dedicated patching cache [10]. See Section 2.3 for further details. Unfortunately, this paper has shown that Flip-and-Patch is ineffective at ultra-low V_{dd} with a high number of faults.

ThUnderVolt focuses on timing faults in the logic circuitry of the Processing Element (PE) array [5]. This technique is based on the fact that CNN layers exhibit different sensitivity to timing faults. Consequently, authors dynamically adjust V_{dd} below V_{min} for every layer, eliminating the impact of such faults on accuracy. However, as discussed in Section 5.4, the energy savings of ThUnderVolt are limited compared to the proposed Shift-and-Safe approach. In addition, such a fine-grain V_{dd} setup per layer imposes a complex profiling effort.

Note that, in our experimental evaluation, we adapted ThUnderVolt to the activation memories of the accelerator. In this sense, we carefully selected the most aggressive V_{dd} value for every CNN layer, making sure that the original accuracy remained unaffected. In addition, we conservatively do not take into account the timing and energy overhead associated with per-layer V_{dd} transitions [37].

Salami et al. alter the placement algorithm of an FPGA compilation process to circumvent permanent faults in memory blocks [6]. Particularly, this approach ensures that vulnerable CNN layers are not mapped to faulty blocks. However, contrary to our proposed technique, this approach requires application profiling.

Zhang et al. modify the training phase of a neural network by exposing permanent faults during the process, resulting in a set of weight values that hide the impact of faults on accuracy during the inference phase [7]. Similarly, Jia et al. propose to retrain neural networks under faults, but their approach is limited to classification layers identified as more vulnerable to faults [8]. The main downside of these solutions is that they depend on the specific neural network architecture and require programmer intervention.

Finally, with the aim to address the system performance of CNN accelerators caused by process variations, Tan et al. bypass the slower PEs that force all the remaining PEs to operate at low frequencies [9]. To do so, authors reallocate computations to idle PEs, preserving the original CNN accuracy. However, this approach only works for relatively small neural networks. To overcome this downside, authors enhance the prior approach with a weight transfer technique that moves the computations to be performed in slower PEs to faster neighboring PEs. However, this solution may compromise the original accuracy, requires complex transformations of weight filters to tailor each neural network to the specific accelerator, and demand a profiling process for effective implementation.

6.2. Addressing transient faults in CNN accelerators

Transient faults have been also addressed in on-chip weight memories [38] and registers within the PE array [39] of CNN accelerators. These works propose word and bit masking techniques, forcing faulty weights to zero values, and protecting the sign bit assuming it has the same logic value as the adjacent bit or vice versa. However, these techniques employ Razor double-sampling methods to detect such faults, which may impose a significant power overhead.

6.3. Patching techniques for general-purpose processors

Patching techniques are a viable solution to store replica values of vulnerable contents in general-purpose systems. In CPU superscalar processors, spare entries of pipeline structures like trace caches, MSHR, or store queues have been employed to maintain reliable replicas

of faulty L1 cache contents [20]. However, this approach imposes a burden to the design and verification of the processor, since memory consistency management has to be propagated to such pipeline structures.

In GPU register files, GR-Guard identifies dead registers with the assistance of the compiler and modifications to the instruction set, leveraging those dead entries to maintain replicas of faulty registers [21]. DC-Patch compresses registers at run time and forces its allocation to faulty register entries, making sure that defective bitcells are not used [40].

6.4. Software algorithms

Prior software efforts exploit clipping algorithms mitigate the impact of faults on accuracy by identifying long magnitude deviations in CNN parameters. Particularly, Ozen and Orailoglu use regularization terms to penalize outlier weights and minimize the loss function during the training phase [41]. During the inference phase, other works profile the CNN applications, introducing additional layers that restrict outlier values to a predefined numerical range [42,43].

7. Conclusions

This work has explored the potential of drastically underscaling the supply voltage (V_{dd}) below the safe voltage level (V_{min}) in on-chip activation memories of CNN inference accelerators to save energy. To address CNN accuracy drops due to permanent faults from voltage reductions, this paper has proposed two cost-effective microarchitectural strategies namely Shift and Safe-Bank. These strategies derive from an analysis showing that most significant bits of activation words are often '0' and the common underutilization of on-chip memory storage in CNN accelerators.

The shift-based technique adjusts the encoding of activations to minimize the impact of faults in the resulting value, whereas the Safe-Bank technique provides a fault-free secondary storage solution for activations where shifting results ineffective. Unlike most prior work, Shift-and-Safe is transparent to the programmer and does not rely on specific CNN application characteristics.

Experimental results have shown that, compared to a conventional CNN accelerator operating at V_{min} (0.6 V), an enhanced accelerator supplied at 0.52 V with Shift-and-Safe reduces the energy consumption of activation memories by 11% while maintaining the original accuracy with a minimal impact on system performance (less than 2% on average). Finally, compared to the state-of-the-art Flip-and-Patch technique and the ThUnderVolt mechanism adapted for on-chip memories, our proposed approach reduces energy consumption by 5% and 7%, respectively.

CRediT authorship contribution statement

Yamilka Toca-Díaz: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Rubén Gran Tejero:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Alejandro Valero:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

All authors acknowledge support from grants (1) PID2022-136454 NB-C22/AEI/10.13039/501100011033 from *Agencia Estatal de Investigación* (AEI), and (2) gaZ: T58_23R research group from Dept. of Science, University and Knowledge Society, Government of Aragon. The funding agencies had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Data availability

Data will be made available on request.

References

[1] J. Leng, Y. Zu, V.J. Reddi, GPU voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in GPU architectures, in: Proceedings of the IEEE 21st International Symposium on High Performance Computer Architecture, 2015, pp. 161–173.

[2] C. Wilkerson, H. Gao, A.R. Alameldeen, Z. Chishti, M. Khellah, S. Lu, Trading off cache capacity for reliability to enable low voltage operation, in: Proceedings of the ACM/IEEE 35th Annual International Symposium on Computer Architecture, 2008, pp. 203–214.

[3] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, J. Hoe, Multi-bit error tolerant caches using two-dimensional error coding, in: Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture, 2007, pp. 197–209.

[4] J. Tan, Q. Wang, K. Yan, X. Wei, X. Fu, Saca-FI: A microarchitecture-level fault injection framework for reliability analysis of systolic array based CNN accelerator, Elsevier Future Gener. Comput. Syst. 147 (2023) 251–264.

[5] J. Zhang, K. Rangineni, Z. Ghodsi, S. Garg, ThUnderVolt: Enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators, in: Proceedings of the 55th ACM/ESDA/IEEE Design Automation Conference, 2018, pp. 1–6.

[6] B. Salami, O. S. Unsal, A. Cristal Kestelman, Comprehensive evaluation of supply voltage underscaling in FPGA on-chip memories, in: Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture, 2018, pp. 724–736.

[7] J.J. Zhang, T. Gu, K. Basu, S. Garg, Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator, in: Proceedings of the IEEE 36th VLSI Test Symposium, 2018, pp. 1–6.

[8] K. Jia, Z. Liu, Q. Wei, F. Qiao, X. Liu, Y. Yang, H. Fan, H. Yang, Calibrating process variation at system level with in-situ low-precision transfer learning for analog neural network processors, in: Proceedings of the 55th Annual Design Automation Conference, 2018, pp. 1–6.

[9] J. Tan, W. Wang, M. Ma, X. Wei, K. Yan, Improving the performance of CNN accelerator architecture under the impact of process variations, ACM Trans. Des. Autom. Electron. Syst. 28 (5) (2023) 1–21.

[10] Y. Toca-Díaz, R. Hernández Palacios, R. Gran Tejero, A. Valero, Flip-and-Patch: A fault-tolerant technique for on-chip memories of CNN accelerators at low supply voltage, Elsevier Microprocess. Microsyst. 106 (2024) 1–13.

[11] N. Landeros Muñoz, A. Valero, R. Gran Tejero, D. Zoni, Gated-CNN: Combating NBTI and HCI aging effects in on-chip activation memories of convolutional neural network accelerators, Elsevier J. Syst. Archit. 128 (2022) 1–13.

[12] N.P. Jouppi, C. Young, N. Patil, D.A. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T.V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C.R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, D.H. Yoon, In-datacenter performance analysis of a tensor processing unit, in: Proceedings of the 44th Annual International Symposium on Computer Architecture, 2017, pp. 1–12.

[13] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, O. Temam, DaDianNao: A machine-learning supercomputer, in: Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, 2014, pp. 609–622.

[14] K. Seshadri, B. Akin, J. Laudon, R. Narayanaswami, A. Yazdanbakhsh, An evaluation of edge TPU accelerators for convolutional neural networks, in: Proceedings of the IEEE International Symposium on Workload Characterization, 2022, pp. 79–91.

[15] A. Samajdar, Y. Zhu, P.N. Whatmough, M. Mattina, T. Krishna, SCALE-Sim: Systolic CNN accelerator, 2018, CoRR arXiv:1811.02883.

[16] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M.A. Horowitz, W.J. Dally, EIE: Efficient inference engine on compressed deep neural network, in: Proceedings of the 43rd International Symposium on Computer Architecture, 2016, pp. 243–254.
[17] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, V. Chandra, H. Esmaeilzadeh, Bit Fusion: Bit-level dynamically composable architecture for accelerating deep neural network, in: Proceedings of the ACM/IEEE 45th Annual International Symposium on Computer Architecture, 2018, pp. 764–775.
[18] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, H.-J. Yoo, UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision, IEEE J. Solid-State Circuits 54 (2019) 173–185.
[19] I.E. Yüksel, B. Salami, O.g. Ergin, O.S. Unsal, A.C. Kestelman, MoRS: An approximate fault modeling framework for reduced-voltage SRAMs, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 41 (6) (2022) 1663–1673.
[20] D.J. Palframan, N. Kim, M.H. Lipasti, iPatch: Intelligent fault patching to improve energy efficiency, in: Proceedings of the IEEE 21st International Symposium on High Performance Computer Architecture, 2015, pp. 428–438.
[21] J. Tan, S.L. Song, K. Yan, X. Fu, A. Marquez, D. Kerbyson, Combating the reliability challenge of GPU register file at low supply voltage, in: Proceedings of the 25th International Conference on Parallel Architectures and Compilation Techniques, 2016, pp. 3–15.
[22] A. Chatzidimitriou, G. Panadimitriou, D. Gizopoulos, S. Ganapathy, J. Kalamatianos, Assessing the effects of low voltage in branch prediction units, in: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, 2019, pp. 127–136.
[23] ARM, ARM11 MPCore™ Processor. Revision: r2p0. Technical Reference Manual, Tech. rep., ARM Limited, 2008.
[24] J.N. Kather, C.-A. Weis, F. Bianconi, S.M. Melchers, L.R. Schad, T. Gaiser, A. Marx, F.G. Zöllner, Multi-class texture analysis in colorectal cancer histology, Nat. Sci. Rep. 6 (2016).
[25] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[26] F.N. Iandola, M.W. Moskewicz, S. Karayev, R.B. Girshick, T. Darrell, K. Keutzer, DenseNet: Implementing efficient ConvNet descriptor pyramids, 2014, CoRR arXiv:1404.1869.
[27] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, CoRR arXiv:1704.04861.
[28] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, 2016, CoRR arXiv:1602.07360.
[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, CoRR arXiv:1409.1556.
[30] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, Springer Lect. Notes Comput. Sci. 8689 (2014).
[31] R. Balasubramonian, A.B. Kahng, N. Muralimanohar, A. Shafiee, V. Srinivas, CACTI 7: New tools for interconnect exploration in innovative off-chip memories, ACM Trans. Archit. Code Optim. 14 (2) (2017) 1–25.
[32] B. Reagen, U. Gupta, L. Pentecost, P. Whatmough, S.K. Lee, N. Mulholland, D. Brooks, G.-Y. Wei, Ares: A framework for quantifying the resilience of deep neural networks, in: Proceedings of the 55th ACM/ESDA/IEEE Design Automation Conference, 2018, pp. 1–6.
[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, 2016, CoRR arXiv:1603.04467.
[34] A. Parashar, P. Raina, Y.S. Shao, Y.-H. Chen, V.A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S.W. Keckler, J. Emer, Timeloop: A systematic approach to DNN accelerator evaluation, in: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, 2019, pp. 304–315.
[35] L. Mei, H. Liu, T. Wu, H.E. Sumbul, M. Verhelst, E. Beigne, A uniform latency model for DNN accelerators with diverse architectures and dataflows, in: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition, 2022, pp. 220–225.
[36] T. Hotfilter, P. Schmidt, J. Höfer, F. Kreß, T. Harbaum, J. Becker, An analytical model of configurable systolic arrays to find the best-fitting accelerator for a given DNN workload, in: Proceedings of the DroneSE and RAPIDO: System Engineering for Constrained Embedded Systems, 2023, pp. 73–78.
[37] J. Park, D. Shin, N. Chang, M. Pedram, Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors, in: Proceedings of the ACM/IEEE International Symposium on Low-Power Electronics and Design, 2010, pp. 419–424.
[38] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S.K. Lee, J.M. Hernández-Lobato, G.-Y. Wei, D. Brooks, Minerva: Enabling low-power, highly-accurate deep neural network accelerators, in: Proceedings of the 43rd International Symposium on Computer Architecture, 2016, pp. 267–278.
[39] B. Salami, O.S. Unsal, A.C. Kestelman, On the resilience of RTL NN accelerators: Fault characterization and mitigation, in: Proceedings of the 30th International Symposium on Computer Architecture and High Performance Computing, 2018, pp. 322–329.
[40] A. Valero, D. Suárez-Gracia, R. Gran-Tejero, DC-Patch: A microarchitectural fault patching technique for GPU register files, IEEE Access 8 (2020) 173276–173288.
[41] E. Ozen, A. Orailoglu, SNR: Squeezing numerical range defuses bit error vulnerability surface in deep neural networks, ACM Trans. Embed. Comput. Syst. 20 (5s) (2021) 1–25.
[42] L.-H. Hoang, M.A. Hanif, M. Shafique, FT-ClipAct: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation, in: Proceedings of the 23rd Conference on Design, Automation and Test in Europe, 2020, pp. 1241–1246.
[43] Z. Chen, G. Li, K. Pattabiraman, A low-cost fault corrector for deep neural networks through range restriction, in: Proceedings of the 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2021, pp. 1–13.

Yamilka Toca-Díaz received the B.S. and M.S. degrees in Computer Science from Universidad de Camagüey, Cuba, in 2007 and 2013, respectively. She is currently working toward the Ph.D. degree in Computer Engineering at the Department of Computer Science and Systems Engineering, Universidad de Zaragoza, Spain. Her research interests include the design of machine learning accelerators with a focus on reliability.

Rubén Gran Tejero graduated in Computer Science from University of Zaragoza, Spain. He received his Ph.D. from Polytechnic University of Catalonia (UPC), Spain, in 2010. He is currently an Associate Professor in the Department of Computer Science and Systems Engineering at University of Zaragoza. He has been Program Committee Member of several conferences and workshops in the area: IPDPS, ICCD, HPCS, and PMBS. His research interests include hard real-time systems, hardware for reducing worst-case execution time and energy consumption, efficient processor microarchitecture, and effective programming for parallel and heterogeneous systems.

Alejandro Valero received the Ph.D. degree in Computer Engineering from Universitat Politècnica de València, Spain, in 2013. From 2013 to 2015, he was a visiting researcher with Northeastern University, Boston, MA, USA, and University of Cambridge, UK. From 2016 to 2021, he was an Assistant Professor with the Department of Computer Science and Systems Engineering, Universidad de Zaragoza, Spain. Since 2021, he has been an Associate Professor with the same department and institution. His Ph.D. research was recognized with multiple awards, including the 2012 Intel Doctoral Student Honor Award and the Gold Medal in the 2013 ACM Student Research Competition (SRC) held in ICS-27. He has been Technical Program Committee Member of several conferences, workshops, and research competitions, including DATE, ICCD, PMBS, and ACM SRC Grand Finals. His research interests include GPU and ASIC architectures, memory hierarchy design, energy efficiency, and fault tolerance. Prof. Valero is a member of the Aragon Institute of Engineering Research (I3A) and the HiPEAC European NoE.