When AI Gets Old: The Effect of Asymmetric Aging in Deep Learning Accelerators

ABSTRACT

Deep neural networks (DNNs) offer phenomenal performance in an ever-increasing number of applications, such as computer vision, natural language processing, video analytics, and mission-critical systems. The growing computational complexity of such models has propelled the development of specialized accelerators that offer improved performance and energy efficiency. Advanced VLSI process nodes have further intensified the development of machine learning (ML) accelerators by providing remarkable transistor miniaturization and power efficiency. Nonetheless, these process nodes are vulnerable to transistor aging, which can lead to a gradual decline in the performance, prediction accuracy, and reliability of ML accelerators and introduce significant reliability concerns. In this work, we present a comprehensive study of how aging affects systolic arrays, which are at the core of many ML accelerators, such as Google's Tensor Processing Unit. Our experimental analysis indicates that systolic arrays undergo asymmetric aging, where logical elements age at different rates. In addition, we show that asymmetric aging produces persistent and transient errors that manifest in the datapath of a systolic array, which in turn may cause major faults in their overall operation and thereby severely degrade the resiliency of the ML model. For example, considering less than 1% of the overall transient failure events, the top-1 prediction accuracy of the Res-Net-18 model drops by 40%. We introduce hardware mechanisms and design flow solutions that mitigate the impact of asymmetric aging reliability on ML accelerators and achieve the original top-1 prediction accuracy of the DNN model.

1. INTRODUCTION

Deep neural networks (DNNs) play a major role in numerous applications, such as recommendation systems, natural language processing, and vision recognition. DNN models can learn and recognize complex patterns and features in large data sets. They are also computationally intensive and require significant processing resources for both training and inference. DNNs consist of multiple layers, where each layer comprises a large-scale matrix multiplication or a convolution operation, which is usually followed by an activation function. Both matrix multiplications and convolutions incur numerous multiply and accumulate (MAC) operations and constitute the lion's share of many machine learning (ML) processing workloads. For example, GoogLeNet [42] and ResNet-101 [15] require approximately 1.5 and 7.8 billion MAC operations, respectively, for a single inference assuming an image resolution of 224×224 pixels.

The deployment of DNNs in diverse platforms with different processing capabilities, real-time requirements, and energy constraints has encouraged the development of specialized accelerators [20, 24]. In addition, DNNs have also been used recently in mission-critical systems such as autonomous vehicles, medical appliances, finance, and security systems [18, 25, 27, 36]. All these new applications set a high bar for DNN resiliency and reliability, which are enforced by regulatory agencies and industry standards [13].

Over the last decade, the semiconductor industry has continued to push the boundaries of VLSI technologies, with several notable trends: New process nodes have continued to keep pace with Moore's law and miniaturize transistors to nanometric dimensions. New materials and devices that offer improved performance and reduced power consumption have been developed. However, the latest advances have exposed the susceptibility of semiconductors to reliability concerns, particularly concerns regarding transistor aging. Transistor aging is the gradual degradation over time of a transistor's performance due to hot carrier injection (HCI) and the bias temperature instability (BTI) [4, 31, 44], which are described in Section 2. This study focuses on BTI, which is widely acknowledged as the predominant aging mechanism in modern integrated circuits.

Transistor aging significantly affects the reliability of DNN accelerators, resulting in substantial performance degradation and serious circuit failures due to setup-timing violations. Asymmetric aging [10] occurs when the aging degradation is unevenly distributed among logical elements, resulting in more severe reliability issues that can lead to overall system failure. Asymmetric aging intensifies setup-timing violations and introduces hold-timing violations, which cannot be mitigated by reducing the clock frequency.

This paper uses SAs as a case study to determine how asymmetric aging affects DNN accelerators. We demonstrate that asymmetric aging causes persistent and transient faults in DNNs, thereby decreasing prediction accuracy and confidence levels. In mission-critical systems, such faults can have catastrophic consequences, potentially even violating functional safety. Our experimental analysis uses three frameworks: (i) functional simulations that use different workloads to extract the aging profile of SAs; (ii) detailed timing analysis coupled with aging models run on a physical implementation of a SA to pinpoint the failure points resulting from asymmetric aging; and (iii) an error-injection model that represents asymmetric aging transient and persistent errors to evaluate the overall impact on DNN performance.

The experimental results indicate that SA DNN accelerators can experience asymmetric aging, which results in persistent transient errors that propagate in the datapath of the array, which not only causes significant faults in the SA but also severely impacts the resiliency of ML models. In addition, our analysis reveals four primary mechanisms that encourage asymmetric aging in SAs: (i) DNN sparsity, (ii) underutilization of the dynamic range for value representation, (iii) clock gating, and (iv) lack of symmetry between logical-cell delays and wire delays.

Our study proposes both hardware and design flow approaches to address the impact of asymmetric aging on ML accelerators. We evaluate the effectiveness and overhead of our solutions on an SA. Our area and power analyses show that, with nearly 1% logical-cell area overhead and 7.85% power overhead, we can fully mitigate the effect of asymmetric aging on the prediction accuracy of model top-1. In addition, we show that a 7% reduction in the SA clock frequency avoids power overhead.

The primary contributions of this paper are as follows:

1. We use SAs as a case study to analyze in-depth transistor aging in DNN accelerators and demonstrate that asymmetric aging can lead to major faults and reliability concerns.
2. We determine that data sparsity, power-saving measures, underutilization of dynamic range of values, and asymmetry in timing delays between wires and cells promote asymmetric aging.
3. Our analysis shows that the spatial location of PEs contributes significantly to the likelihood of incurring asymmetric-aging-related faults.
4. We identify the internal elements and logical paths of a PE that are susceptible to asymmetric aging.
5. The proposed fault model indicates that asymmetric aging transient errors can accumulate within the PE and spread to neighboring PEs and successive DNN layers.
6. We offer hardware- and design-flow solutions to mitigate asymmetric aging in SAs and demonstrate that our techniques avoid degrading the top-1 prediction accuracy of the DNN model.

2. BACKGROUND AND PRIOR WORKS

This section overviews SA architecture, transistor aging, asymmetric aging, and DNN-resiliency-related works.

2.1 Deep Neural Network Accelerators

DNN hardware accelerators are specialized devices designed to accelerate the execution of DNN models. Several types of DNN hardware accelerators exist, such as graphical processing units [21], application-specific integrated circuits [28], which are custom-designed for specific applications, and Tensor Processing Units (TPUs) [20], which use SAs [24] for both ML training and inference.

A SA, which we use in our case study of DNN accelerators, is a homogeneous two-dimensional grid of processing elements (PEs), usually built from multiply and MACs that work coherently together to implement matrix multiplication. The inputs are passed from one PE to its neighbors, and every PE conducts a multiply-accumulate operation between the inputs and stores the intermediate result locally, then transmits the inputs for the adjacent PEs for the next cycle. Given the well-defined interactions between neighboring PEs, tasks can be executed efficiently and data reuse and scalability are possible [12].

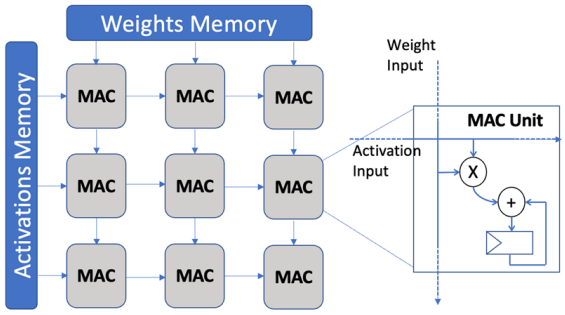

Figure 1: Output-stationary systolic array.

SAs have different forms and shapes and may be used for various tasks. In our work, we use the output stationary (OS) SA variant, which is used to accelerate and efficiently execute matrix multiplication in many different DNN- and ML-related applications. Figure 1 shows the state-of-the-art OS-SA architecture.

DNNs are computation and memory intensive, raising the demand for DNN hardware acceleration to a critical level. SA-based DNN hardware accelerator can offer a significant performance and throughput boost compared with a CPU [20]. MIT Eyeriss is another example of a SA accelerator for convolutional neural networks [9]. Another commercially used SA implementation is Tesla's full self-driving chip [43]. In addition, SAs have been used in multiple fields, for example, for neurocomputing [41], language recognition [**?**], and character string manipulation [26].

2.2 Transistor Aging

Transistor aging is the deterioration over time of transistors in logical elements. Two physical mechanisms govern transistor aging: HCI and BTI [4, 31, 44]. HCI occurs when high kinetic current flows through a transistor, whereas BTI occurs when a static voltage (logical state) is applied to the gate of a transistor without current flow for a long period, typically ranging from 10 s to several weeks [45]. Both BTI and HCI increase the transistor threshold voltage, which increases the switching delay. This study focuses on BTI because it is the dominant aging mechanism in modern integrated circuits [2,37]. The BTI aging model we use to represent the increase in threshold voltage is based on the reaction-diffusion model, which is the main model used by the semiconductor industry [1,2,5,6]. The threshold voltage increment ΔV_{th} due to BTI stress is

$$-\Delta V_{\text{th}} \propto e^{\frac{E_a}{kT}} (t - t_0)^{1/6} \quad (1)$$

where E_a is a constant, T is the operating temperature, k is Boltzmann's constant, t_0 is the time when the BTI stress starts, and t is the overall time. p-type transistors are more susceptible to BTI (known as NBTI) than n-type transistors (known as PBTI) [39]. Therefore, logical gates with a constant idle state of logical 0 are most vulnerable to aging. A

common method to measure the BTI stress profile on logical elements is the signal probability (SP). The SP represents the likelihood that a signal will have a logical value of 1, and it is the ratio of the time a signal spends in the logical 1 state to the overall time. Decreasing the SP increases the likelihood of BTI in the circuit and degrades the circuit performance over time or even causes failure.

BTI can significantly degrade the performance of a logical circuit, and if the degradation is symmetric among all logical elements, it can be mitigated by reducing the clock frequency. However, degradation due to asymmetric aging may produce even more severe reliability concerns.

2.3 Asymmetric Aging

Asymmetric aging occurs when the transistor degradation is nonuniformly distributed between logical elements such as flip flops, gates, clock tree buffers, and memory cells. The high complexity of asymmetric aging presents significant challenges for integrated circuits in terms of modeling, analysis, prediction, and prevention, making it a major reliability concern. Moreover, incorporating detailed timing analysis that considers aging is nontrivial because it depends on the workload and operating conditions, a capability that is absent in conventional design tools [45].

In the next three sections, we identify four primary mechanisms that promote asymmetric aging in SAs: clock gating, DNN sparsity, and asymmetrical delay between logical elements and wires. Each of these mechanisms can independently lead to asymmetric aging, eventually causing severe timing violations and permanent as well as transient faults. The following discussion provides more insight into each of these mechanisms.

2.3.1 Clock gating

One widely accepted method for dynamic power saving is clock gating [40], which involves selectively blocking the clock signal in currently unused parts of the circuit, thereby reducing dynamic power consumption. By turning off the clock in idle parts of the circuit, unnecessary switching and associated power consumption are eliminated. Clock gating is typically implemented by using a clock gate cell containing an AND or OR gate. When the clock is enabled, the clock signal is allowed to pass through the clock gate cell. When the clock is disabled, the output of the gate is held at a constant logic value, blocking the clock signal from passing through the gate.

Clock gating induces BTI because it intensifies the idleness on the clock network and on combinational circuits. In addition, it encourages asymmetric aging, as illustrated in Figs. 2(a) and 2(b). In Fig. 2(a), the clock gate is used in the launch path, causing greater aging in the launch path than in the capture path. This asymmetry can lead to setup-timing violations. Conversely, in Fig. 2(b), using the clock gate in the capture path intensifies the aging in that path compared with the launch path, resulting in hold-timing violations.

2.3.2 Asymmetry between logical-cell delays and wire delays

Another cause of asymmetric aging is the asymmetry between the accumulated delay of logical cells and wires. Although logical cells are affected by BTI, wires are not. When launch

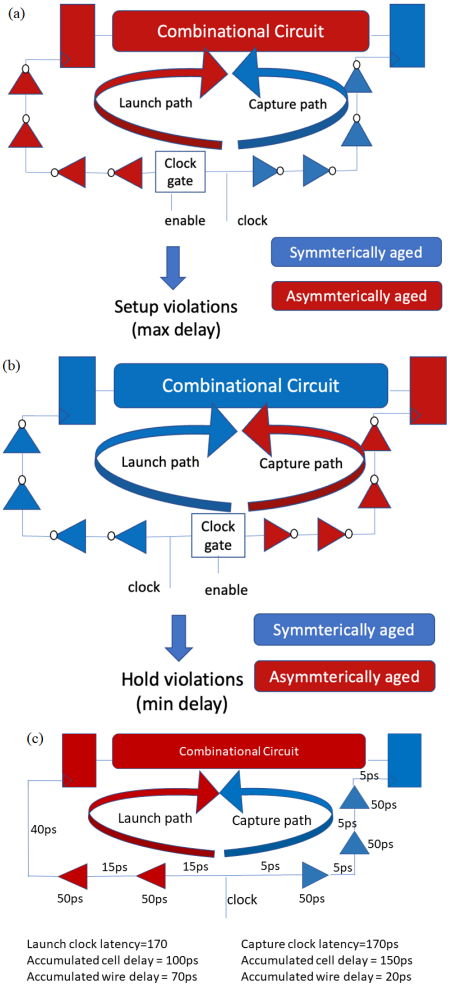

Figure 2: (a) Possible violation due to asymmetric aging induced by (a) launch path clock gate, (b) capture path clock gate, and (c) the asymmetry between the accumulated delay of logical cells and wires.

and capture paths have different accumulated logical-cell delays, BTI can induce asymmetric aging, as shown in Fig. 2(c). If the accumulated logical-cell delay in the launch path is greater than that in the capture path, setup-timing violations may occur. Conversely, if the accumulated cell delay in the launch path is smaller than that in the capture path, it may cause hold-timing violations. As illustrated in Fig. 2(c), both launch and capture clocks are balanced with 170 ps clock latency. However, the accumulated clock buffer delay in the capture path is 150 ps, whereas the clock buffer total delay in the launch path is 100 ps. Such asymmetry between the accumulated cell and wire delay in conjunction with BTI may result in hold-timing violations due to the delay shift in the capture clock. Previous works such as [3] have ignored wire delays; however, our experimental analysis shows that this phenomenon can contribute significantly to asymmetric aging.

2.3.3 *Deep neural network sparsity*

DNNs can exhibit a high degree of sparsity for several reasons, including

1. the use of certain activation functions, such as ReLU [30];
2. various DNN optimizations to avoid overfitting, such as dropout regularization, pruning, and weight decay [16];
3. sparsity in the DNN model [16];
4. when the dynamic range for value representation is not fully utilized (e.g., when the data type used is 16-bit wide, but weights and activations use 8 bits).

As noted earlier, the constant voltage bias of the logical 0 state can promote BTI in DNN accelerators, particularly those induced by sparsity. Moreover, since sparsity is not uniformly distributed across all logical elements and paths in the SA, asymmetric aging may result. For example, if the most significant bits in activations and weights exhibit a high degree of sparsity, aging may intensify on those logical paths with respect to other elements in the SA.

2.4 DNN-Resiliency-Related Works

The need for reliable DNNs accelerators has motivated numerous researchers to study the robustness against both permanent and transient faults of SA-based DNN accelerators. Permanent faults in data paths were studied in multiple works. For example, in [47, 48] the authors showed that, even for fault rates as low as 0.003%, the DNN's accuracy drops significantly from 74.13% to 39.69%. In addition, the authors proposed two techniques to enhance fault tolerance: fault-aware pruning and fault-aware pruning and retraining. Both techniques allow TPUs to work with fault rates as high as 50%. By using the discrete-time Markov chain formalism, the authors of [23] analyzed permanent manufacturing faults and revealed that the accuracy drops from 97.72% to 10.15% in some cases.

Conversely, [14, 32] explore how transient faults affect SAs and DNN models' inference accuracy, along with proposing, high-performance, energy-efficient design for fault prediction and mitigation in near-threshold operation mode for TPUs. Reference [33] examined timing error arising from near-threshold computing. Additionally, Kundu et al. in [22] provided a comprehensive study of both permanent and transient faults for quantized DNNs in SA-based accelerators and assessed in detail their performance in the presence of these errors. Moreover, the authors comparatively analyzed how the decrease in accuracy depends on fault location and proposed efficient methods for in-field functional testing. First, they showed that stuck-at-1 faults produce a much larger effect on accuracy than stuck-at-0 faults. Second, faults in the most significant bits have a larger impact than faults in the least significant bits. Finally, they found that faults in the first two layers have a greater impact than those in lower layers. Nevertheless, none of the works mentioned above examined the impact of aging-induced faults.

Aging-induced faults in SAs have been mentioned in only a few prior works. Reference [38]proposed a new quantization method to eliminate aging guard bands, thus minimizing aging-induced frequency degradation. As part of their work [17] to accelerate timing simulations in SA-based accelerators, Holst et al. proposed a new method to measure DNN accuracy losses caused by arbitrary timing faults. They also discussed how injecting one small-delay random defect in different numbers of PEs affects the inference accuracy.

Additional works such as [19, 29, 34] comprehensively review the manifestation and mitigation techniques (hardware and software) of soft errors induced from multiple sources such as radiation, process variations, temperature, and aging in DNN accelerators, including SA-based accelerators. However, none of these works discussed faults induced by asymmetric aging.

Thus, no previous work appears to have studied asymmetric-aging-induced timing errors in SAs or how they affect DNN inference accuracy. Other works have approached the asymmetric-aging phenomenon from different directions: [10] introduced an asymmetric-aging-aware microarchitecture to mitigate the impact of asymmetric aging on execution units, register files, and memory hierarchy in microprocessors with minimal overhead. Furthermore, [3] proposed an algorithm for analyzing the static timing of asymmetric aging in clock networks.

3. ASYMMETRIC-AGING-INDUCED FAULTS IN SYSTOLIC ARRAYS

Detecting faults induced by asymmetric aging in SAs involves two experimental phases. In the first phase, we analyze the aging profile of the SA architecture and DNN models by evaluating the SP of the microarchitectural elements in the SA. In the second phase, we fully implement the SA, including synthesis, place, route, and timing analysis, by using aging models that represent BTI timing degradation. Through timing analysis, we pinpoint the logical paths that suffer from asymmetric-aging-induced timing violations.

Figure 3 shows a PE cell in the SA under examination. The weight and activation inputs are sampled by registers and forwarded to the neighboring PEs. To reduce clock cycle time, the MAC operation of the PE is pipelined such that the multiplier output is sampled by a register and used in the next clock cycle by the accumulator. The illustrated PE uses three clock gates to save energy consumption in the following two scenarios:

1. Given that certain PEs may not be involved in matrix multiplication operations (as described later), clock gate

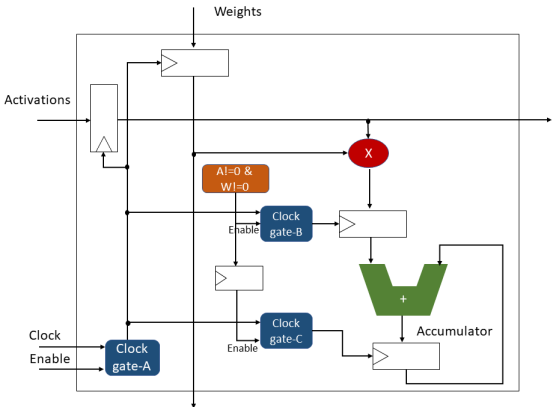

Figure 3: Systolic array processing element.

A is used to disable the clock f the PE.

2. Given zero activation or zero weight, clock gate B disables the clock of the output multiplier sampling register. Clock gate C disables the clock to the accumulator register in the next clock cycle.

3.1 Experimental Environment

Our experimental analysis is based on two environments: a simulation environment that emulates the operation of an OS-SA and extracts the aging profile and a detailed timing analysis electronic design automation environment that examines the impact of asymmetric aging based on the aging profile.

For the aging profile extraction, we have run a co-simulation that consists of a C++ SA simulator that runs in conjunction with a PyTorch-based DNN model written in Python. Our SA simulator is configured to simulate a 128×128 OS SA. As a case study we use pre-trained ResNet-18 and ResNet-50 [**?**] DNN models in PyTorch. The models' weights and activations are quantized to 8-bit signed and unsigned integers, respectively. In addition, we assumed an 8×8-bit integer multiplier and a 32-bit integer accumulator. For the inference process, we used 100 images chosen randomly from the ImageNet dataset [21].

For the timing analysis, we coded the SA in SystemVerilog and synthesized it for 28 nm process technology using Cadence® Genus®. For the place-and-route, we used the Cadence® Innovus® implementation tool. We assumed a SA clock frequency of 340 MHz and adopted as our aging model the reaction-diffusion model, which is widely accepted by industry and research as the preferred model for BTI aging [1,5,7,8]. The timing analysis with the aging model is like the method used in [10, 11]. Their corresponding degradation factors derate the propagation delay of logical elements as a function of their SP extracted in Sec. 3.2.

Figure 4 shows the delay shift of gates under different SPs using our aging model. It also presents the absolute delay shift of gates under variable SPs relative to gates that are symmetrically aged with SP = 0.5. The comparison demonstrates the asymmetrical delay shift of logical elements under constant BTI stress compared with other elements within a logical circuit that are symmetrically aged. The results show that gates with constant stress (when SP = 0 or 1) experience

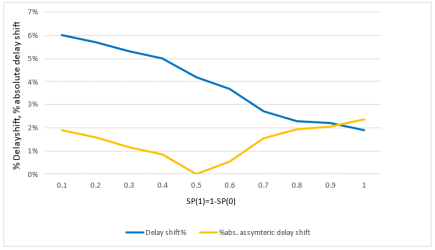

Figure 4: Frequency degradation and absolute asymmetric delay shift over a ten-year lifetime.

a 2.0%–2.5% asymmetric delay shift relative to gates with SP = 0.5. These results reveal that gates with a static stress of 1 may also suffer from this phenomenon despite having minor BTI stress. However, when compared with gates with SP = 0.5, the delay shift becomes significant. The observed asymmetric delay shift, even one as small as 2%–3%, can significantly impact circuit reliability.

3.2 Systolic Array Aging Profile

The measured SP and the idleness of logical elements within every PE describe the aging profile of the SA. Figure 5 shows a sample of heatmaps for activations, weights, multiplier output, and accumulator for ResNet-18 and ResNet-50, respectively, on a subset of ImageNet images. One of our first observations is that matrix multiplications and convolutions within the DNN do not exploit the full spatial dimension of the SA. For example, when the dimensions of a matrix multiplication are smaller than those of the SA, unused rows and columns are clock-gated due to power-saving considerations and therefore kept idle. As a result, PEs in the upper rows and columns are significantly less utilized, which can encourage asymmetric aging relative to all other PEs. We partition each SA into four regions: Region A contains the 68 lower rows and columns, region B contains the 68 lower rows and 64 upper columns, region C contains the 60 upper rows and 64 lower columns, and region D contains the remaining rows and columns.

Our experimental study suggests that logical elements within the SA may experience asymmetric BTI stress, and thereby age differently while inducing critical timing violations.

We summarize in Table 1 the root causes for these potential violations, which we group into the following classes:

1. SA utilization. Our observations indicate that SA regions B, C, and D are underutilized, which incurs idleness and BTI stress. Therefore, they can become susceptible to asymmetric aging, which can induce both setup- and hold-timing violations.
2. Dynamic range. Underutilizing the dynamic range of value representation increases the likelihood that certain bits or signals are in a constant logical state and thereby incur asymmetric aging. This encourages BTI stress on logical computational elements, increases their propagation delay, and results in setup-timing violations.
3. Sparsity. Exploiting sparsity can save unneeded operations and reduce power consumption. However, our analysis indicates that sparsity encourages BTI stress on logical elements and the SA gated clocks. When applied

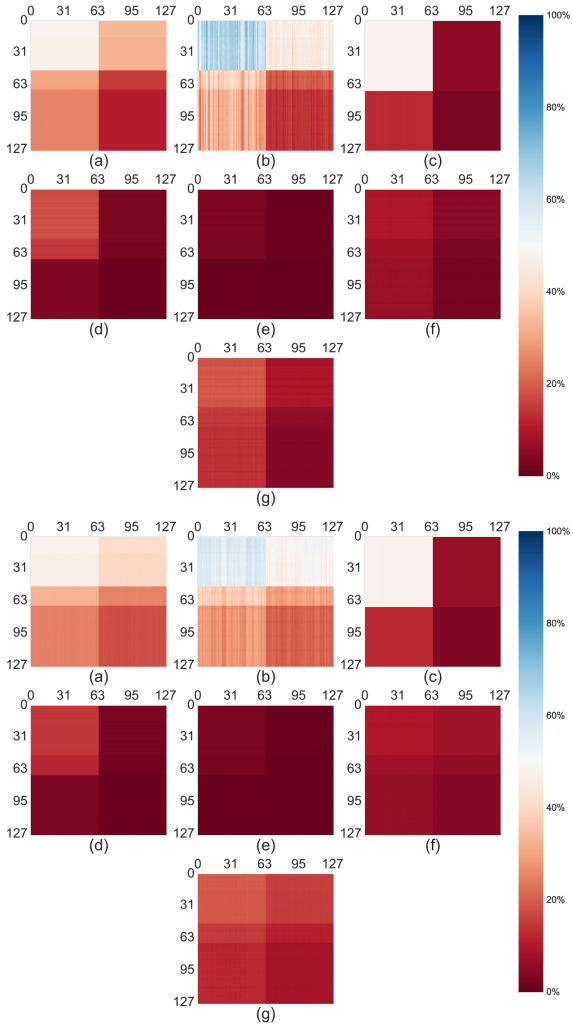

Figure 5: ResNet-18 (top) and ResNet-50 (bottom) signal probability heatmaps for a sample of ImageNet images inference: (a) accumulator bits 0—15, (b) accumulator bits 16—31, (c) weight bits 0-–7, (d) activation bits 0—5, (e) activation bits 6–7, (f) multiplier bit 0, and (g) multiplier bits 1—15.

to logical elements, BTI stress may cause setup-timing violations. In addition, in the case of gated clocks, BTI stress promotes hold-timing violations.

Activations. In region A, the SP of activation bits 0—5 ranges from 15%—19%, while bits 6 and 7 have a significantly lower SP (<0.25%) because (i) the full dynamic range of the Int8 representation is underused, and (ii) high sparsity. In regions B, C, and D, the SP of all activations is even smaller than 5%, which is attributed to the low utilization of these regions.

Weights. Unlike the activations, the SP of weights is approximately 50% in region A. However, in regions B, C, and D, SP is less than 15% due to the low utilization of these regions.

Multiplier output. The SP of bit 0 and bits 1–15 of the multiplier output in the SA falls within the range of 2%–10% and 4%–20%, respectively. The least significant bit of the multiplication product has a lower SP than the other higher-order bits because the likelihood of the product of two arbitrary integers being even is 0.75. Our analysis indicates that the low overall SP of the multiplier can be attributed to the following factors: (i) relatively low utilization of regions B, C, and D, (ii) high sparsity of activations, and (iii) low utilization of the 16-bit value range.

Accumulator. The SP of the accumulator is distributed over a much broader range than the multiplier output: 13%–50% and 15%–72% for bits 0–15 and bits 16–31, respectively. The high-order bits have a higher SP for two reasons: (i) the accumulator values are spread across a broad dynamic range of values, and (ii) the two's complement representation for negative values both increase the likelihood of ones in the most significant bits. In addition, regions B, C and D have lower SP relative to region A due to their lower utilization.

Gated clock. Figures 6(a) and 6(b) show the toggle rate of the accumulator and multiplier gated clock for ResNet-18 and ResNet-50, respectively. The gated clock toggle rate is governed by the sparsity of weights and activations in the DNN (i.e., whenever the weight or activation is zero, the clock is gated). While clock gating can help reduce energy consumption by the SA, it intensifies the BTI stress on the gated clock tree branch and may encourage asymmetric aging. Although region A has the highest toggle rate of nearly 40%, it is significantly less than the maximum toggle rate of a free-running clock (100%). This is explained by the extremely sparse activation, which encourages clock gating. The other regions have toggle rates within the range of 5%–27% because of lower utilization by the DNN model and extremely sparse activations.

3.3 Timing Analysis

The second phase of our experimental analysis involves detailed timing analysis using aging models. We analyze all logical paths in the SA and partition them into the following groups, as shown in Fig. 7:

1. A2A: The logical paths between the sampling register of the input activation and the neighbor activation register.

SA Element	Region	Bits	SP	SA utilization	Sparsity	Dynamic range
Activations	A	5–0	15%–19%		+	+
	A	7–6	<0.25%		+	+
Activations	B–D	5–0	<5%	+	+	+
	B–D	7–6	<0.05%	+	+	+
Weights	A	7–0	50%			
Weights	B–D	7–0	<15%	+		
Multiplier	A	0	<10%		+	+
	A	15–1	16%–20%		+	+
Multiplier	B–D	0	<7%	+	+	+
	B–D	15–1	<12%	+	+	+
Accumulator	A	15–0	30%–50%			+
	A	31–16	40%–72%			
Accumulator	B–D	15–0	13%–30%	+		+
	B–D	31–16	15%–40%	+		+
			Toggle rate			
Gated clock			30%–40%		+	
Gated clock			5%–27%	+	+	

Table 1: ResNet-18 summary of SP and gated clock toggle rate distribution with potential to asymmetric aging timing violations.

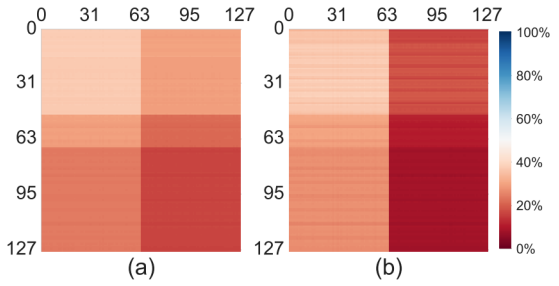

Figure 6: Accumulator and multiplier gated clock toggle rate: (a) Res-Net-18 and (b) ResNet-50.

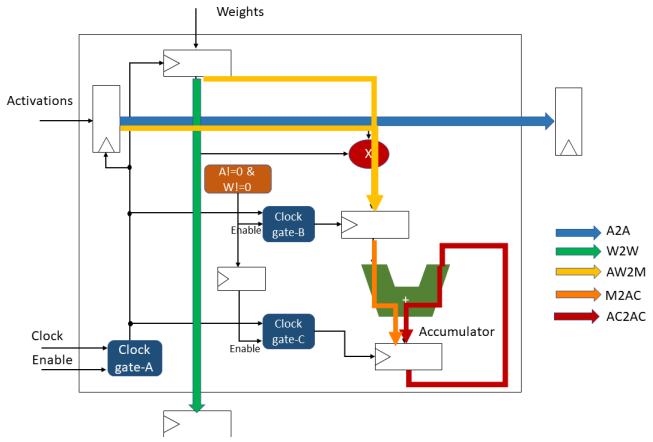

Figure 7: Timing path groups for PE timing.

2. W2W: The logical paths between the sampling register of the input weight and the neighbor cell weight register.
3. AW2M: The logical paths that start from the activation sampling register or the weight sampling register, propagate through the multiplier, and terminate at the multiplier sampling register.
4. M2AC: The logical paths that start from the multiplier sampling register, go through the adder, and end at the accumulator register.
5. AC2AC: The logical paths that start from the accumulator register, go through the adder, and return to the accumulator register.

Table 2 summarizes the detailed timing analysis results for the SA with asymmetric aging. The setup-timing analysis indicates that the path group from the 32-bit accumulator output to the accumulator input (AC2AC) is the most susceptible to BTI since it is the critical timing group of the SA. Table 2 shows that the AC2AC group experiences the highest degradation in worst negative slack (WNS) in all regions, dropping from 0 to −174 ps. Table 2 also shows that the number of setup-timing violations for the AC2AC group is in the range of 14 000 to 17 000 in every region. The M2AC group also experiences setup violations due to aging, but its WNS and the number of violating paths are less than those introduced by the AC2AC group. The remaining group paths do not exhibit any setup violations; however, since their WNS decreased, their resiliency is degraded.

Table 2 also presents the results of the hold-timing analysis of the SA. As opposed to setup-timing violations, which can be mitigated by reducing the SA clock frequency, hold-timing violations cannot be mitigated and thereby are even more severe than setup-timing violations. In the hold-timing analysis, asymmetric aging affects two opposing mechanisms. The following discussion summarizes our observations for each path group:

A2A: The A2A group incurs hold-timing violations in all regions with a WNS of −4 ps, with regions B and D having the highest number of hold-timing violations. Our timing analysis indicates that, despite the high utilization of region A, it also experiences hold-timing violations. This is attributed to the asymmetry between the accumulated wire delay and the logical-cell delay in certain paths, as discussed in Sec.

2.3. In addition, the timing violations in regions B and D are induced by the low utilization of these regions in conjunction with the asymmetry between the accumulated wire delay and the logical-cell delay. Our timing analysis indicates that all activation signals traversing from the boundary of regions A to B and C to D make additional contributions to the timing violations in these regions. This is due to the capture clock in regions B and D, which incurs a greater delay shift than the launch clock in regions A and C, resulting in hold-timing violations. The hold-timing violations in region C are also due to its low utilization and the asymmetry between logical and wire cell delays. Region C has fewer violations than region A because it has fewer rows. Our setup-timing analysis indicates that the A2A group incurs no setup-timing violations due to aging because it has a significant positive timing slack.

W2W: The W2W group also incurs hold-timing vio-lations in all regions with WNS in the range -2 to -1 ps. The asymmetry between the accumulated cell and wire delay causes hold-timing violations in all regions. These violations in regions B–D are also due to their low utilization. All weight signal crossing from regions A and B to regions C and D, respectively, encounter hold-timing violations. The low utilization of regions C and D creates a bigger delay shift in the capture clock with respect to the launch clock. The more numerous rows in regions A and B compared with regions C and D contributes to the increased number of hold-timing violations.

AW2M: The AW2M group has a hold WNS of -3 ps in all regions, where regions A and B have a greater number of hold-timing violations due to their more numerous rows. Our timing analysis indicates that hold-timing violations are induced by (i) activation sparsity and (ii) asymmetry between the accumulated wire delay and cell delay. In both cases, the capture clock incurs a larger delay shift, which results in hold-timing violations. The AW2M group presents no setup violations due to asymmetric aging, however, its positive timing slack is reduced in approximately 130 ps.

Logical path	Setup WNS [ps] before and after asymmetric aging				Hold WNS [ps] before and after asymmetric aging			
	Region A	Region B	Region C	Region D	Region A	Region B	Region C	Region D
W2W	2605/2598	2605/2598	2605/2598	2605/2599	0/−2	0/−4	0/−3	0/−3
	2576/2573	2576/2571	2576/2571	2576/2571	0/−1	0/−1	0/−2	0/−1
	1038/910	1038/906	1038/908	1038/905	0/−3	0/−3	0/−3	0/−3
M2AC	119/−32	119/−44	119/−38	119/−49	0/−1	0/−2	0/−2	0/−2
AC2AC	0/−155	0/−170	0/−162	0/−174	30 / 31	30 / 31	30 / 31	30 / 32
	Number of violated setup paths				Number of violated setup paths			
W2W	0	0	0	0	8606	9150	7680	12,000
	0	0	0	0	8606	8606	7872	7680
	0	0	0	0	4303	4303	3840	3840
M2AC	4303	4303	3840	3840	8606	47 333	34 560	42 240
	9822	12 909	11 520	11 520	0	0	0	0
Total (% of violating paths)	14125 (0.00014%)	17212 (0.00017%)	15360 (0.00015%)	15360 (0.00015%)	30121 (0.0003%)	69392 (0.0007%)	53952 (0.0005%)	65760 (0.00065%)

Table 2: Summary of the Worst Negative Slack (WNS) and total number of timing violations in the SA due to asymmetric aging.

M2AC: The M2AC path group has a hold WNS of -2 ps and the largest number of violating paths. In this group, both the launch clock and the capture clock are governed by the same control logic, so all clock buffers on the launch and capture clock branches age symmetrically. However, our timing analysis indicates that all regions incur hold-timing violations due to the asymmetry between the accumulated cell and wire delays. This asymmetry is emphasized by the high sparsity, which intensifies the aging on both the launch and capture clocks. In addition, the low utilization in regions B–D further encourages clock tree aging, resulting in an even greater number of violations than in region A. The M2AC group also presents setup-timing violations with a WNS of -49 ps. Our timing analysis indicates that this is attributed to the high sparsity on the accumulator path, which accelerates the timing degradation on the logical elements in the M2AC path.

AC2AC: The AC2AC path group is the longest path in the SA and therefore incurs the most severe setup violations due to the aging of the 32-bit ad-der. The setup violations are ascribed to (i) the low utilization of regions B–D and (ii) the lack of utili-zation of the full 32-bit dynamic range in all re-gions. Additionally, this path group has no hold-timing violation even when asymmetric aging is considered. In this case, both launch and capture paths of the clock tree are the same since the path begins and ends in the same register, and as a re-sult, they degrade symmetrically. In addition, the aging slows the logical path between the accumu-lator output to the accumulator input, so it con-tributes to improve hold margins.

4. FAULT ANALYSIS

Our fault injection experimental model examines how timing violations due to asymmetric aging affect the prediction accuracy of the DNN model. Violation of timing paths may cause flip-flops to transform into metastable states, resulting in bit flips. In severe cases, when the data consistently miss the boundaries of the flip-flop sampling window, they may manifest as persistent errors. The rate at which a metastable state is entered in a flip-flop when timing constraints are violated is [35]

$$Failure\ Rate = T_W F_C F_D e^{-S/\tau}, \quad (2)$$

where S is a pre-determined time for metastability resolution, F_C is the clock frequency, and F_D is the data transition rate. Both τ and T_W are intrinsic flip-flop circuit parameters that

Path group	Number of failure events in an inference
A2A	0.01
W2W	0.01
AW2M	853
M2AC	853
AC2AC	853

Table 3: Failure events per single flip-flop with timing violations in an inference for every path group.

represent the resolution time constant and the metastability window width, respectively. When plugging in the design parameters of our 28 nm SA into Eq. (2) and considering the resolution time available for every path group to resolve the failure events, we obtain the failure rate as summarized in Table 3 per a single flip flop with timing violation. When considering both setup- and hold-timing violations for our fault injection model, the overall number of flip-flop failure events can reach up to 190 million per single inference. We perform a sensitivity analysis of the DNN model prediction accuracy to the number of flip-flop failure events in every inference. In our sensitivity analysis, we increase the number of failure events in every inference in steps of 10 000 [0.00526% of the overall number of failure events predicted by Eq. (2)]. Additionally, the failure events are distributed randomly over all DNN layers, excluding the first and last layers. Within a model layer, all flip-flop failure events are randomly distributed over time. We run every image inference five times and calculate the average prediction accuracy. Figure 8(a) presents the sensitivity of ResNet-18 prediction accuracy when the fault injection model is considered. It can be observed that by considering less than 0.00001% of the overall failure events, the prediction accuracy of the model drops by 40%. In addition, when less than 0.00005% of the failure events are considered, the model prediction accuracy drops to nearly 0%. In addition, such faults can also have a major impact on DNN confidence level [46].

The next step in our fault injection analysis considers only hold timing violations, assuming that setup violations can be mitigated by reducing the clock frequency. The hold fault injection analysis is performed with a failure event distribution similar to the combined setup-and-hold analysis. In the case of hold-related faults, the overall number of flip-flop failure events can reach up to 127 million per single inference. In our analysis, we increase the number of failure events in every inference in steps of 10 000, which is 0.008% of the overall number of failure events predicted by Eq. (2). Figure 8(b) illustrates the sensitivity of ResNet-18 prediction accuracy when hold-related faults are considered. It can be observed that by considering less than 0.7% of the overall failure events, the prediction accuracy of the model drops by 40%. Such a significant reduction in prediction accuracy, even when setup violations are excluded and only a small portion of the overall hold failure events are considered, suggests that asymmetric aging can induce catastrophic functionality failures in SAs and DNN models. Therefore, developing mitigation techniques for asymmetric aging is crucial to maintain the resiliency of DNNs.

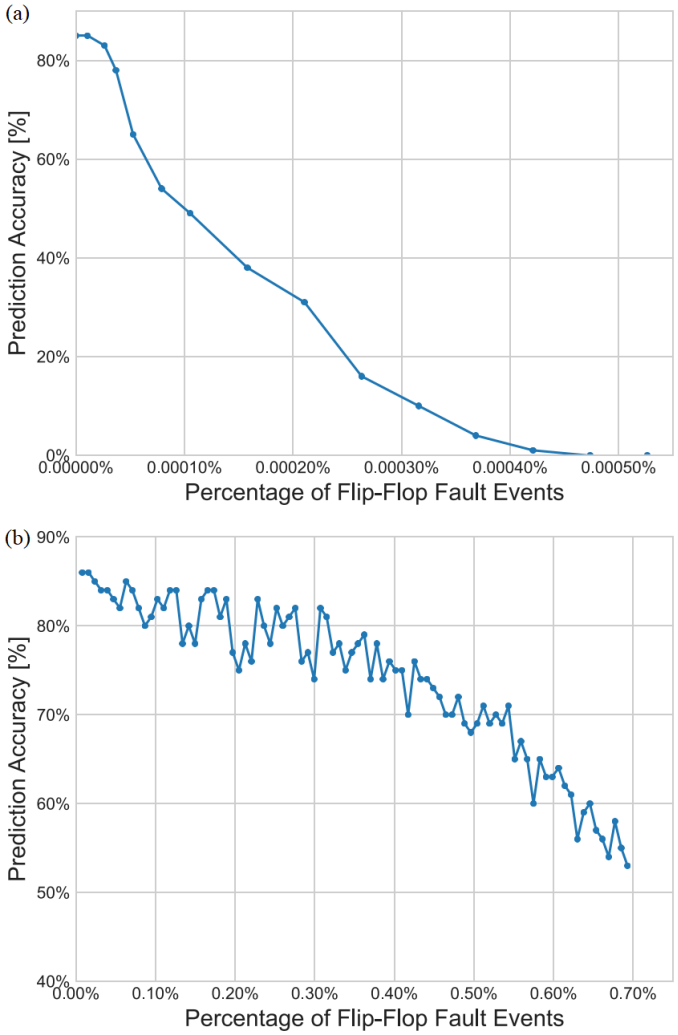

Figure 8: ResNet-18 Prediction accuracy with fault injection: (a) setup and hold faults and (b) hold faults.

5. STRATEGIES TO MITIGATE ASYMMETRIC AGING

We identify several approaches to mitigate asymmetric aging in ML accelerators. We demonstrate these techniques on SAs; however, they are applicable to ML accelerators in general. Our mitigation techniques are

1. introducing a new clock gate circuitry to alleviate asymmetric aging of clock buffers;
2. adding timing guard bands to the clock cycle time to mitigate setup violations;
3. using selective hold-timing violation fixes;
4. presenting a completed design flow for ML accelerators that integrates the flows and analysis described herein.

5.1 Symmetric Clock Gate

This study shows that clock gate circuitry promotes asymmetric aging on clock branches, resulting in severe timing violations. Figure 9(a) shows a common clock gate circuit, which consists of a latch and an AND gate. When the enable signal En is set to logical 1, the clock signal is allowed to propagate through the clock branch. However, when the enable signal En is set to logical 0, the clock path is maintained under a constant logical state of 0, which promotes BTI stress. To overcome the limitation of the common clock gate, we propose the symmetric clock gate circuit shown in Figure 9(b). In this clock gate, the logical state of the gated clock is controlled by the mode signal. When the mode is set to logical 0, the symmetric clock gate operates like the original clock gate, i.e., the logical state of the gated clock is 0. However, when the mode is set to logical 1 the gated clock state is logical 1. The proposed clock gate is free from static hazards, allowing the mode signal to be toggled at a low rate by the SA control logic. This ensures that, when the clock is gated, it spends nearly an equal amount of time in logical 1 as it does in logical 0.

Our aging profiling simulation shows that the utilization of the symmetric clock gate produces an SP of approximately 50% on the gated clock. In addition, the timing analysis of the SA with the symmetric clock gate is summarized and compared with the conventional clock gate in Table 4. The results show that the symmetric clock gate improves the hold WNS by 50% in most of the violated path groups. In addition, it reduces the number of hold-timing violations by 55%. The timing analysis also shows that the symmetric clock gate negligibly affects setup violations. Our synthesis and place-and-route analysis indicates that symmetric clock gates introduce an overhead of 1% on the total cell area, which can be absorbed by the implementation tools with no overhead to the overall floorplan area. In addition, the symmetric clock gate power overhead is nearly 0.09% of the total SA power.

5.2 Clock Cycle Time Guard Band

Overcoming setup-timing violations requires tightening the clock cycle time and considering aging degradation in timing closure. Our timing results, summarized in Table 4, indicate that the setup WNS is -174 ps, which can be mitigated by tightening the clock cycle time by 7%. Table 5 presents the results of our power analysis, which indicate that such a mitigation strategy introduces a 1.3%, 8%, and 7.25% increase in leakage power, dynamic power, and total power, respectively.

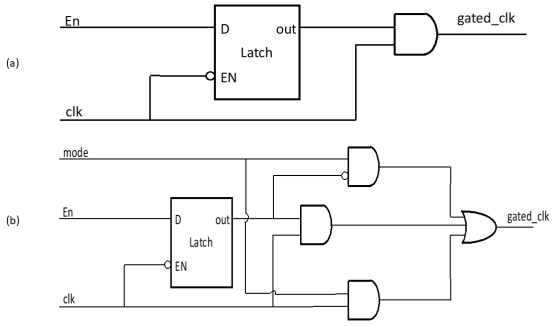

Figure 9: (a) A typical clock gate circuitry and (b) a Symmetric clock gate circuitry.

Path groups	WNS with conventional (Table 2) symmetric clock gate [ps]	
	Setup	Hold
A2A	2598/2569	$4/2$
W2W	2571/2573	$2/1$
AW2M	905/904	$3/2$
M2AC	$-49/-50$	$2/1$
AC2AC	$-174/-174$	31/31
	Number of violated paths with conventional (Table 2) symmetric clock gate [ps]	
	Setup	Hold
A2A	0 / 0	37436 / 32572 (-13%)
W2W	0 / 0	32764 / 16286 (-51%)
AW2M	0 / 0	16286 /16286 (0%)
M2AC	16286 / 16286	132739 / 32572 (-75%)
AC2AC	45771 / 45771	0 / 0 (0%)
Total	62057 / 62057	219225 / 97716 (-55%)

Table 4: Path groups failure rate per single flip-flop with timing violations.

SA	Leakage power	Dynamic power	Total power
Original	156.4 mW	1249 mW	1405.4 mW
With aging clock cycle guard band	158.51 mW (+1.3)%	1348.8 mW (+8%)	1507.3 mW (+7.25%)

Table 5: Power consumption of SA with aging guard band with respect to original SA.

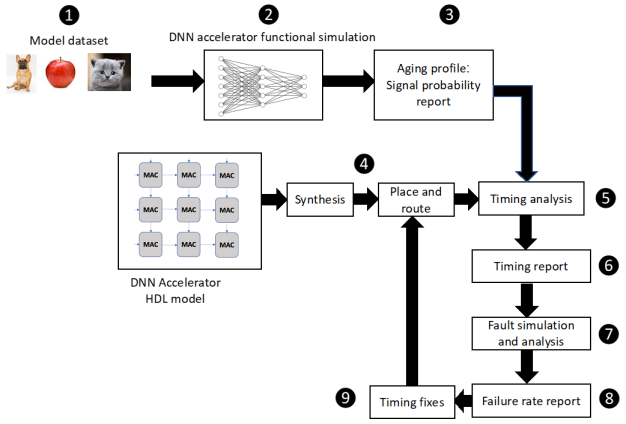

Figure 10: Asymmetric-aging-aware design flow for ML accelerators.

In addition, our SA area analysis indicates that this approach involves negligible area overhead since logical cells on the critical path are swapped with lower Vth cells that have a similar area footprint. An alternative approach for tightening the clock cycle is to compromise SA performance and reduce its clock frequency by 7%.

5.3 Selective Hold Timing Violation Fixes

The remaining timing violations after employing the previously described techniques are the hold-timing violations that are not solved by the symmetric clock gate. This time, we selectively fix hold-timing violations based on their contribution to the failure rate. Table 3 shows that both A2A and W2W failures occur at a relatively low rate. Additionally, our fault-injection simulations also indicate that such faults do not affect the DNN prediction accuracy. Therefore, we consider fixing only the AW2M and M2AC hold-timing violations. The remaining hold-timing violations are fixed by adding a delay buffer to the violated logical path. Our area and power analyses indicate that these remaining fixes incur 0.07% and 0.01% area and power overhead, respectively, with no impact on clock cycle time.

5.4 A Complete Design Flow

Finally, we summarize the complete design flow for ML accelerators, which integrates the flow and analysis described herein.

The full flow is depicted in Fig. 11 and consists of the following stages:

1. dataset preparation;
2. simulation of DNN accelerator on the related dataset.
3. aging profile produced by functional simula-tion consisting of SP measurement for the building blocks of the DNN accelerator;

	Power overhead	Area overhead
Symmetric clock gate	0.09%	1%
Clock cycle guard band	7.25%	0%
Selective hold-timing fixes.	0.01%	0.07%
Total	7.85%	1.07%

Table 6: Overhead of asymmetric aging mitigation.

4. synthesis and place-and-route of DNN accel-erator Hardware Description Language (HDL) model.
5. timing analysis combined with aging libraries and the aging profiles produced in stage 3;
6. generation of timing reports for all setup and hold timing violations;
7. fault injection analysis, which combines DNN accelerator functional simulation with fault injec-tions for the violated paths;
8. failure rate report, which details the impact of faults on the overall accuracy of the model;
9. timing fixes, which combine symmetric clock gating, clock cycle guard band, and selective fixes for hold-timing violations (the necessary timing fixes are then pushed to the place-and-route tool to be implemented in the design).

Stages 4–9 are repeated until the design is free from timing violations that affect model accuracy. Table 6 summarizes the overall power and area overhead for these techniques to mitigate asymmetric aging on the SA case study.

6. SUMMARY

This paper uses systolic arrays as a case study to comprehensively study how asymmetric aging affects ML accelerators. We demonstrate that asymmetric aging can cause major faults in DNNs, severely impacting their resiliency and decreasing their prediction accuracy, which can lead to functional safety violations in mission-critical systems. We develop herein a complete flow for simulating, analyzing, and mitigating asymmetric aging in ML accelerators, which encompasses (i) aging-profile extraction from functional simulation, (ii) a timing analysis with aging models, and (iii) a fault injection model to evaluate the DNN's performance under asymmetric-aging conditions.

Our analysis reveals four primary mechanisms that promote asymmetric aging in systolic arrays: (i) DNN sparsity, (ii) underutilization of the dynamic range for value representation, (iii) clock gating, and (iv) a lack of symmetry between logical-cell delays and wire delays. In addition, our analysis shows that the spatial location of PEs contributes significantly to the likelihood of incurring asymmetric-aging-related faults. We propose mitigation techniques that combine a novel symmetric clock gate circuitry, selective hold violation fixes, and clock cycle guard band adjustment. These techniques eliminate asymmetric-aging reliability concerns in SAs.

REFERENCES

[1] M. Alam and C. A. K. Roy, "Reliability- and process-variation aware design of integrated cir-cuits—a broader perspective," in *IEEE International Reliability Physics Symposium Proceedings*, 2011, https://doi.org/10.1109/IRPS.2011.5784500.

[2] M. Alam and S. Mahapatra, "A comprehensive model of pmos nbti degradation," *Microelectronics Reliability*, vol. 45, pp. 71–81, 2005, https://doi.org/10.1016/j.microrel.2004.03.019.

[3] S. Arasu, M. Nourani, F. Cano, J. Carulli, and V. Reddy, "Asymmetric aging of clock networks in power efficient designs," in *Proceedings - International Symposium on Quality Electronic Design*, 2014, pp. 484–489, https://doi.org/10.1109/ISQED.2014.6783365.

[4] K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, S. Nassif, E. Nowak, D. Pearson, and N. Rohrer, "High-performance cmos variability in the 65-nm regime and beyond," *Please give journal*, vol. Please give volume, p. please give pages, 2006.

[5] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbti effect for reliable design," in *Proceedings of the Custom Integrated Circuits Conference*, 2006, pp. 189–192, https://doi.org/10.1109/CICC.2006.320885.

[6] A. Calimera, M. Loghi, E. MacIi, and M. Poncino, "Aging effects of leakage optimizations for caches," in *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*, 2010, pp. 95–98, https://doi.org/10.1145/1785481.1785504.

[7] A. Calimera, M. Loghi, E. Macii, and M. Poncino, "Dynamic indexing: Leakage-aging co-optimization for caches," *IEEE Transactions on Computer-Aided Design of Integrated Cir-cuits and Systems*, vol. 33, pp. 251–264, 2014, https://doi.org/10.1109/TCAD.2013.2287187.

[8] S. Chakravarthi, A. Krishnan, V. Reddy, C. Machala, and S. Krish-nan, "A comprehensive framework for predictive modeling of negative bias temperature instability," in *IEEE International Reliability Physics Symposium Proceedings, Institute of Electrical and Electronics Engineers Inc.,*, 2004, pp. 273–282.

[9] Y.-H. Chen, J. Emer, and V. Sze, *Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks*. please give publisher, please give year.

[10] F. Gabbay and A. Mendelson, "Asymmetric aging effect on modern microprocessors," *Microelectronics Reliability*, vol. 119, p. please give pages, 2021, https://doi.org/10.1016/j.microrel.2021.114090.

[11] F. Gabbay, A. Mendelson, B. Salameh, and M. Ganaiem, "Aging avoidance eda tool," 2021.

[12] P. give author, *Systolic Arrays (for VLSI)*. please give publisher, 1979.

[13] G. Goos, E. Bertino, W. Gao, B. Steffen, G. Woeginger, R. Aachen, G. Aachen, and Y. Moti. Please give publisher, Please give year, http://www.springer.com/series/7408.

[14] N. Gundi, P. Pandey, S. Roy, and K. Chakraborty, "Implementing a timing error-resilient and energy-efficient near-threshold hardware accelerator for deep neural network infer-ence," *Journal of Low Power Electronics and Applications*, vol. 12, p. please give pages, 2022, https://doi.org/10.3390/jlpea12020032.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, http://arxiv.org/abs/1512.03385.

[16] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient infer-ence and training in neural networks," *http://arxiv.org/abs/2102.00554*, 2021.

[17] S. Holst, L. Bumun, and X. Wen, "Gpu-accelerated timing simulation of systolic-array-based ai accelerators," in *Proceedings of the Asian Test Symposium, IEEE Computer Socie-ty*, 2021, pp. 127–132, https://doi.org/10.1109/ATS52891.2021.00034.

[18] J. Huang, J. Chai, and S. Cho, "Deep learning in finance and banking: A literature review and classification," *Frontiers of Business Research in China*, vol. 14, 2020, https://doi.org/10.1186/s11782-020-00082-6.

[19] Y. Ibrahim, H. Wang, J. Liu, J. Wei, L. Chen, P. Rech, K. Adam, and G. Guo, "Soft errors in dnn accelerators: A comprehensive review," *Microelectronics Reliability*, vol. 115, p. please give pages, 2020, https://doi.org/10.1016/j.microrel.2020.113969.

[20] N. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-L. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. 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. Naga-rajan, 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, and D. Yoon, "In-datacenter performance analysis of a tensor processing unit tm," 2017.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton, *ImageNet Classification with Deep Convolutional Neural Net-works*. please give publisher, please give year.

[22] S. Kundu, S. Banerjee, A. Raha, S. Natarajan, and K. Basu, "Toward functional safety of systolic array-based deep learning hardware accelerators," *IEEE Trans Very Large Scale Integr VLSI Syst.*, vol. 29, pp. 485–498, 2021, https://doi.org/10.1109/TVLSI.2020.3048829.

[23] S. Kundu, A. Soyyiğit, K. Hoque, and K. Basu, "High-level modeling of manufacturing faults in deep neural net-work accelerators," *please give journal*, vol. please give volume, p. please give pages, 2020, https://doi.org/10.1109/IOLTS50870.2020.9159704.

[24] Kung, "Why systolic architecture?" 1982.

[25] G. Litjens, T. Kooi, B. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017, https://doi.org/10.1016/j.media.2017.07.005.

[26] D. Lopresti, "A systolic array for rapid string comparison," *please give journal*, vol. please give volume, p. please give pages, please gie year, https://www.researchgate.net/publication/243706366.

[27] M. Macas, C. Wu, and W. Fuertes, "A survey on deep learning for cybersecurity: Progress, challenges, and opportunities," *Computer Networks*, vol. 212, 2016, https://doi.org/10.1016/j.comnet.2022.109032.

[28] R. Machupalli, M. Hossain, and M. Mandal, "Review of asic accelerators for deep neural network," *Microprocess Microsyst*, vol. 89, p. please give pages, 2022, https://doi.org/10.1016/j.micpro.2022.104441.

[29] S. Mittal, "A survey on modeling and improving reliability of dnn algorithms and accelerators," *Journal of Systems Architecture*, vol. 104, p. please give pages, 2020, https://doi.org/10.1016/j.sysarc.2019.101689.

[30] V. Nair and G. Hinton, *Rectified Linear Units Improve Restricted Boltzmann Machines*. please give publisher, please give year.

[31] S. Naseh, M. Deen, and C. Chen, "Hot-carrier reliability of submicron nmosfets and integrated nmos low noise amplifiers," *Microelectronics Reliability*, vol. 46, pp. 201–212, 2006, https://doi.org/10.1016/j.microrel.2005.04.009.

[32] P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "Greentpu: Improving timing error resilience of a near-threshold tensor processing unit," 2019, https://doi.org/10.1145/3316781.3317835.

[33] P. Pandey, N. Gundi, P. Basu, T. Shabanian, M. Patrick, K. Chakraborty, and S. Roy, "Challenges and opportunities in near-threshold dnn accelerators around timing errors," *Journal of Low Power Electronics and Applications*, vol. 10, pp. 1–19, 2020, https://doi.org/10.3390/jlpea10040033.

[34] please give authors, "please give title," in *Institute of Electrical and Electronics Engineers, Proceedings, 2020 26th IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS): IOLTS 2020: July 13-16, 2020, Virtual edition*, 2020.

[35] C. Portmann and T. Meng, "Metastability in cmos library elements in reduced supply and technology scaled applications," *IEEE J Solid-State Circuits*, vol. 30, pp. 39–46, 1995, https://doi.org/10.1109/4.350196.

[36] R. Ravindran, M. Santora, and M. Jamali, "Multi-object detection and tracking, based on dnn, for auton-omous vehicles: A review," *IEEE Sens. J.*, vol. 21, pp. 5668–5677, 2021, https://doi.org/10.1109/JSEN.2020.3041615.

[37] V. Reddy, J. Carulli, A. Krishnan, W. Bosch, and B. Burgess, "Impact of negative bias temperature instability on product para-metric drift," in *Proceedings - International Test Conference*, 2004, pp. 148–155, https://doi.org/10.1109/test.2004.1386947.

[38] S. Salamin, G. Zervakis, O. Spantidi, I. Anagnostopoulos, J. Hen-kel, and H. Amrouch, "Reliability-aware quantization for anti-aging npus," in *2021 Design, Automation Test in Europe Conference Exhibition (DATE), Grenoble, France*, 2021, pp. 1460–1465,

https://doi.org/10.23919/DATE51398.2021.9474094.

[39] D. Schroder, "Negative bias temperature instability: What do we understand?" *Microelectronics Reliability*, vol. 47, pp. 841–852, 2007, https://doi.org/10.1016/J.MICROREL.2006.10.006.

[40] N. Srinivasan, N. Prakash, S. D., S. D., S. S. L. G., and B. Sundari, "Power reduction by clock gating technique," *Procedia Technology*, vol. 21, pp. 631–635, 2015, https://doi.org/10.1016/j.protcy.2015.10.075.

[41] S. Subathradevi and C. Vennila, "Systolic array multiplier for augmenting data center networks communication link," *, Cluster Comput.*, vol. 22, pp. 13 773–13 783, 2019, https://doi.org/10.1007/s10586-018-2092-4.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going Deeper with Convolutions*. Please give publisher, Please give year.

[43] E. Talpes, A. Gorti, G. Sachdev, D. das Sarma, G. Venkata-ramanan, P. Bannon, B. McGee, B. Floering, A. Jalote, C. Hsiong, and S. Arora, "Compute solution for tesla's full self-driving computer," *IEEE Micro.*, vol. 40, pp. 25–35, 2020, https://doi.org/10.1109/MM.2020.2975764.

[44] S. v Kumar, C. Kim, and S. Sapatnekar, "An analytical model for negative bias temperature instability," *Please give journal*, vol. Please give volume, p. please give pages, please give year.

[45] J. Velamala, V. Ravi, and Y. Cao, "Failure diagnosis of asymmetric aging under nbti," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2011, pp. 428–433, https://doi.org/10.1109/ICCAD.2011.6105364.

[46] R. Yazdani, M. Riera, J. M. Arnau, and A. González, "The dark side of dnn pruning," in *The Dark Side of DNN Pruning, 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA*, 2018, pp. 790–801, https://doi.org/10.1109/ISCA.2018.00071.

[47] J. Zhang, T. Gu, K. Basu, and S. Garg, "Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator," *http://arxiv.org/abs/1802.04657*, 2018.

[48] J. Zhang, K. Basu, and S. Garg, "Fault-tolerant systolic array based accelerators for deep neural network execution," *IEEE Des Test.*, vol. 36, pp. 44–53, 2019, https://doi.org/10.1109/MDAT.2019.2915656.