# Spiking Transformer Hardware Accelerators in 3D Integration

Boxun Xu[1], Junyoung Hwang[2], Pruek Vanna-iampikul[2, 3], Sung Kyu Lim[2], Peng Li[1*]
{boxunxu,lip}@ucsb.edu,{jyh,v.pruek,limsk}@gatech.edu
[1]Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, USA
[2]Department of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA
[3]Department of Electrical Engineering, Burapha University, Chonburi, Thailand

## ABSTRACT

Spiking neural networks (SNNs) are powerful models of spatiotemporal computation and are well suited for deployment on resource-constrained edge devices and neuromorphic hardware due to their low power consumption. Leveraging attention mechanisms similar to those found in their artificial neural network counterparts, recently emerged spiking transformers have showcased promising performance and efficiency by capitalizing on the binary nature of spiking operations. Recognizing the current lack of dedicated hardware support for spiking transformers, this paper presents the first work on 3D spiking transformer hardware architecture and design methodology. We present an architecture and physical design co-optimization approach tailored specifically for spiking transformers. Through memory-on-logic and logic-on-logic stacking enabled by 3D integration, we demonstrate significant energy and delay improvements compared to conventional 2D CMOS integration.

## KEYWORDS

Spiking neural networks, Spiking transformers, HW/SW Co-Design, F2F Bonding, 3D integration

**ACM Reference Format:**
Boxun Xu[1], Junyoung Hwang[2], Pruek Vanna-iampikul[2, 3], Sung Kyu Lim[2], Peng Li[1*]. 2024. Spiking Transformer Hardware Accelerators in 3D Integration. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3676536.3676826

## 1 INTRODUCTION

Transformer models have significantly advanced model capabilities in language modeling and computer vision, and have found widespread adoption across various application domains [8, 24]. At the heart of these models lies a self-attention mechanism, which captures rich contextual information by considering all elements in a long input sequence, blending global and local sequence details into a unified representation.

Spiking neural networks (SNNs) are more biologically plausible than their non-spiking artificial neural network (ANN) counterparts [11]. Notably, SNNs can harness powerful temporal coding, facilitate spatiotemporal computation based on binary activations, and achieve ultra-low energy dissipation on dedicated neuromorphic hardware [1, 5, 16]. Recent spiking transformers showcased promising performance and efficiency by capitalizing on the binary nature of spiking activation [28, 30, 33, 34].

However, there is a current lack of dedicated hardware architectures for spiking transformers [28, 30, 33, 34]. Our goal is to fill this gap by developing optimized architectures capable of accelerating spatiotemporal spiking workloads for spiking transformers using 3D integration as a technology enabler. We see many opportunities that 3D integration can offer to enable biologically-inspired spiking transformers. Firstly, memory-on-logic stacking capability in 3D configurations allows for the storage of a significant portion of model parameters within local memory, ensuring swift and parallel memory access. Secondly, logic-on-logic stacking in 3D opens avenues for significant enhancements in energy efficiency, particularly in spike delivery management within SNN architecture. Ultimately, in a longer run, the ultra-dense neuron-to-neuron connectivity enabled by 3D integration promises improvements in SNN learning accuracy and efficiency, thereby propelling semiconductor chip emulation closer to the capabilities of the human brain.

**Challenges and Contributions** In this work, we adopt face-to-face(F2F)-bonded 3D integration technology to enable dedicated spiking transformer accelerators with memory-on-logic and logic-on-logic configurations.

Contribution 1: We propose **the first** dedicated 3D accelerator architecture for spiking transformers, which explore spatial and temporal weight reuse to support spike-based computation in transformer models .

Contribution 2: We enable **the first** 3D memory-on-logic and logic-on-logic interconnection schemes to significantly minimize energy consumption and latency, whereby delivering highly-efficient spiking neural computing systems with low area overhead.

Compared to 2D CMOS integration, the 3D accelerator offers substantial improvements. For the spiking MLP workload, it provides a 7.0% increase in effective frequency, 50% area reduction, and reductions of 7.8% in power consumption, 68.3% in memory access latency, and 69.5% in memory access power. For the spiking self-attention workload, the enhancements include a 6.3% increase in effective frequency, 50% area reduction, and reductions of 1.5% in power consumption, 74.2% in memory access latency, and 49.3% in memory access power.

## 2 BACKGROUND

### 2.1 Spiking Neural Networks

**LIF and IF Models.** The Leaky-Integrate-and-Fire(LIF) neuronal model is widely adopted in SNNs [11], which has the following


*ICCAD '24, October 27–31, 2024, New York, NY, USA*

ACM ISBN 979-8-4007-1077-3/24/10
https://doi.org/10.1145/3676536.3676826

*Corresponding author.

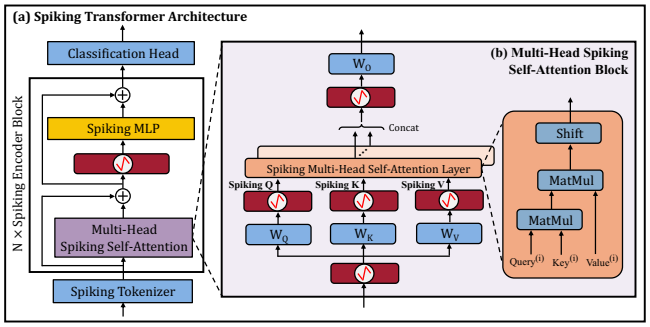


**Figure 1: (a) Model Architecture of spiking transformers. (b) Multi-head spiking self-attention block within each spiking encoder block of spiking transformers.**

discretized dynamics over time:

$$V_i\,[t_k] = V_i\,[t_{k-1}] + \sum_{j \in RF} w_{ji} S_j[t_k] - V_{leak} \tag{1}$$

$$S_i\,[t_k] = \begin{cases} 1 & \text{if } V_i[t_k] > V_{th} \rightarrow V_i[t_k] = 0 \\ 0 & \text{else } \rightarrow V_i[t_k] = V_i[t_k] \end{cases} \tag{2}$$

The Integrate-and-Fire (IF) spiking neuron model, as a simplified spiking neuronal model, is commonly used in SNNs which are converted from a pretained ANN [3, 18]. The dynamics of the IF model is obtained by setting $V_{leak}$ in Equ. 1 to zero.

**Spiking Attention.**The spiking attention mechanism is proposed in [33] as the basic component of various variants of spiking-based transformers [27, 30]. The binary queries(Q) and keys(K) are correlated at each time point to compute the dependencies between tokens in attention maps; the binary values(V) are computed to reflect the attention-weighted accumulation for each token. Fig. 1 illustrates the architecture of spiking transformers.

## 2.2 Neuromorphic Hardware

**Neuromorphic inference/training accelerators.** There exist various neuromorphic accelerators for SNN inference on the level of devices[25], circuits[21], micro-architectures [14], architectures and on-chip communication networks [5, 6, 15, 16, 22]; Some neuromorphic accelerators have been proposed for efficient SNN training [20, 26, 29]. However, these accelerators are mainly tailored for spiking CNNs such as spiking AlexNet and ResNet [9, 19, 31]. Although relevant to the acceleration of generic SNNs, these architectures are not optimized for large spiking transformers.

**3D Neuromorphic hardware.** Existing studies on the 3D IC realization of spiking neural networks have primarily employed monolithic 3D (M3D)[13] and face-to-face (F2F) bonding techniques[12], which are rooted in traditional liquid state machines (LSM)-based architectures. While these works have adapted the M3D or F2F design methodologies to enhance power-performance-area (PPA) metrics, they exhibit compatibility issues with current spiking transformers and fail to provide optimized support for the latest advancements in spiking transformers, which have demonstrated competitive performance. Furthermore, although the concept of memory-on-logic has been adopted in these work[12, 13], they face limitations due to the restricted number of neurons, thereby constraining the potential for dataflow optimization. These limitations underscore the need for a more comprehensive approach that supports both logic-on-logic and memory-on-logic configurations, specifically tailored to enhance the functionality of spiking transformers.

# 3 PROPOSED 3D ARCHITECTURE DESIGN

We introduce our proposed 3D dedicated architecture tailored for modern spiking transformers, which aims to enhance area utilization, energy efficiency, and processing speed.

In Section 3.1, we propose dedicated architecture support for managing the workloads of spiking MLP layers, which are key computational bottlenecks of transformers. Our proposed 3D dataflow minimizes data movement inherent in spiking transformers and maximizes weight reuse across tokens and timesteps, as well as the reuse of spiking activities across output features in MLP layers. Additionally, we design a dedicated systolic array that supports this optimized dataflow. In Section 3.2, we introduce a dedicated 3D architecture tailored for the workloads of spiking attention layers, the other bottleneck of transformer models. To address the challenges of heavy data movement associated with spiking attention maps, a kernel fusion strategy is employed to mitigate the need for extensive storage and data movement. In Section 3.3, we introduce a reconfigurable spiking self-attention array designed to flexibly handle various workloads of spiking attention layers, including key operations such as attention score computation $A = QK^T$ and $X = AV$. Our architecture and dataflow fully utilize fetched spiking query/key/value (Q/K/V) and computed spiking attention score (A) data via maximized data reuse during execution. This approach not only minimizes area overhead but also reduces data movement, significantly enhancing both the efficiency and flexibility of the system.

## 3.1 3D Acceleration for Spiking MLP layers

*3.1.1 Workload Processing in MLP Layers.* MLP (linear) layers in spiking transformers encompass three core processing steps: ① synaptic integration, ② membrane potential accumulation, and ③ spike generation, as detailed in Equ. 1 and Equ. 2.

The step of ① is to process a pre-synaptic activation $S_{in}$ with a shape of $\mathbb{R}^{N \times T \times D_{in}}$, using a pre-trained weight $W$ of shape $\mathbb{R}^{D_{in} \times D_{out}}$. Here, $N$ denotes the number of spiking tokens; $T$ represents the number of timesteps on which the model executes; $D_{in}$ and $D_{out}$ denote the number of input features and output features, respectively. As in typical SNNs [9], $W$ can be shared for processing all-or-none input spikes across multiple timesteps, based on which [15, 16] have proposed methods for executing such temporal workload in parallel. Unlike traditional spiking neural networks, spiking transformers are unique in the sense that weights can be further shared when processing different tokens from $S_{in}$ to gain additional benefits. While offering an outstanding opportunity, processing such complex workload both spatially and temporally is nontrivial, and requires a customized systematic design to exploit the potential data reuse within the model.

The processing step of ② is to sequentially accumulate the computed synaptic integration of each neuron $i$ for each token $n$ at timestep $t$, denoted by $X_{n,i}[t]$, onto the membrane potential from the previous timestep, $V_{n,i}[t-1]$, to compute $V_{n,i}[t]$. Following

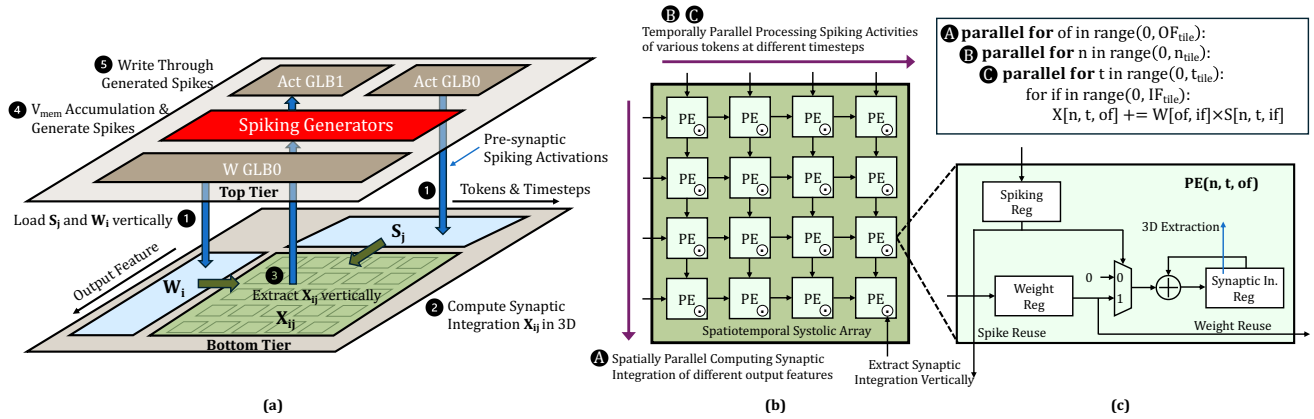


**Figure 2: Proposed 3D Architecture for processing spiking MLP layers: (a) 3D partitioning and dataflow, (b) systolic PE array for synaptic integration on the bottom tier, and (c) PE design.**

**Algorithm 1** Kernel Fusion for Spiking MLP Layers.

**Input:** the number of token tiles $\#tile_N$, the number of time steps $\#tile_T$, the number of input features $\#tile_{IF}$, the number of output features $\#tile_{OF}$, the height/width of the systolic array $H/W$, spiking activation $S_{in} \in \mathbb{R}^{N\times T\times D_{in}}$, weight $W \in \mathbb{R}^{D_{in}\times D_{out}}$.
**Output:** spiking output $S_{out} \in R^{N\times T\times D_{out}}$.
**for** $of = 0$ to $\#tile_{OF} - 1$ **do**
  **for** $n = 0$ to $\#tile_N - 1$ **do**
    **for** $t = 0$ to $\#tile_T - 1$ **do**
      **for** $if = 0$ to $\#tile_{IF} - 1$ **do**
        **if** $W(of, if)$ not in W buf. **then**
          **Load** $W(of, if)$ chunk from W GLB into W buf.
        **end if**
        **Load** $S_{in}(n, t, if)$ chunk from Act GLB0 into Act buf.
        **Compute** synaptic integration chunk $X(n, t, of)$
      **end for**
      **Extract** $X(n, t, of)$ to **Compute** $S_{out}(n, t, of)$ and **Write through** Act GLB1.
    **end for**
  **end for**
**end for**

this, step ③ performs conditional spike generation at each timestep as outlined in Equ. 2. This is done by comparing the current membrane potential $V_{n,i}[t]$ with the broadcasted voltage threshold $V_{th}$, determining whether an output spike shall be generated at each timestep.

Previous designs of neuromorphic accelerators typically provide limited parallelism, either in temporal or spatial dimensions, and are not tailored for spiking transformers. Additionally, synaptic integration is hampered by a serial register readout chain, which introduces significant delays, or by complex wire routing, which complicates the extraction of computed integrated synaptic input at a given time point and introduces high energy consumption.

*3.1.2 Proposed 3D MLP Layer Architecture and Dataflow.* In our proposed 3D integration with two silicon tiers as shown in Fig. 2, we have a dedicated core systolic PE array core at the bottom tier to execute spiking kernel operation ①, and another dedicated Spiking Generator core at the top tier for the spiking kernel operation of ②+③, respectively. As shown in the Alg. 1, we adopt an efficient tiling scheme to enable kernel fusion of the above two spiking kernel operations. Each tile indexed by $of$, $n$, $t$ is loaded and processed sequentially; the computation between $W$ and $S_{in}$ tiles and spike generation computation of $S_{out}$ are operated in parallel.

Due to the significantly higher energy consumption and latency associated with data memory access compared to computation, most accelerators are designed to maximize the utilization of accessed data and enhance parallel computation[16]. For spiking transformers given that a weight matrix is invariant with respect to various corresponding tokens and timesteps in spiking MLP layers, it is feasible to implement weight reuse strategies for different tokens across different timesteps. Similarly, input spiking activities for a particular input feature can be reused across neurons that produce different output features. Enabling parallel execution across these three dimensions on hardware improves throughput without increasing data loading overhead.

As illustrated in Fig. 2(a), the global memory buffers and spiking generators are placed on the top tier; the local buffers and systolic array are placed on the bottom tier. We design the following optimized dataflow. In step ❶, a pre-synaptic spiking activation tile, $S_{in}$, and a weight tile, $W$ are vertically loaded from the global buffers Act GLB0 and W GLB at the top tier to the $S$ and $W$ buffer at the bottom tier. In step ❷, synaptic integration of $OF_{tile}$ neurons for $n_{tile}$ tokens at $t_{tile}$ timesteps is computed within the spatiotemporal systolic array located at the bottom. In step ❸, the computed synaptic integration is extracted vertically and fed into the spiking generators based on an appropriate time index. In step ❹, the spiking generators compute the membrane potential and conditionally generate postsynaptic output spikes; in step ❺, the spiking generators write through the generated spikes to the global buffer Act GLB1 at each timestep.

The bottom tier, shown in Fig. 2(b), features a dense systolic array with 2D mapping of PEs. To leverage data reuse opportunities, spiking activities of different $n_{tile}$ tokens across different

$t_{tile}$ timesteps are processed by PEs in different columns, and the spiking activities of different $OF_{tile}$ output features are handled by PEs in different rows. Between left-right-connected PEs, multi-bit weights across $OF_{tile}$ output features are propagated from left to right, being reused across different tokens and timesteps. Meanwhile, the input spiking activities $S_{in}$ are reused by different output neurons, by propagation from top to bottom.

Each PE, designed to be synaptic integration-stationary as shown in Fig. 2(c), contains three registers serving as scratchpad memory. The registers store 1-bit input spiking activity, multi-bit weight, and multi-bit synaptic integration output, respectively. The accumulation of weight at the $if$-th input feature takes place only if the corresponding input spike is active. The synaptic integration output registers across the array are directly connected to the spiking generators at the top tier, leveraging the 3D extraction readout ports and high-density vertical wiring between the two silicon tiers.

## 3.2 3D Acceleration for Spiking Self-Attention Layers

The computation of spiking self-attention layers is another bottleneck and encompasses several key operations: ① the computation of spiking attention maps ($A = QK^T$), ② attention-weighted synaptic integration ($X = AV$), which provides inputs to a set of LIF neurons for generating the final binary spike-based attention output, ③ membrane potential accumulation of these LIF neurons, and ④ conditional generation of the LIF neuron output spikes as the final attention output. In operation ①, the spiking query $Q$ and spiking key $K$, initially shaped as $\mathbb{R}^{T\times N\times D_{in}}$, are subdivided into $\mathbb{R}^{T\times N\times H\times d}$. Here, $T$ represents the number of timesteps; N denotes the number of tokens; $H$ and $d$ indicate the number of self-attention heads and the number of features per head, respectively. A spiking attention map $S \in \mathbb{R}^{T\times H\times N\times N}$ is computed for each head at each timestep. For instance, the spiking attention map at $t$-th timestep for $h$-th self-attention head results from the binary matrix multiplication of the spiking query and key at the specific head and timestep. In ②, the attention-weighted synaptic integration is executed for each head at each timestep. The spiking attention map $A$, serving as the attention weights, is combined with the spiking value $V$, shaped in $\mathbb{R}^{T\times N\times H\times d}$ to compute attention-weighted synaptic integration, denoted by $X$ shaped as $\mathbb{R}^{T\times N\times H\times d}$.

Fig. 3 illustrates the dataflow for 3D integration based acceleration of spiking self-attention computation. In step ❶, the partitioned spiking query $Q_i$ ($i$-th token) and key $K_j$ ($j$-th token) are vertically loaded from the global buffer activation GLB at the top tier to the Q buffers and K buffers at the bottom tier, respectively. In step ❷, $Q_i$ and $K_j$ stream through the spiking self-attention array, detailed in Section 3.3, where the multi-bit spiking attention map, $A_{ij}$, mapped to a submatrix of the whole spiking attention map $A$, is accumulated in an attention-stationary manner. Once this step is completed, for computing $X = AV$, in step ❸ , the spiking value $V_j$ and partial synaptic integration $X_i$ vertically loaded into the V buffer and X buffer at the bottom tier are used to update synaptic integration. The attention-weighted $V$, that is the multiplication of $A_{ij}$ and $V_j$, is accumulated onto partial synaptic integration $X_i$. After the synaptic integration $X_i$ is computed, in step ❹, it is written back to synaptic integration global buffer X GLB. In step ❺ and step ❻, the spiking generators are activated to temporally accumulate the synaptic integration onto the membrane potential of each LIF neuron, and conditionally generate LIF neurons' spike outs, and write through the generated spikes to global buffer Act GLB 1, as the final spike-based outputs of spiking self-attention layers.

**Algorithm 2** Kernel Fusion for Spiking Attention

**Input:** the number of token tiles $\#tile_{N_k}$ and $\#tile_{N_q}$, the number of input features $\#tile_{IF}$, spiking query activation $Q \in \mathbb{R}^{T\times N\times H\times d}$, spiking key activation $K \in \mathbb{R}^{T\times N\times H\times d}$, spiking value activation $V \in \mathbb{R}^{T\times N\times H\times d}$.
**Output:** spiking output $S_{out} \in \mathbb{R}^{T\times N\times H\times d}$.
**for** $h = 0$ to $H-1$ **do**
  **for** $t = 0$ to $T-1$ **do**
    **for** $i = 0$ to $\#tile_{N_k} - 1$ **do**
      **Load** $K(h,t,i)$ and $V(h,t,i)$ from ActGLB0 to K/V buf.
      **for** $j = 0$ to $\#tile_{N_q} - 1$ **do**
        **Load** $Q(h,t,j)$ from Act GLB0 to Q buffer.
        **Compute** $A(h,t,i,j)$ within the reconfigurable array.
        **Load** partial synaptic integration $X(h,t,j)$ from X GLB to X buf.
        **Compute** $X(h,t,j) = X(h,t,j) + A(t,i,j) \times V(t,i)$.
        **Extract** partial synaptic integration $X(h,t,j)$ to X GLB.
      **end for**
    **end for**
    **Compute** $S_{out}(h,t)$ and **write** Act GLB1.
  **end for**
**end for**

The space complexity of attention map is $O(H \times N^2)$, which can be greater than the size of weight data with a space complexity of $O(D^2)$, especially when dealing with multiple-framed videos and long linguistic contexts, where the number of tokens $N$ can be much greater than $D$. Furthermore, the size of attention maps quadratically depends on $N$. Thus, reducing data movements of such huge attention maps is essential for efficient processing [4].

We present a kernel fusion dataflow in spiking transformers, adapting from [4], for this purpose as illustrated in Alg. 2. In operation ①, a given spiking query token is reused while processing different spiking key tokens. Meanwhile, a spiking key token is also reused when processing different spiking query tokens. Similarly, in operation ②, each computed spiking attention element is reused when processing spiking values for different output features, and spiking values of a given output feature are reused while processing different spiking attention elements. Then, we design the dense array to enable the aforementioned two parallel processing schemes and two data reuse schemes for both operation ① and operation ②.

## 3.3 Reconfigurable Attention Array

To optimize dataflow and minimize data movement, we propose a reconfigurable spiking self-attention array that supports flexible matrix multiplication operations of $A = QK^T$ and $X = AV$. This design enables a flexible dataflow and reduces the frequency and volume of data movements associated with large, multi-bit attention matrices.

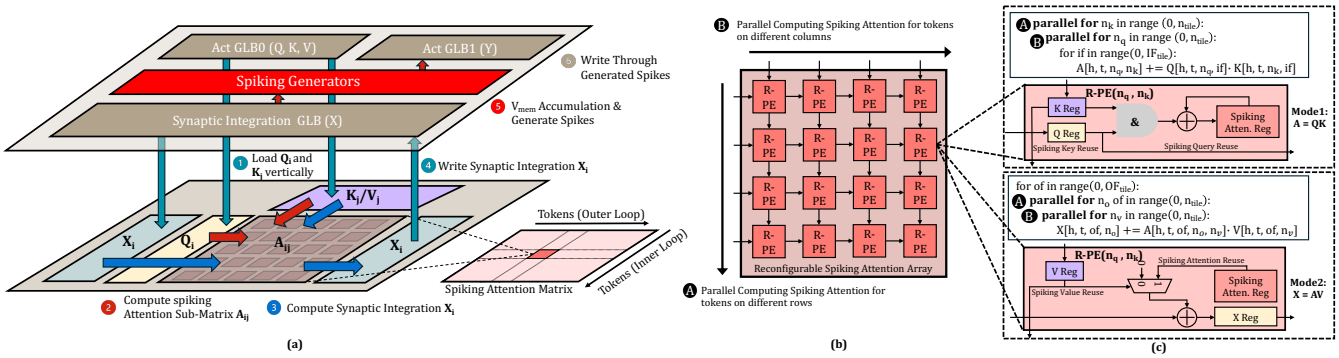


**Figure 3: Proposed 3D Architecture for processing spiking attention layers: (a) 3D partitioning, (b) proposed reconfigurable dense systolic array on the bottom tier, and (c) reconfigurable PE design supporting two different computations.**

The proposed reconfigurable spiking self-attention array involves two modes.

**Mode 1**: Each reconfigurable Processing Element (R-PE) computes one element of the spiking attention matrix. The spiking query is loaded from left to right, while the spiking key streams from top to down. The bit multiplication of spiking queries and keys is performed using a single two-input AND gate, with the results being accumulated onto a spiking attention register. Once computed, the attention map is stored within the array.

**Mode 2**: Each R-PE computes attention-weighted spiking value. If a propagated spiking value is active, the stored attention in the R-PE will be accumulated onto synaptic integration. This partial synaptic integration is propagated from left to right, and the spiking value V is propagated from top to bottom, with the synaptic integration streaming out from the right boundary of the systolic R-PE array.

Each R-PE involves two 1-bit registers for storing Q and K/V, and two multi-bit registers to store spiking attention $A_{n_q,n_k}$ and synaptic integration, respectively. Due to the binary nature of the spiking query, key, and value, we optimize the bitwidth to reduce redundancy by eliminating unnecessary high bit resolutions. The resolution required for all positive attention maps depends on the number of input features per head, necessitating a maximum of $log_2(d)+1$ bits. Additionally, the resolution for synaptic integration is determined by both the maximum number of input features and tokens, requiring up to $log_2(d) + log_2(N) + 2$ bits. For an 8-head 128-feature spiking transformer with 128-token inputs, the bitwidth requirement of the attention map is $log_2(128/8) + 1 = 5$ bits, and the bitwidth requirement of synaptic integration is 10 bits.

# 4 PROPOSED 3D PHYSICAL DESIGN

## 4.1 Memory on Logic

To minimize the latency and energy consumption between memory and computing modules in spiking MLP layers and spiking attention layers of spiking transformers, we employ a memory-on-logic 3D stacking for 3D accelerators. In the spiking MLP accelerators, we group the spiking generators, spiking activation global buffers(Act GLB), and weight global buffer(W GLB) as a memory die on the top. The remaining components are organized as a logic die at the bottom. Similarly, in spiking self-attention accelerators, we group the spiking generators, activation global buffers(Act GLB), and synaptic integration global buffer (X GLB) as memory die, with other components placed on the logic die at the bottom. This configuration ensures balanced cell utilization between the top and bottom dies, thereby reducing latency and energy consumption of memory accesses to speed up the overall computation.

In the memory die of both accelerators, the activation global buffers, weight global buffer, and synaptic integration global buffer are implemented using SRAM to ensure its high density and compact footprint. Meanwhile, the remaining spiking generators are synthesized using the logic gates. In the logic die, the buffers adjacent to the systolic array also utilize SRAM to optimize space and efficiency, while other components are synthesized using logic gates during the logic synthesis phase. The placement of the SRAM modules is strategically pre-determined based on the data flow connections detailed in Section 5.2.1 for both MLP and attention layers. This strategic placement is designed to maximize the 3D connectivity between the memory and logic dies, enhancing both data transfer efficiency and overall system performance.

## 4.2 Logic on Logic

Unlike memory-on-logic stacking, logic-on-logic stacking provides enhanced flexibility in design space by allowing cell movement of standard cells on both dies. This flexibility supports various types of tier partitioning where the memory and logic areas are unbalanced. In the logic-on-logic stacking, we group the activation global buffers, weight global buffers, and spiking generators on the top, while the remaining compute logics are placed at the bottom. Similarly to the memory on logic stacking, we use SRAM for both activation global buffer and weight global buffer for high-density memory storage, and the remaining cells are synthesized with the combinational circuit to represent their functionality defined in the SystemVerilog. Therefore, the logic cells are placed alongside the memory macros and buffers are inserted in both top and bottom dies. The logic-on-logic stacking enables the spiking generators to be connected with PEs with synaptic integration systolic array, and the spiking generators can extract the synaptic integration from bottom to top.

**Table 1: The overall performance comparisons between 2D and 3D design of spiking MLP(linear) accelerator across different array size and bitwidth**

| Array size H×W, weight/synaptic integration bitwidth | 16 × 128, 8b/16b | | 64 × 16, 8b/16b | | 64 × 16, 4b/12b | |
|---|---|---|---|---|---|---|
| | 2D | 3D | 2D | 3D | 2D | 3D |
| Effective Frequency ($GHz$) | 1.57 | **1.68** | 1.68 | **1.79** | 1.76 | **1.85** |
| Area Footprint ($mm^2$) | 0.45×0.9 | **0.45×0.45** | 0.45×0.78 | **0.45×0.4** | 0.45×0.78 | **0.45×0.4** |
| Number of Cells | 152,335 | 152,012 | 88,838 | 88,447 | 83,931 | 83,923 |
| Wire length(m) | 1.37 | 1.10 | 1.17 | 0.99 | 1.00 | 0.81 |
| Internal Power (mW) | 334 | 310 | 221.2 | 215.8 | 201.2 | 186.2 |
| Switching Power (mW) | 152 | 137 | 118.1 | 107.0 | 101.2 | 86.0 |
| Leakage Power (mW) | 30.0 | 29.1 | 22.8 | 21.0 | 19.1 | 14.4 |
| Total Power (mW) | 516 | **476.1** | 362.1 | **343.8** | 321.5 | **286.6** |
| Memory Access Latency (ps) | 82 | **26** | 77 | **19** | 80 | **58** |
| Memory Access Power (mW) | 4.17 | **1.27** | 4.6 | **1.3** | 4.4 | **0.99** |

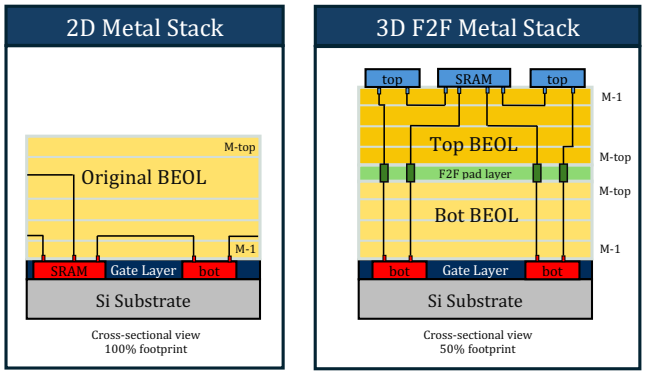


**Figure 4: Cross-section view comparison between 2D and F2F 3D IC metal stack.**

## 4.3 Mapping 2D design to 3D

With specific SRAM configuration to support the spiking generator buffers, we perform logic synthesis to obtain the initial gate-level netlist. The netlist is partitioned and grouped according to the specific hierarchy blocks for logic and memory groups. Following the tier partition, since the memory die contains the standard cells in both dies, we leverage the 3D design flow in [23] to support the logic-on-logic physical implementation. This flow honors the partitioning information with two distinct cells for the top and bottom dies, where the pins are located according to the location of the die in the 3D metal stacks. The physical design (PD) stage is performed iteratively for one die at a time, while cells from another die are fixed. After the final step of the PD stage, we perform a static timing analysis (STA) to estimate the final Power, Performance, and Area (PPA).

## 4.4 Face-to-Face Bonding

Fig. 4 illustrates the difference between 2D and F2F 3D IC metal stack. To support the two-tier F2F 3D IC with [23] design flow, we model the 3D interconnect and parasitics by combining two original 2D interconnect and connect them with vias to represent F2F bonding. We specify the via spacing to ensure that it meets the F2F pitch requirement. The F2F pitch is selected according to the grid size in the bonding layer.

In the case of the standard cells, we created two set of cells from original 2D cells for top and bottom die where pin layer are mapped into the 3D stacking. For memory macros, we changes their type to cover cells to allow standard cell placement of another die in the same region.

**Table 2: The overall performance comparisons between 2D and 3D design of spiking attention accelerator across different array size**

| Array size H×W | 16 × 16 | | 16 × 8 | |
|---|---|---|---|---|
| | 2D | 3D | 2D | 3D |
| Effective Frequency ($GHz$) | 1.58 | **1.68** | 1.68 | **1.93** |
| Area Footprint ($mm^2$) | 0.45×0.9 | **0.45×0.45** | 0.45×0.8 | **0.45×0.4** |
| Number of Cells | 59,299 | 58,271 | 31,257 | 30,671 |
| Wire length(m) | 0.79 | 0.60 | 0.61 | 0.39 |
| Internal Power (mW) | 146 | 146 | 97.8 | 95.8 |
| Switching Power (mW) | 52 | 51 | 30.6 | 26.3 |
| Leakage Power (mW) | 4.0 | 3.0 | 2.6 | 1.9 |
| Total Power (mW) | 203 | **200** | 130.9 | **124.0** |
| Memory Access Latency (ps) | 388 | **100** | 388 | **72** |
| Memory Access Power (mW) | 3.22 | **1.63** | 3.86 | **1.36** |

# 5 EVALUATIONS

## 5.1 Experiment Settings

**Models, Datasets and Training Settings** We evaluated the spiking transformer models trained on two widely adopted neuromorphic datasets: CIFAR10-DVS[17] and DVS-Gesture[2]. We adapted the same model setting using 4-bit quantized and 8-bit quantized results, respectively. Given that image sizes are uniform within a specific vision dataset, the token length of different samples remain consistent as in [7]. DVS-Gesture contains 11 hand gesture categories from 29 individuals under 3 illumination conditions; CIFAR10-DVS is a neuromorphic dataset containing dynamic spike

streams captured by a dynamic vision sensor camera viewing moving images from the CIFAR10 datasets. In Tab. 3, we demonstrate the superiority of the quantized spiking transformers over other SNNs, that can be efficiently executed on our proposed tiny 3D accelerators. The bitwidth in Tab. 3 indicates the bitwidth of spiking activation, synaptic weight and synaptic integration, respectively.

| Model | CIFAR10-DVS | | DVS-Gesture | |
|---|---|---|---|---|
| | Bitwidth | Acc. | Bitwidth | Acc. |
| Spiking VGG[10] | 1/32/32b | 74.8% | 1/32/32b | 97.6% |
| Spiking ResNet[32] | 1/32/32b | 67.8% | 1/32/32b | 96.9% |
| Spiking Transformer | 1/8/16b | 81.2% | 1/8/16b | 98.26% |
| | 1/4/12b | 80.5% | 1/4/12b | 97.92% |

**Table 3: Comparison of the spiking transformer with other existing SNNs on CIFAR10-DVS and DVS-Gesture.**

**Hardware Platform Setup** In this work, we use the commercial 28nm PDK to implement both 2D and 3D F2F designs. The 2D design consists of 6 metal layers, while the 3D design has double metal stack of 2D design with the F2F bond pitch varies from 0.5um to 1um. We use the Synopsys Design Compiler to synthesize the RTL to gate-level netlist and Cadence Innovus to perform physical synthesis.

For the memory, we utilize SRAM modules generated by a commercial memory compiler for various global buffers and storage functions within our system architecture. Specifically, 3072×128b SRAM units are employed for the Activation Global Buffer (Act GLB), Weight Global Buffer (W GLB), and Synaptic Integration Global Buffer (X GLB), all placed on the top tier of our design. Additionally, smaller 96×128b SRAM macros are allocated for the Query (Q) buffer, Key/Value (K/V) buffer, and Spiking (S) buffer on the bottom tier. Two 96×256b SRAM macros are configured to serve as extended X buffers.

## 5.2 Overall Performance Comparision between 2D and 3D

*5.2.1 Layout Comparision between 2D and 3D.* In Fig. 5 and In Fig. 5 and Fig. 6, the layout is presented to show the difference between 2D and 3D design of spiking MLP accelerators and spiking self-attention accelerators.

In Fig. 5(a), the 2D design occupies $700um \times 450um$ while the stacked 3D spiking design occupies a $396um \times 446um$. On the top tier, the W GLB and Act GLB are placed on the edge in Fig. 5(c), and the spiking generator array are occupied in the middle; on the bottom tier in Fig. 5(d), the W and S buffers are placed on the edge, and the spiking spatiotemporal array is placed below the spiking generators. Fig. 5(b), the F2F map indicates the interconnection between the top tier and the bottom tier.

In Fig. 6(a), the 2D design of spiking attention accelerator occupies $900um \times 450um$ while the stacked 3D spiking design occupies an area of $445um \times 446um$. On the top tier, the synaptic integration X GLB and Act GLB are placed on the edge in Fig. 6(c), and the spiking generator array is occupied in the middle; on the bottom tier in Fig. 6(d), the Q, K/V and X buffers are placed on the edge, and the spiking spatiotemporal array is placed between the buffers at the bottom tier. Fig. 6(b), the F2F map indicates the interconnection between the top tier and the bottom tier.

*5.2.2 PPA Comparision between 2D and 3D.* Tab. 1 and Tab. 2 demonstrate that 3D designs significantly enhance power, performance, and area efficiency compared to 2D designs. Using two dies in 3D designs allows the same cells to be accommodated within almost 50% of the original area. This reduced physical distance between the spiking generator and accelerator array leads to shorter total wire lengths, reducing net delay and improving performance. The experiments showed an average 8% increase in effective frequency. Furthermore, fewer buffers were used, decreasing the cell count and leading to a 6% reduction in power consumption.

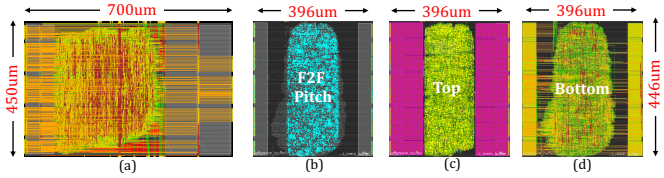


**Figure 5: The layout comparison between 2D and 3D spiking MLP accelerators.(a)2D design. (b)(c)(d) 3D design.**

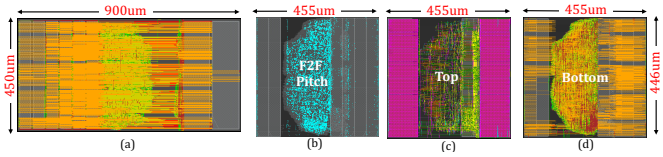


**Figure 6: The layout comparison between 2D and 3D spiking self-attention accelerators.(a) 2D design. (b)(c)(d) 3D design.**

*5.2.3 3D improvement under different Array size.* Both the Spiking MLP accelerator and the Spiking Attention accelerator exhibit a decrease in effective frequency as the array size increases due to the increased number of cells, leading to higher routing congestion. To connect all these cells, ample routing resources are required, or the distance between cells must be minimized. In 2D designs, the spacing between cells is wider, and only a single layer of BEOL is utilized as a routing resource, necessitating longer wires. Conversely, 3D designs, which maintain narrower cell spacing and possess multiple metal layers as routing resources, demonstrate superior routing quality. As seen in Tab. 5, the wire length per net is significantly shorter in 3D designs. Longer wire lengths within a net can induce critical paths; thus, minimizing the distance between connected cells or increasing routing resources can enhance performance. Despite the decrease in effective frequency with increasing array size, as shown in Tab. 1 and Tab. 2, 3D designs consistently exhibit higher effective frequencies than 2D designs.

*5.2.4 Analysis of Memory Access.* We make a hierachical memory access analysis of different memory blocks. In Tab. 4, under different design points with different array size and precision, the memory access latency and energy consumption within 3D design is significantly less than 2D.

**Table 4: The hierachical memory access overhead comparisons between 2D and 3D design of spiking MLP accelerator across different array size and bitwidth**

| Array size H×W, weight/synaptic integration bitwidth | 16 × 128, 8b/16b | | 64 × 16, 8b/16b | | 64 × 16, 4b/12b | |
|---|---|---|---|---|---|---|
| | 2D | 3D | 2D | 3D | 2D | 3D |
| Activation GLB Latency ($ps$) | 24 | 16 | 68 | 19 | 53 | 58 |
| Activation GLB Power ($mW$) | 1.13 | 0.76 | 1.1 | 0.77 | 1.26 | 0.62 |
| Weight GLB Latency ($ps$) | 82 | 26 | 77 | 18 | 80 | 42 |
| Weight GLB Power ($mW$) | 0.46 | 0.1 | 0.47 | 0.09 | 0.45 | 0.11 |
| Activation Buffer Latency ($ps$) | 40 | 16 | 40 | 19 | 47 | 58 |
| Activation Buffer Power ($mW$) | 1.92 | 0.52 | 1.66 | 0.27 | 1.64 | 0.24 |
| Weight Buffer Latency ($ps$) | 28 | 26 | 77 | 18 | 80 | 42 |
| Weight Buffer Power ($mW$) | 1.01 | 0.17 | 1.50 | 0.39 | 1.14 | 0.29 |

**Table 5: 2D and 3D Average Wire Length.**

| | Bitwidth | Array Size $H \times W$ | Aver. Wire Length($\mu$m) | |
|---|---|---|---|---|
| | | | 2D | 3D |
| MLP | 8b/16b | 16 × 128 | 12.7 | 10.8 |
| | 4b/12b | 16 × 64 | 11.8 | 9.59 |
| Attention | 16b | 16 × 16 | 18.6 | 12.3 |
| | 16b | 16 × 8 | 12.9 | 11.5 |

*5.2.5 Wire Length Distribution.* Fig. 7 illustrates the variance in net wirelength distribution between 2D and 3D IC architectures. In 2D design, the frequency of nets exceeding 50um is consistently higher than in 3D design. This disparity is attributed to the vertical stacking employed in 3D designs, notably between the PE array and spiking generator, which facilitates connections through significantly shorter interconnects. Minimizing wire length is paramount as longer interconnects within a signal path lead to critical paths, adversely affecting the chip's operational speed and overall efficiency.

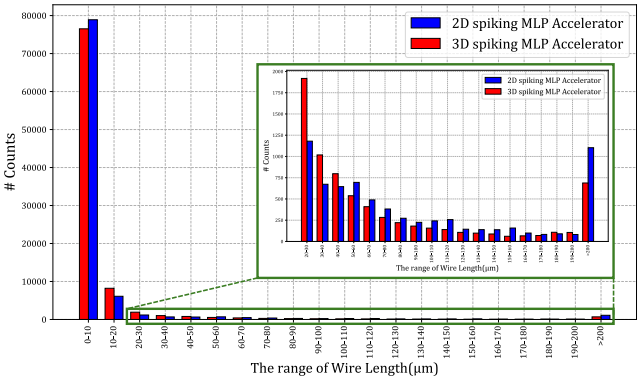


**Figure 7: The distribution of Wire Length in the proposed 3D spiking MLP accelerator,**

## 6 CONCLUSION

In this paper, we introduce the first dedicated 3D accelerator specifically designed for emerging spiking transformers. We identify the spatial and temporal data reuse opportunity on 3D dataflow optimization, and fully exploit this on dedicated 3D accelerators. A tile strategy coupled with kernel fusion is proposed to enable the efficient execution of workloads in spiking transformers. Additionally, our 3D accelerator employs a memory-on-logic and logic-on-logic interconnection scheme via face-to-face (F2F) bonded 3D integration, optimized to minimize energy consumption and latency. Compared to 2D CMOS integration, the 3D accelerator offers substantial improvements. For the spiking MLP workload, it provides a 7.0% higher effective frequency with 7.8% less power reduction and 50% area reduction. The memory access latency and memory access power is reduced by 68.3% and 69.5%, respectively. For the spiking self-attention workload, the 3D accelerator is executed at a 6.3% higher effective frequency, with 50% area reduction and 1.5% less power consumption. The memory access latency and memory access power are reduced by 74.2% and 49.3%.

## 7 ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grants No. 1948201 and No. 2310170 and work supported by the Ministry of Trade, Industry & Energy of South Korea (1415187652, RS-2023-00234159) and the National Science Foundation under CNS-2235398.

## REFERENCES

[1] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. 2015. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems* 34, 10 (2015), 1537–1557.

[2] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha. 2017. A Low Power, Fully Event-Based Gesture Recognition System. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7388–7397.

[3] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. 2023. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. *arXiv preprint arXiv:2303.04347* (2023).

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* 35 (2022), 16344–16359.

[5] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.

[6] Michael V DeBole, Brian Taba, Arnon Amir, Filipp Akopyan, Alexander Andreopoulos, William P Risk, Jeff Kusnitz, Carlos Ortega Otero, Tapan K Nayak, Rathinakumar Appuswamy, et al. 2019. TrueNorth: Accelerating from zero to 64 million neurons in 10 years. *Computer* 52, 5 (2019), 20–29.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*. https://openreview.net/forum?id=YicbFdNTTy

[9] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 21056–21069.

[10] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. 2021. Incorporating Learnable Membrane Time Constant To Enhance Learning of Spiking Neural Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2661–2671.

[11] Wulfram Gerstner and Werner M Kistler. 2002. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.

[12] Bon Woong Ku, Yu Liu, Yingyezhe Jin, Peng Li, and Sung Kyu Lim. 2018. Area-efficient and Low-power Face-to-Face-bonded 3D Liquid State Machine Design. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–6.

[13] Bon Woong Ku, Yu Liu, Yingyezhe Jin, Sandeep Samal, Peng Li, and Sung Kyu Lim. 2018. Design and architectural co-optimization of monolithic 3D liquid state machine-based neuromorphic processor. In *Proceedings of the 55th Annual Design Automation Conference* (San Francisco, California) *(DAC '18)*. Association for Computing Machinery, New York, NY, USA, Article 165, 6 pages. https://doi.org/10.1145/3195970.3196024

[14] Dayeol Lee, Gwangmu Lee, Dongup Kwon, Sunghwa Lee, Youngsok Kim, and Jangwoo Kim. 2018. Flexon: a flexible digital neuron for efficient spiking neural network simulations. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 275–288.

[15] Jeong-Jun Lee and Peng Li. 2020. Reconfigurable Dataflow Optimization for Spatiotemporal Spiking Neural Computation on Systolic Array Accelerators. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*. 57–64. https://doi.org/10.1109/ICCD50377.2020.00027

[16] Jeong-Jun Lee, Wenrui Zhang, and Peng Li. 2022. Parallel Time Batching: Systolic-Array Acceleration of Sparse Spiking Neural Computation. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 317–330. https://doi.org/10.1109/HPCA53966.2022.00031

[17] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. 2017. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience* 11 (2017), 309. https://doi.org/10.3389/fnins.2017.00309

[18] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. 2021. A Free Lunch From ANN: Towards Efficient, Accurate Spiking Neural Networks Calibration. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6316–6325. https://proceedings.mlr.press/v139/li21d.html

[19] Yuhang Li, Shikuang Deng, Xin Dong, and Shi Gu. 2022. Converting artificial neural networks to spiking neural networks via parameter calibration. *arXiv preprint arXiv:2205.10121* (2022).

[20] Ling Liang, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yujie Wu, Lei Deng, Guoqi Li, Peng Li, and Yuan Xie. 2021. H2learn: High-efficiency learning accelerator for high-accuracy spiking neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 11 (2021), 4782–4796.

[21] Ying Liu, Yufei Ma, Ninghui Shang, Tianhao Zhao, Peiyu Chen, Meng Wu, Jiayoon Ru, Tianyu Jia, Le Ye, Zhixuan Wang, et al. 2024. 30.2 A 22nm 0.26 nW/Synapse Spike-Driven Spiking Neural Network Processing Unit Using Time-Step-First Dataflow and Sparsity-Adaptive In-Memory Computing. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. IEEE, 484–486.

[22] Surya Narayanan, Karl Taht, Rajeev Balasubramonian, Edouard Giacomin, and Pierre-Emmanuel Gaillardon. 2020. SpinalFlow: An Architecture and Dataflow Tailored for Spiking Neural Networks. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture* (Virtual Event) *(ISCA '20)*. IEEE Press, 349–362. https://doi.org/10.1109/ISCA45697.2020.00038

[23] Sai Surya Kiran Pentapati, Kyungwook Chang, Vassilios Gerousis, Rwik Sengupta, and Sung Kyu Lim. 2020. Pin-3D: a physical synthesis and post-layout optimization flow for heterogeneous monolithic 3D ICs. In *Proceedings of the 39th International Conference on Computer-Aided Design* (Virtual Event, USA) *(ICCAD '20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3400302.3415720

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8821–8831. https://proceedings.mlr.press/v139/ramesh21a.html

[25] Sonali Singh, Anup Sarma, Nicholas Jao, Ashutosh Pattnaik, Sen Lu, Kezhou Yang, Abhronil Sengupta, Vijaykrishnan Narayanan, and Chita R Das. 2020. NEBULA: A neuromorphic spin-based ultra-low power architecture for SNNs and ANNs. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 363–376.

[26] Sonali Singh, Anup Sarma, Sen Lu, Abhronil Sengupta, Mahmut T Kandemir, Emre Neftci, Vijaykrishnan Narayanan, and Chita R Das. 2022. Skipper: Enabling efficient SNN training through activation-checkpointing and time-skipping. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 565–581.

[27] Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Zhang, Zhongrui Wang, and Renjing Xu. 2023. Masked Spiking Transformer. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 1761–1771. https://doi.org/10.1109/ICCV51070.2023.00169

[28] Man Yao, Jiakui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. 2023. Spike-driven transformer. *arXiv preprint arXiv:2307.01694* (2023).

[29] Ruokai Yin, Abhishek Moitra, Abhiroop Bhattacharjee, Youngeun Kim, and Priyadarshini Panda. 2023. SATA: Sparsity-Aware Training Accelerator for Spiking Neural Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42, 6 (2023), 1926–1938.

[30] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. 2022. Spiking Transformers for Event-based Single Object Tracking. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8791–8800. https://doi.org/10.1109/CVPR52688.2022.00860

[31] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11062–11070.

[32] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11062–11070.

[33] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. 2023. Spikformer: When Spiking Neural Network Meets Transformer. In *The Eleventh International Conference on Learning Representations*.

[34] Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. 2023. SpikeGPT: Generative Pre-trained Language Model with Spiking Neural Networks. arXiv:2302.13939 [cs.CL]