Scalable Pitch-Constrained Neural Processing Unit for 3D Integration with Event-Based Imagers

Maxence Bouvier, Alexandre Valentian, Gilles Sicard

Scalable Pitch-Constrained Neural Processing Unit for 3D Integration with Event-Based Imagers

Maxence BOUVIER
Univ. Grenoble Alpes
CEA, List
F-38000 Grenoble, France
maxence.bouvier@cea.fr

Alexandre VALENTIAN
Univ. Grenoble Alpes
CEA, List
F-38000 Grenoble, France
alexandre.valentian@cea.fr

Gilles SICARD
Univ. Grenoble Alpes
CEA, Leti
F-38000 Grenoble, France
gilles.sicard@cea.fr

***Abstract*—Event-based imagers are bio-inspired sensors presenting intrinsic High Dynamic Range and High Acquisition Speed properties. However, noisy pixels and asynchronous readout result in poor energy-efficiency and excessively large output data rates.**

In this work, we use Convolutional Spiking Neural Network filters to compensate these drawbacks and reduce output bandwidth by 10x.

We designed a neuromorphic core as a distributable block that benefits from 3D integration technology with direct and parallel access to 32x32 pixels, enabling reduced frequency operation. Post-layout simulations depict a peak energy efficiency with 2.83pJ per Synaptic Operation (equivalent to 0.093fJ/event/pix) at the nominal literature input event rate.

***Index Terms*—Spiking Neural Networks, Neuromorphic, 3D IC Technology, Near-Sensor Computing, Event-Based Vision**

I. Introduction

We observe an increasing number of applications where being able to reliably distinguish movement or objects is necessary, e.g. autonomous machines like drones or taxis, or mixed reality. Under certain conditions, to acquire images whose quality is high enough to guarantee object detection or movement evaluation may require the use of High Dynamic Range (HDR) and High Acquisition Speed (HAS) cameras. Moreover, most of these applications concern embedded devices or machines, which generally require low power components and reduced form factor. Traditional CMOS imagers deliver exhaustive and spatio-temporally redundant data, and usually do not combine HDR and HAS properties without post-treatment.

Thus, we focus on Event-Based (EB) imagers [1], [2], which have successfully been used for movement evaluation and autonomous navigation [3]. Because of the bio-inspired behavior of EB image sensors, they naturally combine HDR, HAS, and event-based data acquisition. The latter means that the sensor does not deliver any data (noise excepted) when nothing occurs - i.e. when the sensor and the scene are motionless. No redundant data are generated nor treated, thus minimizing power consumption. However, EB imagers come with several drawbacks. They are noisy and hardly scalable at HD resolutions because of the required output bandwidth.

As a solution, this work presents a neural processing unit evaluating a Convolutional Spiking Neural Network (CSNN) conceived to be easily distributed behind an EB pixel grid by employing 3D IC technologies [4], [5]. Analogously to biological neurons of the striate cortex [6], the CSNN evaluated by the neuromorphic core detects oriented edges in the raw input EB data stream. It reduces data throughput by up to an order of magnitude and filters out noise; thus minimizing several drawbacks of raw EB data streams, while conserving spatial and temporal information. Such construct would be a first step in the realization of a complete bio-inspired vision system, and the initial motivation for this work was to verify that CSNN algorithm could be implemented near-sensor.

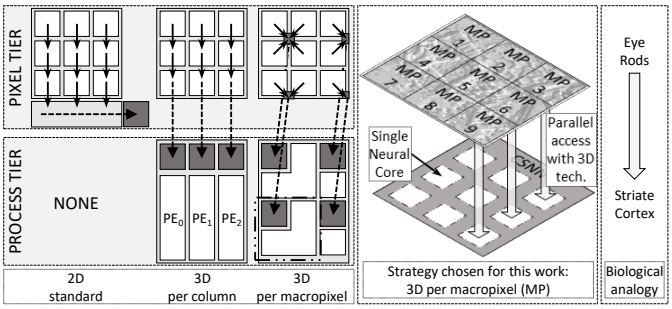


Fig. 1: Left: illustration of different 3D readout possibilities. Right: chosen strategy and analogous biological system.

Our neural core evaluates 256 convolutional spiking neurons and consumes only 2.86pJ per Synaptic Operation (SOP). It takes advantage of the envisaged 3D parallel interconnection with a block of 32x32 pixels by integrating the pixel readout module which encodes the spikes' address in a custom format that allows for optimized synaptic mapping information - i.e. weights and associated neuron addresses - storage requiring only 300 bits. The timing constraints on the arbiter is also drastically reduced. Moreover, the neural core can be tiled to manage high resolution sensors without inducing overhead and the readout. The allowed surface is constrained by the pixels above the core, whose pitch is set at 5μm targeting the state of the art 720p EB imager [7] characteristics, which results in a core surface of only 0.026mm^2. To the best of our knowledge, it is the first SNN hardware accelerator realized targeting a near-sensor implementation.

Related works are introduced section II, and algorithm to hardware optimization efforts are analyzed section III. Section IV depicts the circuit architecture, and post-layout results are presented and compared with other neuromorphic cores section V. We finally conclude in section VI.

II. Related Works

A. Event-Based Imagers

Event-based imagers are bio-inspired sensors based on asynchronous readout of individual pixels. Each pixel emits an event - a spike - when a change in the illumination measured overcomes a threshold [1], [2] as would the rods in human eyes [8]. Whereas simultaneously providing HDR, HAS, and EB data acquisition; they are hardly scalable with standard 2D integration because the complexity of the readout system - called arbiter - increases linearly with the number of cells to read [9]. The output bandwidth tends to reach important values, of the order of tens of Gb/s, not suitable for low power systems. Moreover, the pixels can be very noisy.

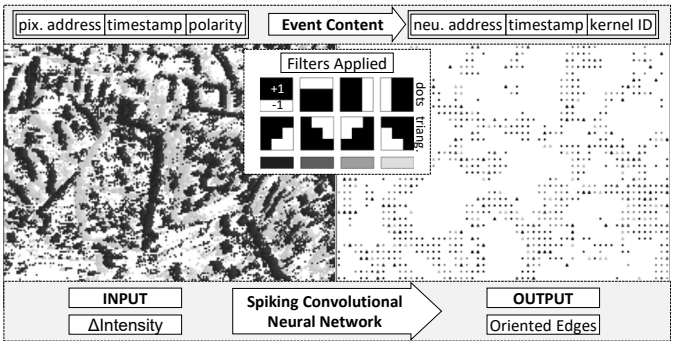


Fig. 2: CSNN results. Left: input data, B&W dots correspond to -1 & +1 event polarity respectively. Right: corresponding output data, round dots correspond to vertical and horizontal filters, triangles dots correspond to diagonal ones. Input data from [22].

Recent examples of EB sensors trying to solve these problems are [7], [10], [11].

Reference [11] accesses eight pixels simultaneously in order to reduce the timing constraints on the arbiter. Reference [10] filters out faulty pixels and noise by counting and thresholding spikes emitted by group of 2x2 pixels. Reference [7] exploits 3D IC technology to readout the pixel matrix by row, reducing the arbiter complexity by by 1280. Moreover, more space is available on the bottom tier thanks to 3D stacking, they thus implemented a filter by Region of Interest to reduce the output bandwidth of the sensor.

B. 3D Integrated Circuit Technology

Image sensors largely benefit from 3D IC technologies, with for example Cu-Cu bonding, for numerous reasons [4], [5]. It provides more space for maximizing the pixel fill factor and allows heterogeneous integration, i.e. conjoint integration of high quality light sensing devices on the pixel tier with efficient readout and complex compute modules on the bottom tier - also called process tier. Several actors have already shown achievements basic 2D imaging technologies alone would not have permitted, for smart or autonomous [12]–[14] and EB [7] imagers. Fig.1 illustrates the 3D readout mechanisms used in these works. The 3D per macropixel (MP) strategy, as employed in [14], enables to homogeneously tile processing cores that have direct parallel access to pixels, straightforwardly permitting distributed processing architectures design on the bottom tier.

C. Spiking Neural Networks Hardware Implementation

Hubel and Wiesel have shown that one of the first steps related to visual perception in mammals' brain is edge orientation detection [6]. Similarly, edge orientation discrimination is usually realized by the first layers of CSNNs when trained with bio-inspired methods on EB data [15], [16]. However, simulating SNNs on classical computers is time and power consuming because of the sparse and event-based nature of data and computation flow [17].

Hence, many academic [18]–[20] and industrial [21] actors are working on dedicated hardware for evaluating SNN algorithms. The neural cores presented are usually designed with more [18], [21] or less [19], [20] programmability regarding the final algorithm, with more [18], [21] or less [19], [20] bio-plausibility regarding the neuron mechanisms, and target either shallow [18]–[20] or deep [21] SNN topologies. As a general rule: more programmability or complexity is usually obtained at the expense of energy efficiency, operation speed, or area [17].

TABLE I: CSNN Algorithmic Parameters and Values

Parameter name	Symbol	Value
Number of Kernels	N_k	8
RF Width	W_{RF}	5pix
Threshold Voltage	V_{th}	8
Stride	d_{pix}	2
Refractory Period	T_{refrac}	5ms
Leakage Type	f_{leak}	exponential
Leakage Time Constant	τ	$1/3^{rd}$ of 20ms

Our design mixes several ideas presented by related works. It exploits the benefits that would be brought by combining EB imaging with 3D stacked macropixel readout for implementing a fully hardwired shallow CSNN evaluator. We minimize the flexibility and size of the CSNN implemented for performance matters aiming for embedded operation. The long-term idea of our work is that realizing spatio-temporal edge orientation filtering near-sensor would permit to reduce drawbacks related to EB sensors. We choose to implement the neural core in a data-stream processing perspective to take advantage of the EB nature of data: no computation or data movement is uselessly realized when no input data is available.

III. Algorithm Hardware Co-Optimization

A. Convolutional Spiking Neural Network

A convolutional layer of a CSNN is a mesh of local filters - called neurons - distributed throughout the input frame. Each neuron takes in input a small window of the whole input frame - called Receptive Field (RF). Each spike generated in this window adds a value - called synaptic weight - to the neuron's membranes - called kernel potentials V_{k_i}. When a kernel potential overcomes the neuron's threshold V_{th}, the neuron emits a spike - it fires. Neurons operating as described are called Integrate and Fire (IF).

CSNNs conserve and exploit the spatio-temporal information contained in the input data. As neurons are regularly distributed throughout the input space, and observe only a small window, spatial information is conserved by labelling output events with the address of the emitting neuron. Temporal information is exploited with two mechanisms. First the leakage, which consists in progressively decreasing the neurons' kernel potentials through time. Contributions of low frequency noisy spikes are thus deleted with time. On the other hand, spikes arriving in quick succession are more likely to make the neuron fire. This denotes that leaky neurons favor temporally correlated input spikes, and consequently limit the impact of noise-generated spikes. Leakage is the direct equivalent of capacitors discharge in analog electronics. Secondly the refractory period, which limits the maximum output frequency of a neuron by forbidding firing during an amount of time T_{refrac} after each spike emission. The impact of faulty always-on pixels is thus diminished.

Hence, applying a mono-layer CSNN with Leaky Integrate and Fire (LIF) neurons on a raw stream of events acquired by an EB camera enables to conserve spatial information while reducing noise and event-rates. The output is a new stream of events, each corresponding to a spatio-temporal pattern - called feature or kernel - that fired, as illustrated Fig. 2.

B. From Algorithm to Hardware

SNN acceleration near-sensor has been discussed but not yet demonstrated [10]. 3D integration facilitates the realization, even more with Convolutional NNs whose locality property is perfectly suited for macropixel arrangement [7], [14] (Fig. 1). However, memory requirements of Spiking NNs, with every neuron states continuously stored, make them area-hungry [18]–[20]. Our objective is thus to permit CSNN evaluation in an area constrained to only the

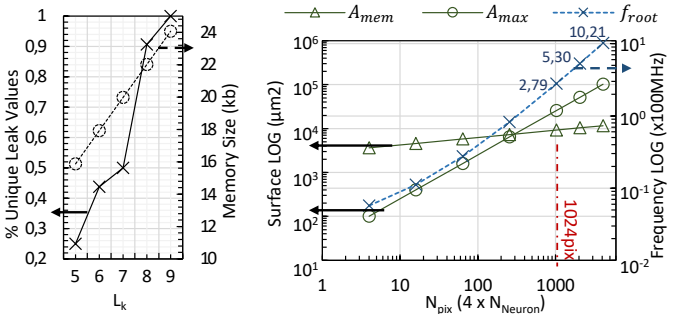


Fig. 3: Design Space Exploration. Left: impact of L_k on the LUT precision and M. Right: $N_{pix]}$ trade off between f_{root} and A_{mem} requirements. A_{mem} values are obtained with SRAM cut generation exploration.

area of the pixels above the core in the macropixel construct. For that, we minimize the chip programmability and we apply several standard NNs related hardware optimizations. We also propose a custom method for mapping the pixel grid to the neurons taking advantage of the local 3D interconnections to pixels.

1) Algorithmic Parameters: Apart from the kernel patterns, the neuron threshold value V_{th}, and the refractory period duration T_{refrac}, every algorithmic parameters is fixed and hardwired in the design. We arbitrarily set the distance in pixel between two RF centers - called the stride d_{pix} -, the number of kernels N_k per neuron, the RF width W_{RF}, and the weight precision. The rest has been set after an exploration that aimed at obtaining a compression ratio $CR = n_{ev_{in}} \div n_{ev_{out}}$ of approximately 10. The resulting parameter values are listed table I.

With the stride set at two, there is only one neuron to evaluate per group of four pixels. W_{RF} is chosen so that neighboring neurons' RFs overlap. The kernel patterns, visible in the inlet Fig. 2, are inspired from oriented edges obtained with Spike Timing Dependent Plasticity (STDP) training, which analogously represent actual neurons of the striate cortex [6], [15]. Finally, because near-binary weight distribution is sometimes spontaneously obtained by training [16], possible weight values are reduced to -1 or +1.

2) Standard Optimizations: We deploy a few standard hardware optimization technics for improving the energy and area efficiency of NN accelerators, namely clock-gating, fixed frequency scaling, and approximate computing. Our core behaves as a data-stream accelerator where each module works sequentially on a data, without storing intermediate results (apart from neuron states). Data transits one-way, from input to output, and no time nor energy is lost in a ping-pong between a central processing unit and an external memory. The frequency of each module is adapted to its local data rate; and if a module has no valid data in input, most of its components are clock gated. Approximate computing is realized with reduced bitlengths L_X of stored timestamps (L_{TS}) and kernel potentials (L_k). L_{TS} is set to represent the full 20ms of leak range. Hence, with a LSB corresponding to 25μs, 10 bits are sufficient. An additional bit is used as a flag indicating overflow, resulting in L_{TS} = 11. The bitlength of the kernel potentials L_k is set to guarantee high precision exponential leakage. Each time a neuron state is loaded, leak is applied by multiplying every kernel potential with the decrement factor $leak_{value} = exp(-(t_{curr.} - t_{in})/\tau)$, where t_{in} is the timestamp of the last input spike. Leak values are stored in a 64-input Look Up Table (LUT) and the left graph of Fig. 3 depicts the variation of the precision - actually of the number of identical decrement factors stored - of the LUT with L_k (left axis).

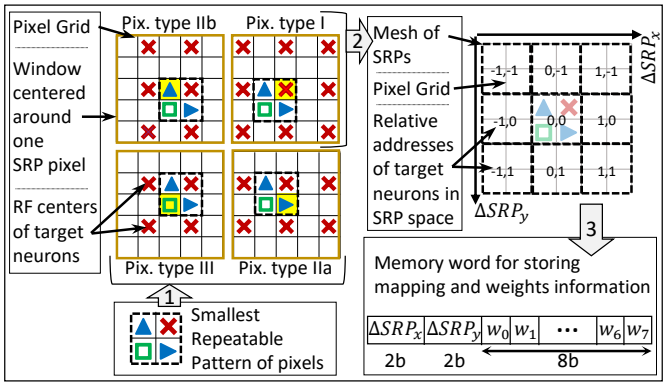


Fig. 4: Smallest Repeatable Pattern definition and associated network mapping memory storage strategy.

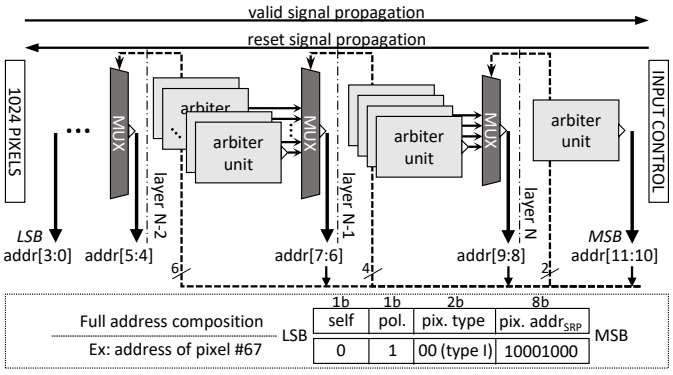


Fig. 5: Schematic of the arbiter working principle.

The 50% drop in precision when L_k decreases from 8b to 7b led us to set it at 8b.

3) 3D-Enabled Optimization: To take advantage of the distributed parallel interconnect of the macropixel stack, each core embeds its own readout module to directly access its pixels. We customize the spike addressing which enables us to design a specific network mapping strategy we call mapping by Smallest Repeatable Pattern (SRP). Mapping consists in listing the target neurons and associated weights from the address of an input spike.

A SRP is the smallest block of pixels and RF centers that permits to fully represent the CSNN when meshed uniformly. With $d_{pix} = 2$, the SRP is a group of 2x2 pixels, as depicted Fig. 4. In step 1, the connections between every pixel of a SRP and neighboring neurons are found by taking a window of width W_{RF} around this pixel and looking at the RF centers inside this window. Then in step 2, for each pixel of a SRP, the associated target neurons are mapped by their relative addresses in ΔSRP. Finally in step 3, for each target neuron of each pixel composing a SRP a 12b word is stored in the mapping memory, composed of the $\Delta SRPs$ coordinates (both stored on 2 bits) and the eight 1b weights w_i associated to each kernel potential. Pixels type I, II (a & b), and III respectively have 9, 6, and 4 target neurons. The full mapping memory thus requires only (9 + 6 + 6 + 4)x12 = 300b.

Using SRP to describes pixels to RF centers connections mapping information is only dictated by the stride d_{pix}. It is independent of the total number of neurons evaluated or of the core position with respect to the matrix of pixels. Hence, no overhead would be induced by tiling cores as illustrated Fig. 1.

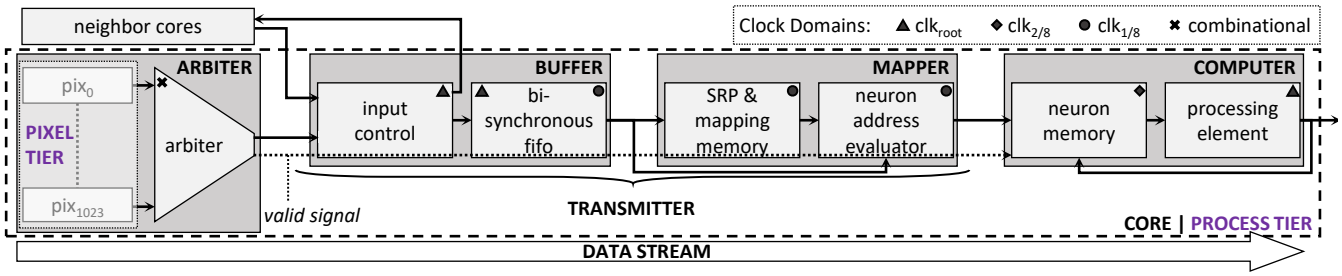


Fig. 6: Overview of the data-stream architecture of the neural processing unit.

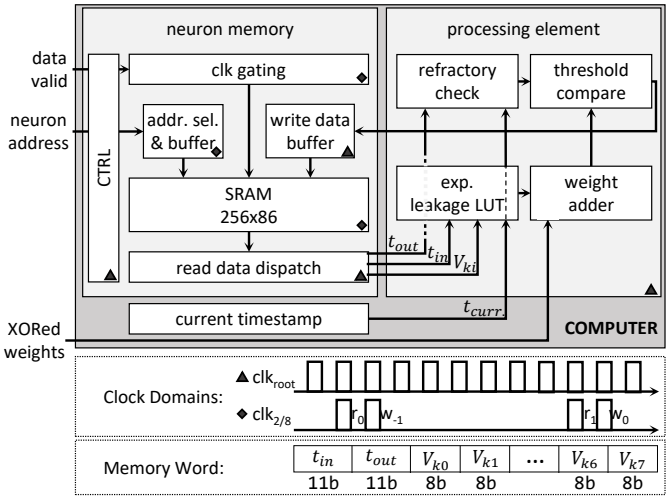


Fig. 7: Schematic of the computer behavior.

C. Number of Pixels and Target Performances

For deciding the number of neurons N_{neuron} a core evaluates, we look at the impact on minimal frequency f_{root} and core area A_{MP} requirements. f_{root} is directly proportional to the average pixel input event rate f_{pix}, the number of target neurons per input spike $N_{RF_{max}}$ - nine for pixel type I -, the number of kernels evaluated per neuron N_k, and N_{neuron}. The maximal core area A_{max} is constrained by the number of pixels N_{pix} of the macropixel stack and their pitch p_{pix}. Also, as a core must instantiate its own SRAM to store neuron states (eight 8b kernel potentials and two 11b timestamps accounting for 86b memory words), the minimal core area is bounded from bellow by the memory cut area A_{mem}.

The core is designed with a single PE, and f_{pix} and p_{pix} are taken to compete with the state of the art event-based imager, namely 3.16kev/s/pix (maximum *internal* pixel event rate) and 5μm respectively [7]. With that, the right graph of Fig. 3 depicts the evolution of the area boundaries (green) and f_{root} (blue) as a function of N_{pix}. When $N_{pix} < 1024$, $A_{mem} > A_{max}$ which is not possible, so N_{pix} should at least be 1024. However, with $N_{pix} \geq 2048$, f_{root} is at least 530MHz, which could lead to important power consumption. So, N_{pix} is set at 1024, resulting in a macropixel composed of 32x32 pixels on top of a core evaluating 265 spiking convolutional neurons.

IV. Proposed Architecture

The overall architecture of our data-stream neural processing unit is illustrated Fig. 6. The behavior is kept non-synchronous with combinatorial circuitry up to the input control module where the signal is sampled by a metastable tolerant synchronizer. After that it is fully synchronous with several clock domains and clock gating levels to reduce power consumption of each module individually.

A. Arbiter

The spike first passes through the arbiter, where its address is encoded. The arbiter implemented, illustrated Fig. 5 is adapted from [23] with digital electronics and modified address encoding protocol to avoid using tristate buffers, which are replaced by multiplexers.

To emit an event, a pixel sets at 1 its valid signal which then propagates through the arbiter up to the input control module. The input control synchronously samples the valid signal and send a reset pulse. When the rightmost arbiter unit (AU) receives the reset signal, it encodes a 2b-address that corresponds to its input AU with active signal high.The multiplexer of layer N uses this 2b-address for selecting the next correct AU to which propagate the reset signal and from whom to read its encoded 2b-address.This 2b address is concatenated with the previous one to generate a 4b-address that will be used by the multiplexer of layer N-1. This goes on sequentially. The SRP address $addr_{SRP}$ of the pixel is encoded by concatenation of every 2b address. The AU closest to pixels directly encodes the pixel type. It also conserves the bit of polarity $pol.$ encoded by the pixel and set at 0 a bit called $self$ to indicate that the event does not come from a neighbor MP. These four elements compose the full event address.

B. Transmitter

The transmitter is composed of a buffer and the pixel to neuron mapper. The buffer includes the input control which manages spikes from the arbiter and neighboring MPs (distinguished with the $self$ bit) and stores them in a bisynchronous FIFO [24]. When an event is available in the FIFO, the mapper fetches it and directly passes the valid signal and the $addr_{SRP}$ downstream. The event pixel type dictates which target neurons mapping information ($\Delta SRPs$ and associated weights) to load. The neuron address evaluator first decomposes the $addr_{SRP}$ into SRP coordinates SRP_x and SRP_y. It then recomposes the target neuron address as $addr_{RF}$ = [SRP_x + ΔSPR_x; SRP_y + ΔSRP_y]. It also XOR the 8 weights with the event polarity and passes the resulting vector of bits to the computer module along with $addr_{RF}$. This operation is repeated at $f_{1/8} = 1/8 \times f_{root}$ for every target neuron associated with the pixel type of the event (e.g. 9 times for pixel type I).

C. Computer

The computer is composed of the neuron state memory that stores the 256 neuron states and a single PE that updates the eight kernel potentials. Fig. 7 depicts the dataflow inside the computing unit of the core.

1) Neuron Memory: $addr_{RF}$ is used to load the neuron's previous state at the read cycle r_0 (see chronograms Fig. 7). A single memory word contains the eight kernel potentials V_{k_i} and the timestamps t_{in} and t_{out}, corresponding to the times of the last input and outputted spikes respectively. The V_{k_i} are sequentially sent to the PE for update, one V_{k_i} by clock cycle at clk_{root}. To guarantee functional read/write

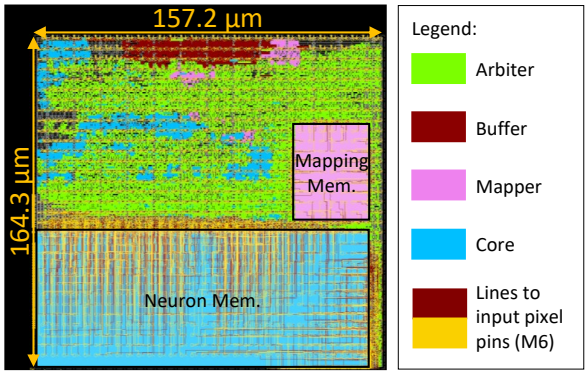


Fig. 8: Layout of a single neural core for 32x32 pixels.

synchronization with a single port SRAM, a write data buffer is placed at the input of the memory data port. It consists in seven registers in parallel, each sequentially storing an updated V_{k_i}. The last updated V_{k_7} is not stored in a register but directly written, at write cycle w_0, along with the seven others V_{k_i} and the timestamps t_{in} and t_{out}, at the same $addr_{RF}$ which has been memorized by the address selector and buffer. The bits corresponding to t_{out} are masked, excepted when the neuron fires, in which case all V_{k_i} are set to zero at write time. The control module manages clock gating and read and write addresses selection throughout each read/write cycle. The SRAM works under the $clk_{2/8}$ clock domain, and is clock gated if the valid signal propagated all the way from the arbiter is low.

2) Processing Element: The PE is fully combinational. It first applies leakage on V_{k_i} - as described sec. III-B2. Then, the XORed weight associated to the ith kernel selects if the leaked V_{k_i} is added or subtracted one. V_{k_i} is now fully updated and compared with V_{th}. If $V_{k_i} > V_{th}$, the neuron spikes. Otherwise the memory simply stores the updated V_{k_i} and t_{in} at $addr_{RF}$ - as explained above. In parallel, the refractory checker verifies $t_{curr.} - t_{out} < T_{refrac.}$. If the condition is true the neuron is in refractory period, in which case the spiking operation is not allowed, even if $V_{ki} > V_{th}$ is true. If an output spike is produced, the PE sends an event word composed of [add_{SRP}, $t_{curr.}$, i] to a virtual output port.

V. Results

A. Methodology

We evaluated the area and power consumption of the circuit based on post-layout simulations using uniform random spiking patterns as input to the neural core. All numbers presented below are associated to a single neural core. The power contributions of pixels and 3D interconnects are not considered in this work. Only the neural processing block is designed, and pixel-related input pins are distributed on top the core at metal layer 6, with pitches of 5μm. Nevertheless, the IP proposed here could be straightforwardly tiled and integrated within a full 3D stacked EB imager conception flow using available 3D process design kits. The snapshot of the layout is shown Fig. 8.

The design was synthesized with Synopsys© Design Compiler using ST 28nm FDSOI technologies standard libraries. Place and route was realized with Cadence© Innovus™. Finally, after signoff timing and power analysis with the Synopsys PrimeTime© suite, the core power evaluation is done through post-layout simulations under Mentor® Questa® Simulator.

B. Input Rate, Chip Frequency and Power Consumption

Two different target f_{root} for synthesis and simulations are tested: 400MHz and 12.5MHz. These frequency values permit to easily represent timestamps with LSB corresponding to 25μs. They are adapted to manage the 720p equivalent input event rates of 3.5Gev/s and 300Mev/s respectively. The former is close to the maximal internal event rate of 2.92Gev/s measured by [7], and the latter is the nominal event rate for comparing EB sensors. As our core manages 900 times less pixels than a full 720p sensor, the input event rates of 3.5Gev/s, 300Mev/s, and 100kev/s are scaled down for simulation at 3.89Mev/s, 333kev/s and 111ev/s respectively.

Fig.9 illustrates the power consumption of a single core obtained from post layout simulations at these different frequencies and event rates. The 400MHz permits to handle a huge number of events, however, a power consumption approaching 1mW is not adapted for an embedded device. Moreover, a CR of 10 still leads to 350Mev/s in output, easily corresponding to a few Gbit/s when encoding spikes individually with a neuron address, a timestamp, and a kernel number. Thus, 12.5MHz is more suited for embedding our core into an actual device, consuming 47.6μW at the nominal input event rate. The SRAMs and most of the registers being clock gated when no data is available, the power consumption drops by 2.5x to 19μW at the minimal input activity.

TABLE II: Comparison with State of the Art SNN Accelerators

Reference	**This Work**		[18]	[19]	[21]	[20]	
Implementation	Digital		Digital	Digital	Digital	Digital	
IC Technology	28nm FDSOI		28nm FDSOI	65nm	14nm FinFET	10nm FinFET	
Data Obtained From	Post-Layout		Chip	Chip	Post-Layout	Chip	
NN Type	C-SNN		FC-SNN	FC-BaNN[c]	Various	Various	
Core Area (mm^2)	**0.026**		0.086	10.08	0.4	1.72	
Neu. Behaviors (nb)	1		20	1	6	1	
Neu. per Core (nb)	256		256	1194	max. 1024	64	
Syn. Weights Storage	1bit SRAM		3+1bit SRAM	SRAM	1 to 9bit SRAM	7bit SRAM	
On-Chip Training	No		Yes	Yes	Yes	Yes	
Synapses per Core (nb)	30.4k		64k	238k	1M to 114k	16k	
Neu. Density (nb/mm^2)	9.8k		3.0k	0.1k	max. 2.6k	2.4k	
Syn. Density (nb/mm^2)	1.17M		741k	23.7k	2.5M to 285k	595k	
Supply Voltage (V)	1.2		0.55-1.0	0.8-1.2	0.5-1.25	0.45-0.9	
Chip Frequency (MHz)	400	12.5	75	20	-	105	506
SOP/s	**194.4M**	16.7M*	37.5M	-	min. 285.7M	81.3M	393.8M
Energy per SOP	4.8	**2.86**	12.7 (0.55V)	-	>23.6 (0.75V)	3.8 (0.525V)	8.3 (0.9V)
Total Core Power (µW)	948.4	**47.6**	476.3[a]	23.6k (0.8V)[b]	6.7k	308.75[a]	3.3k

[a] Includes IO and classical to event-based input converter.
[b] Obtained from SOP/s and energy per second values, normalized by the number of cores.
[c] BaNN stands for Binarized activation Neural Network.

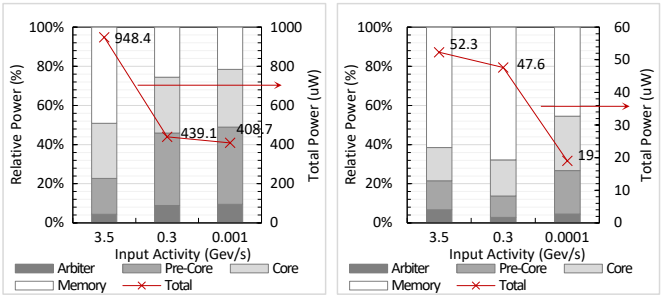
C. Comparison With Other Works

A common metric to evaluate the efficiency of SNN evaluators is the energy per Synaptic Operation (SOP). A SOP is defined as a complete update operation onto one kernel potential of a neuron. For instance, a spike impinging from a pixel type I updates 9 neurons, each characterized by 8 kernels, thus resulting in 9x8 = 72 SOPs. The four different pixel types characterizing a SRP lead to an average number of target neurons per input spike of 25/4 = 6.25, thus the number of SOP per second at an input event rate of 333kev/s is 6.25x8x333k = 16.65M SOP/s. With that in mind results the reader can find a comparison with other SNN accelerators table II.

On top of that, to characterize power consumption of an eventual 3D stacked EB imager, we extract the *dynamic energy per event*, as proposed in [7], [10]. Aiming for a scalable metric independent of the matrix pixel size, we divide this metric by the number of pixels, resulting in an energy per event per pixel. At 333kev/s and 12.5MHz, our core consumes 93.0aJ/ev/pix. The reader can find an exhaustive comparison with EB imagers in table III.

D. Discussion

A strong point in our design resides in having the arbiter local to our core. Arbitrating 1024 pixels with 4-input to 1-output AUs requires only 5 layers. Moreover, with f_{pix} = 3.16kHz the average inter-spike delay for 1024 pixels is 309ns, which corresponds to a minimum sampling frequency of 324kHz. A full 720p sensor would



(a) $f_{root} = 400$MHz (b) $f_{root} = 12.5$MHz

Fig. 9: Post-Layout power distribution for several input event rates. The bars are normalized by the total power (red line).

require 10 arbitration layers and a minimum sampling frequency of 2.92GHz [7]. This is a strong argument in favor of 3D readout per MP, in addition to requiring only 300b of storage for mapping the full network.

We hardwired most of the algorithm parameters, drastically reducing our core flexibility. However, it enables us to make it fit in the 5μm pixel pitch constrained area representing only 0.026mm^2.

Nevertheless, because the floorplan is not 100% dense, several tracks can be followed to improve our design. For example we could add more programmability to our core or implement 4 PEs in parallel instead of a single one which would permit to reduce f_root at 3.125MHz.

VI. Conclusion

We see many applications appearing where HDR and HAS imagers are required. EB sensors present both these properties, but come with several drawbacks. To solve their limitations, we deploy a CSNN to filter noise and reduce the output event rate by 10x while conserving spatial and temporal information. Considering embedded evaluation of the algorithm, we implemented a data stream EB neural core mostly hardwired. By envisaging 3D stacking integration for near-sensor computing directly behind an EB imager, the core area is constrained to 5μm-pitch pixels and fits in only 0.026mm^2. It consumes only 2.86pJ/SOP at a 720p equivalent nominal input event rate of 300Mev/s, corresponding to 93.0aJ/ev/pix.

This performance can be achieved thanks to direct access to pixels with 3D IC technology. The core instantiates a direct distributed 1024-input pixel encoder, which enables to store the full mono-layer CSNN in only 300 bits and to drastically reduce the chip frequency. Moreover, it can be tiled without overhead to manage high resolution EB image sensors. As a next step we will integrate the proposed neural processing unit within a 3D stacked EB imager design for ego-motion evaluation target application.

TABLE III: Comparison with State of the Art EB Imagers

Reference		This Work		[7]	[10]	[11]
IC Technology		3D			2D	
Filter Type		Convolutional Spiking Neurons		Regions of Interest	Event Counting	None
Technological Node		None[a] + 28nm FDSOI		90nm BI CIS + 40nm CMOS	65nm CMOS	90nm CIS BSI
Implementation Stage		Post-Layout		Circuit Chip		
Resolution		N[b] x (32 x 32)		1280 x 720	132 x 104	640 x 480
Pixel Size (µm2)		5,03 x 4,80		4.86 x 4.86	10 x 10	9 x 9
Clk Frequency (MHz)		400	12.5	100	50	50
Input event rate full res.	Low (kev/s)	100	100	100	100	100
	High (Mev/s)	3500	300	300	180	300
Power full res. (mW)	Low IN rate	367.83	17.1[c]	32	0.25	27
	High IN rate	854.01	42.8[c]	84	4.9	50
Power 1024 pix eq.[d] (µW)	Low IN rate	408.7	19	35.6	18.6	90.0
	High IN rate	948.9	**47.6**	93.3	365.5	166.7
Energy/event/pix[e] (aJ/pix)		150.7	**93.0**	188.1	1882.8	249.6
Static Power (nW/pix)		399.1	18.5	34.7	18.0	87.9
Max. IN Event Rate (Mev/s)		**3500**[f]	300	2920[f]	180	300

[a] Pixel tier not implemented.
[b] Results shown for a block of 32x32 pixels, designed to be scalable.
[c] Values for an equivalent resolution of 1280x720 pixels; i.e. $N = 900$.
[d] Values for an equivalent resolution of 32x32 pixels.
[e] Dynamic energy, defined as in [10], divided by the total number of pixels.
[f] Peak *internal* activity.

References

[1] P. Lichtsteiner *et al.*, "A 128 x 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor," *JSSCC*, 2008.
[2] C. Posch *et al.*, "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS," *JSSCC*, 2011.
[3] G. Gallego *et al.*, "Event-based Vision: A Survey," *TPAMI*, 2020.
[4] K. Sakuma, 3D Integration in VLSI Circuits, 1st ed. Boca Raton: CRC Press., 2018.
[5] P. Vivet *et al.*, "Advanced 3D Technologies and Architectures for 3D Smart Image Sensors," in *DATE*, 2019.
[6] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J Physiol*, 1959.
[7] T. Finateu *et al.*, "1280 x 720 Back-Illuminated Stacked Temporal Contrast Event-Based Vision Sensor with 4.86μm Pixels, 1.066GEPS Readout, Programmable Event-Rate Controller and Compressive Data-Formatting Pipeline," in *ISSCC*, 2020.
[8] D. Purves *et al.*, "Vision: the Eye," in Neuroscience. Oxford University Press, 2018.
[9] S.-C. Liu *et al.*, "On-Chip AER Communication Circuits," in Event-Based Neuromorphic Systems, John Wiley & Sons, Ltd, 2014.
[10] L. Chenghan *et al.*, "A 132 by 104 10μm-Pixel 250μW 1kefps Dynamic Vision Sensor with Pixel-Parallel Noise and Spatial Redundancy Suppression," in *Symposium on VLSI Circuits*, 2019.
[11] B. Son *et al.*, "4.1 A 640x480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation," in *ISSCC*, 2017.
[12] A. Nose *et al.*, "Design and Performance of a 1 ms High-Speed Vision Chip with 3D-Stacked 140 GOPS Column-Parallel PEs," *Sensors*, 2018.
[13] https://www.sony-semicon.co.jp/e/technology/imaging-sensing/.
[14] L. Millet *et al.*, "A 5500-frames/s 85-GOPS/W 3-D Stacked BSI Vision Chip Based on Parallel In-Focal-Plane Acquisition and Processing," *JSSCC*, 2019.
[15] S. R. Kheradpisheh *et al.*, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, 2018.
[16] F. Paredes-Vallés *et al.*, "Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
[17] M. Bouvier *et al.*, "Spiking Neural Networks Hardware Implementations and Challenges: A Survey," *ACM JETC*, 2019.
[18] C. Frenkel *et al.*, "A 0.086-mm^2 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28-nm CMOS," *IEEE Trans. Biomed. Circuits and Systems*, 2019.
[19] J. Park *et al.*, "A 65-nm Neuromorphic Image Classification Processor With Energy-Efficient Training Through Direct Spike-Only Feedback," *JSSCC*, 2020.
[20] G. K. Chen *et al.*, "A 4096-Neuron 1M-Synapse 3.8-pJ/SOP Spiking Neural Network With On-Chip STDP Learning and Sparse Weights in 10-nm FinFET CMOS," *JSSCC*, 2019.
[21] M. Davies *et al.*, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, 2018.
[22] E. Mueggler *et al.*, "The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM," 2016.
[23] P. Yang *et al.*, "Low-power priority Address-Encoder and Reset-Decoder data-driven readout for Monolithic Active Pixel Sensors for tracker system," *NIMPR, Section A*, 2015.
[24] I. Miro Panades *et al.*, "Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures," in *NOCS'07*, 2007.