UNIVERSITY OF CALIFORNIA, SAN DIEGO

Large Scale Asynchronous Low-power VLSI Systems for Event-driven Sensory and Neural Processing

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Electronic Circuits and Systems)

by

Jongkil Park

Committee in charge:

Professor Gert Cauwenberghs, Chair
Professor Peter M. Asbeck, Co-Chair
Professor Nuno M. Vasconcelos
Professor Kenneth Kreutz-Delgado
Professor Terrence J. Sejnowski

2014

The dissertation of Jongkil Park is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2014

DEDICATION

To my parents, Hoonkyu Park and Myoung-gui Lee
my brother Jongho Park, sister-in-law Kyungsun Oh,
and all of those who shared their lives with me

EPIGRAPH

If you fully obey the LORD your God and carefully follow all his commands I give you today, the LORD your God will set you high above all the nations on earth.

—Deuteronomy 28:1

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation would not have been possible without collaborations from many acquaintances and their support. Because of them, I was able to stay firm and calm during extremely nervous Ph.D. life. I thank God for bringing them into my life and involving them in my Ph.D. journey at UCSD. I cannot list all of them here, but I thank all who shared their lives with me at San Diego.

First of all, I would like to thank my academic advisor Gert Cauwenberghs for his guidance and full support. He taught me his brilliant idea, ways of research, and balancing my life. Most of all, he helped me pioneer into the research area of neuromorphic engineering. He was always patient and encouraged me when I was frustrated with lots of failures in my Ph. D. life. I would also thank my dissertation committee: Professor Peter M. Asbeck, Professor Kenneth Kreutz-Delgado, Professor Nuno M. Vasconcelos, and Professor Terrence J. Sejnowski. Their support and valuable feedback strengthened contents and qualities of this dissertation.

I thank my colleagues and friends who shared great discussions and coffee hours; Theodore Yu, Mike Chi, Siddharth Joshi, Sohmyung Ha, Chul Kim, Abraham Akinin, Cory Stevenson, Srinjoy Das, Bruno Pedroni, Chris Thomas, Dr. Christoph Maier, Dr. Emre Neftci, Dr. Frederic Broccard, Dr. Massoud L. Khraiche, and Dr. Sadique Sheik.

I also would like to thank all my friends and pastors from Blessing church in Korea and San Diego Onnuri church. All their encouragements and prayers for me strengthened my mind.

Finally, I thank my family who deserve to get all this honor and acknowledgements. I thank my parent, Hoonkyu Park and Myoung-gui Lee. Because they shared graceful sermons everyday, I was able to endure all these years. I also thank my brother, Jongho Park, and sister-in-law, Kyungsun Oh. Daily chats with them made me feel a strong connection with Korea all the time and relieve my homesickness.

Chapter 2 is largely a reprint of material that was accepted to 2014 Biomedical Circuits and Systems Conference : J. Park, S. Ha, C. Kim, S. Joshi, T. Yu, W. Ma and G. Cauwenberghs, “A 12.6 mW 8.3 Mevents/s Contrast Detection 128$\times$128 Imager with 75 dB Intra-Scene DR Asynchronous Random-Access Digital Readout”, *IEEE Biomed-*

ical Circuits and Systems Conference (BioCAS 2014), Oct 2014. The author is the primary author and investigator of this paper.

Chapter 3 is largely a reprint of material that was accepted to 2014 Biomedical Circuits and Systems Conference : J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-Neuron 73-Mevents/s 22-pJ/event Asynchronous Micro-Pipelined Integrate-and-Fire Array Transceiver", *IEEE Biomedical Circuits and Systems Conference (BioCAS 2014)*, Oct 2014. The author is the primary author and investigator of this paper.

Chapter 4 is largely a reprint of material that was submitted to 2014 Transaction on Neural Networks and Learning Systems : J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical Address Event Routing for Reconfigurable Large-Scale Neuromorphic Systems", *IEEE Transaction on Neural Networks and Learning Systems.* The author is the primary author and investigator of this paper.

VITA

2007	B. S. in Electrical Engineering *cum laude*, Korea University, Seoul, Korea
2010	M. S. in Electrical Engineering, University of California, San Diego
2014	Ph. D. in Electrical Engineering, University of California, San Diego

PUBLICATIONS

Jongkil Park, Sohmyung Ha, Theodore Yu, Emre Neftci, and Gert Cauwenberghs, "A 22-pJ/spike 73-Mspikes/s 130k-Compartment Neural Array Transceiver with Conductance-based Synaptic and Membrane Dynamics" manuscript in preparation for *IEEE Transaction on Biomedical Biomedical Circuits and Systems*

Jongkil Park, Theodore Yu, Siddharth Joshi, Christoph Maier, and Gert Cauwenberghs, "Hierarchical Address Event Routing for Reconfigurable Large-Scale Neuromorphic System" submitted to *IEEE Transaction on Neural Networks and Learning Systems*

Jongkil Park, Sohmyung Ha, Chul Kim, Siddharth Joshi, Theodore Yu, Wei Ma, and Gert Cauwenberghs, "A 12.6 mW 8.3 Mevents/s Contrast Detection 128×128 Imager with 75 dB Intra-Scene DR Asynchronous Random-Access Digital Readout", *IEEE Biomedical Circuits and Systems Conference (BioCAS 2014)*, Oct 2014

Jongkil Park, Sohmyung Ha, Theodore Yu, Emre Neftci, and Gert Cauwenberghs, "A 65k-Neuron 73-Mevents/s 22-pJ/Event Asynchronous Micro-Pipelined Integrate-and-Fire Array Transceiver", *IEEE Biomedical Circuits and Systems Conference (BioCAS 2014)*, Oct. 2014

Chul Kim, Sohmyung Ha, Chris Thomas, Siddharth Joshi, **Jongkil Park**, Lawrence Larson, and Gert Cauwenberghs "A 7.86 mW +12.5 dBm in-Band IIP3 8-to-320 MHz Capacitive Harmonic Rejection Mixer in 65nm CMOS", *IEEE European Solid-State Circuits Conference (ESSCIRC 2014)*, Sep. 2014

Sohmyung Ha, Chul Kim, **Jongkil Park**, Siddharth Joshi, and Gert Cauwenberghs, "Energy-Recycling Integrated 6.78-Mbps Data 6.3-mW Power Telemetry over a Single 13.56-MHz Inductive Link", *IEEE Symposia on VLSI Technology and Circuits 2014*, Jun. 2014.

Sohmyung Ha, **Jongkil Park**, Yu M. Chi, Jonathan Viventi, John Rogers, and Gert Cauwenberghs, "85dB Dynamic Range 1.2mW 156kS/s Biopotential Recording IC for High-Density ECoG Flexible Active Electrode Array", *IEEE European Solid-State Circuits Conference (ESSCIRC 2013)*, Sep. 2013.

Teresa Serrano-Gotarredona, **Jongkil Park**, Alejandro Linares-Barranco, Ryad Benosman, and Bernabè Linares-Barranco, "Improved Contrast Sensitivity DVS and its Application to Event-Driven Stereo Vision", *IEEE International Symposium on Circuits and Systems (ISCAS 2013)*, May. 2013

Theodore Yu, **Jongkil Park**, Siddharth Joshi, Christoph Maier, and Gert Cauwenberghs "65k-Neuron Integrate-and-Fire Array Transceiver with Address-event Reconfigurable Synaptic Routing", *IEEE Biomedical Circuits and Systems Conference (BioCAS 2012), Hsinchu, Taiwan,* Nov. 2012

Theodore Yu, **Jongkil Park**, Siddharth Joshi, Christoph Maier, and Gert Cauwenberghs "Event-driven Synchronous Neural Integration in Analog VLSI" *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* Aug. 28 $\sim$ Sep. 1, 2012

Jongkil Park, Theodore Yu, Christoph Maier, Siddharth Joshi, and Gert Cauwenberghs, "Hierarchical Address-Event Routing Architecture for Reconfigurable Large Scale Neuromorphic Systems", *IEEE International Symposium on Circuits and Systems (ISCAS 2012)* May. 2012

Siddharth Joshi, Steve Deiss, Mike Arnold, **Jongkil Park,** Theodore Yu, and Gert Cauwenberghs, "Scalable Event Routing in Hierarchical Neural Array Architecture with Global Synaptic Connectivity" *12th IEEE International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), Berkeley* Feb. 2010

ABSTRACT OF THE DISSERTATION

Large Scale Asynchronous Low-power VLSI Systems for Event-driven Sensory and Neural Processing

by

Jongkil Park

Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems)

University of California, San Diego, 2014

Professor Gert Cauwenberghs, Chair
Professor Peter M. Asbeck, Co-Chair

This dissertation investigates a low-power temporal event encoding imaging sensory system front end and a neural computation analog VLSI backend embedded within a custom scalable architecture enabling highly energy efficient processing of these event streams. We explore the differences in event encoding and conventional computing emphasizing that computation and communication are data-driven and energy costs scale with information transfer and processing. The application of this principle in the imaging sensory system increases efficiency by ensuring that light intensity information is gathered only when and where warranted by temporal change and spatial proximity. This temporal contrast detection imager having 128×128 pixel array die size of

5×5 mm^2 and pixel size of 33×33 μm^2 is fabricated in 0.18 μm CMOS. With supporting asynchronous event-driven information compression we achieved 1.52 nJ per pixel event detection and readout. Similarly for neural computation slow but densely arrayed neural units are fabricated on a 4×4 mm^2 die in 90 nm CMOS. We present a 65-k integrate-and-fire array transceiver (IFAT) on a single die implementing 65-k neurons each with two compartments and four conductance based programmable analog synapses at 18.2 Mevents/s per each quadrant at sustained peak synaptic event throughput and 22 pJ per synaptic input event in average. Operating at very low power the IFAT is robust to noisy inputs and high throughput is enabled by an asynchronous two-tier micro-pipelining scheme. This system is formed in a tree based hierarchical address event routing (HiAER) architecture. HiAER is implemented in 5 Xilinx Spartan-6 FPGAs enabling 262k neurons and 262M synapses on a level of hierarchy, at 3.6×10^7 synaptic events per second per each 16k-neuron node in the hierarchy.

Chapter 1

Introduction

1.1 Neuromorphic Engineering

Spiking neural networks implemented using neuromorphic circuits are promising research tools for investigating the computational abilities of the brain [30,52]. Their compact and low-power characteristics makes them ideal for emulating the brain's performance for robotic and mobile applications in real-world environments. Synthesis of very large-scale silicon models of biological neural networks approaching the computational complexity and cognitive function of the human brain has long posed a grand challenge in neuromorphic systems engineering and has been met with strengthened effort and enthusiasm in recent years. The challenge is not only one of massive scale in biological neural networks, with billions of neurons and trillions of synapses, but also in supporting flexible mechanisms to dynamically configure synaptic connectivity, driven by neural activity, across all scales.

The various approaches to build a large-scale neuromorphic processor range from custom built ARM cores integrated with specialized Network on Chip (NoC) routers [25, 62, 74], custom digital implementations with quasi-asynchronous elements helping to maintain synchrony [29, 53, 55] to standard OTA based neuron circuits with wafer scale integration and connectivity [58, 70], analog quadratic integrate-and-fire neuron sharing synapse, axon and dendrite with neighbor neurons implementing diffusive neural network as layered in cortex [4, 45], and subthreshold CMOS VLSI analog neurons with digitally controlled conductance based synapses [85].

1.2 Silicon Retina Modeling

Despite tremendous advances in semiconductor technology and in our understanding of the mammalian retina, todays imaging technology for artificial vision is far inferior to its biological counterpart. The mammalian retina is the gold standard in engineering design as the most efficient image processor with superior coding and energy efficiency. A low power silicon retina approaching some of the metrics of efficacy and efficiency of the mammalian retina is of critical importance for neuromorphic object recognition.

1.3 Integrate-and-Fire Array Transceiver

For modeling a neuron as computational basic unit in spiking neural networks, many depths of neural and synaptic dynamics exist. Depending on levels of modeling detail, these range from a model of ion channel kinetics with hundred of differential equations and parameters for biological plausibility [28], to models of simplified conductance-based differential equations for computational efficiency [32, 57]. The leaky integrated-and-fire neuron model is a popular choice for implementing large scale neuromorphic processor, because of its relative simplicity and its ability to emulate many dynamical features of biological neurons [9, 32].

The Integrate-and-fire array transceiver (IFAT) is intended as a large-scale and power efficent implementation of integrate-and-fire neurons. In previous work, the IFAT neuron was realized using analog switched capacitor techniques and included 2k neurons with large transistor feature size [27, 80]. The current IFAT includes single-transistor conductance based synapse, with first-order linear dynamics. The conductance-based feature enables more biological plausibility and the single-transistor implementation allows the integration of 65k neuron in a single chip [84]. In this dissertation, we present a fully asynchronous pipelined structured event communication, enabling low power and high synaptic throughput. This chip offers an ideal building block for large-scale hierarchical neuromorphic systems [64].

1.4 Event-driven Asynchronous System

Biological system has temporal sparsity of spike events such as average 10 Hz firing rate per each neuron. Neuromorphic engineering takes this idea to build power efficient computing hardware in analog VLSI. Temporal sparsity of spike events can be implemented in a system driven by events, asynchronous system. The asynchronous system is activated only when it needs to serve an event and idle when it doesn't have events request. It reduces system active time and dynamic power dissipation while clock based digital system requires these. Fully asynchronous design has an advantage of power consumption [50] achieving a sub-nonojoule energy efficient asynchronous micro-controller [51]. Also, in large scale neural processor design, fully asynchronous implementations [4, 53, 80] and globally asynchronous and locally synchronous [55, 58] approaches are presented.

1.5 Address Event Representation (AER)

In biological neural networks, action potentials (or "spikes") traveling along axons carry neural information over long distances projecting to large numbers of other neurons distributed over varying spatial scales [16, 76]. Naturally the question arises whether similar principles of distributed communication with spike "events" can be employed for efficient and scalable computation across large networks of silicon systems. The Address-Event Representation (AER) protocol was introduced as an efficient means for point-to-point communication of neural spike events between arrays of neurons, in which addresses of neurons are asynchronously communicated over a shared digital bus, whenever they spike [7, 18, 42, 49, 77]. The AER communication protocol lends itself directly to implementing synaptic connectivity in a dynamically reconfigurable manner by routing address events through synaptic routing tables in memory, which map presynaptic source addresses to postsynaptic destination addresses along with synaptic parameters [27, 31, 47, 72, 78, 80, 83].

The virtual wiring of AER synaptic connections between neurons, residing in programmable routing tables in memory, offers the flexibility to connect in principle any pair of neurons. Such is not generally possible with hardwired synaptic array re-

alizations, except for fully connected and hence relatively small networks of neurons. Furthermore, AER synaptic connections can be freely created, updated, and pruned as needed. In particular, Hebbian-like spike-timing dependent plasticity (STDP) and other forms of adaptive updates in synaptic strength and connectivity based on spiking neural activity can be conveniently implemented in the address-event domain, by monitoring relative timing of presynaptic and postsynaptic spike events entering and exiting the synaptic routing tables [81] or through more general forms of activity-dependent reprogramming of AER connectivity [1]. From a systems perspective, AER synaptic connectivity further permits multi-chip integration of spike event-based sensory and neural processing systems such as silicon retinae [8, 44, 48, 73], silicon cochleae [12, 41], and systems comprising them for various applications such as object recognition [11], accident detection [24], word recognition [69], texture recognition [66], sequence recognition [60], among many others.

One intrinsic challenge of AER synaptic connectivity for large-scale neuromorphic systems is the limitation in bandwidth of the digital bus shared among all time-multiplexed synapses. Advances in high-speed serial communication links using low-voltage differential signaling (LVDS) [5, 20, 88] support bus bandwidths up to 100 Mevents/s at 16 bits per event. With peak neural firing rates up to 100 Hz, the number of synapses shared per AER bus is thus limited to millions, or thousands of neurons for a typical 100-10,000 fan-out.

To mitigate this AER bandwidth limitation for very large-scale neuromorphic systems, several solutions have been proposed to extend the standard single-bus AER architecture using grid (or mesh) and tree inter-chip interconnect topologies. Neurogrid [4, 54, 56] employs linear grid and tree topologies in which global address events are broadcasted across chips through multiple point-to-point AER buses, leading to improvements in overall point-to-point communication channel bandwidth although with limited long-range connectivity. Two-dimensional grid topologies are also pursued in systems with differing address event mapping schemes. SpiNNaker [39, 62] assigns unique global addresses enabling direct neuron-to-neuron access across chips by implementing larger local routing tables. Multicasting Mesh AER [87] stores router-to-router connectivity rather than neuron-to-neuron connectivity in local routing tables, reducing

table size by implementing address translation for local neural event routing. Wafer-scale integration of 2-D AER grid multi-chip neuromorphic systems [21, 58] further mitigates communication cost in chip-to-chip interconnectivity issue by connecting 450 chips on a single wafer through metal post-processing. Inter-wafer communication extends such systems to another larger level for longer range interconnects [71].

Chapter 2

Event-driven Temporal Contrast Detection Imager

2.1 Introduction

Event-driven dynamic vision sensors with temporal change detection [6, 14, 43, 44, 68, 73] inspired by the biological retina enable effective high speed image capture at reduced cost in power consumption and communication bandwidth. An important attribute of retinal modeling that is often missing in event-driven dynamic vision sensor designs is the range of spatiotemporal dynamics of signals feeding into the optic nerve, which include both sustained and transient ganglion cell responses to complex visual stimuli. Zaghoul et al [86] developed a silicon retina that accounted for several of the spatiotemporal attributes of a range of ganglion cells, but at the expense of relatively large power consumption, almost a factor thousand times larger than that of the mammalian retina. Other silicon retina designs have abstracted the biological model to highly simplified spatial and/or temporal dynamics that lend to more efficient implementation of a few attributes for practical use in a particular application. For modeling temporal dynamics, both asynchronous [6, 43, 44, 68, 73] and inter-frame change based [14] architectures for temporal change detection have been presented. For instance, Lichtsteiner et al [44] presented an event-based image sensor for rapid change detection and coding. This silicon retina modeled three key properties of biological vision: sparsely

coded event-based output, representation of relative luminance change, and rectification of positive and negative signals into separate output channels.

The lack of an on-chip ADC in purely event-based imager designs increases the post-processing requirements on down stream signal processing. The presented work enables fully asynchronous detection of temporal intensity change with simultaneous random-access asynchronous ADC readout of intensity, facilitating efficient postprocessing for spatiotemporal coding, *e.g.,* [13]. Log-encoding of the intensity readout with current-domain correlated double sampling provides 75 dB intra-scene dynamic range.

2.2 Pixel and System Design

The dual-port architecture of the imager, providing independent simultaneous asynchronous data streams for temporal contrast event registration and random-access intensity readout, is illustrated in Fig. 2.1 (a). Row and column arbiters register addresses of output events coding threshold detection of temporal changes in pixel intensity. Separate positive and negative thresholds for increasing and decreasing temporal changes determine "on" and "off" output events, respectively. Linear feedback shift register (LFSR) based arbitration of multiple simultaneous events aids in equalizing coverage of event registration. Separate row and column decoders provide Independent addressing of pixel selection for random-access asynchronous intensity readout. Independent control over pixel location and readout time, separate from registered change events, allows for spatiotemporal coding in visual processing, e.g., detection of motion, or motion-based video compression. The selected pixel intensity readout is quantized to 10-bit digital output using an on-chip asynchronous ADC.

The simplified schematic of the pixel is given in Fig. 2.1 (b). Owing to logarithmic relationship between photodiode current and voltage through MOS transistors operating in subthreshold, voltage amplification is required in order to resolve a detectable temporal change input to the comparator [6, 44, 68]. High-gain amplification normally requires a wide range in capacitance values, reducing the available area in the pixel for reasonable fill factor. An alternative approach in [73] employs transduction

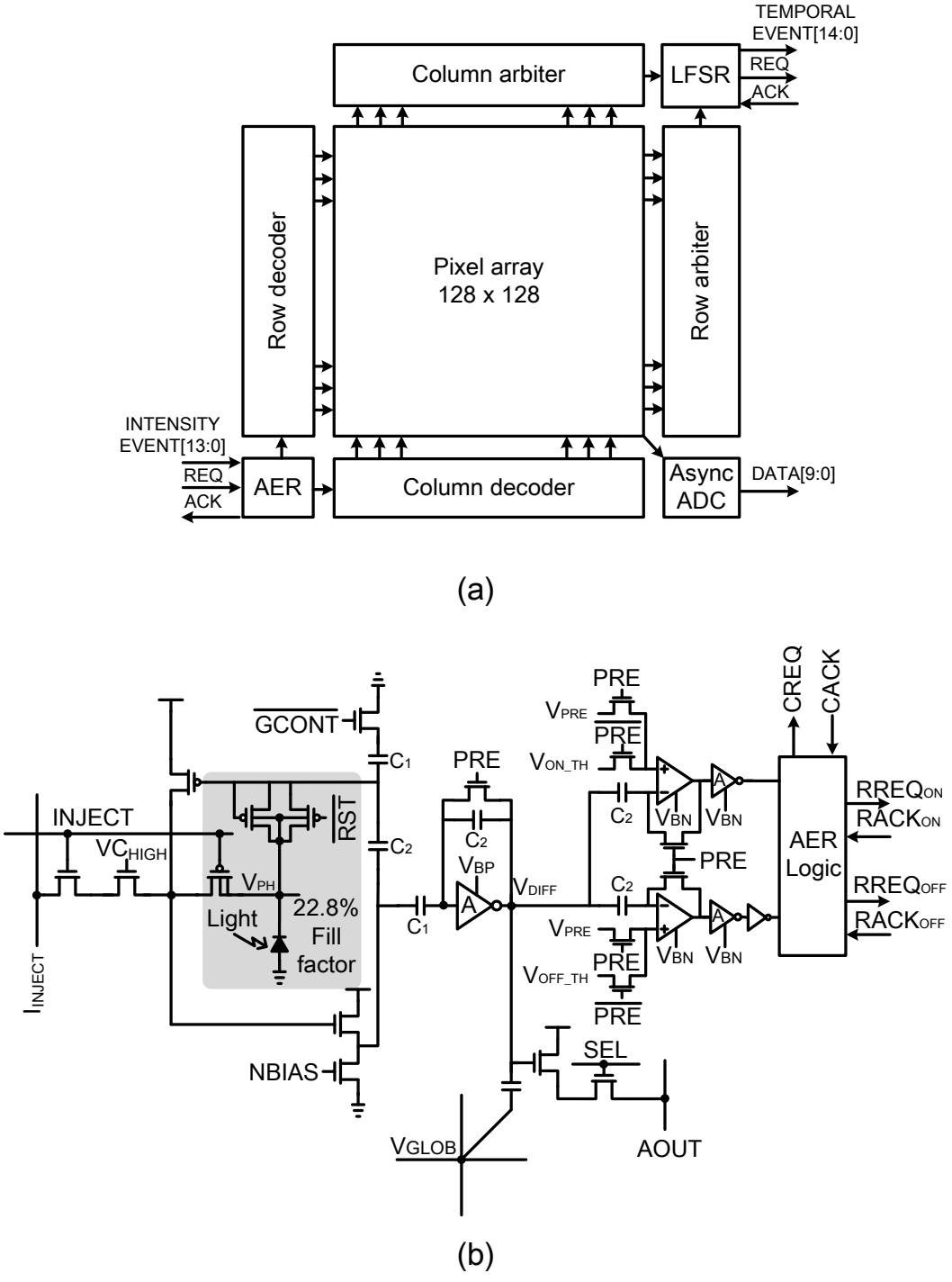


Figure 2.1: Asynchronous imager with temporal contrast threshold detection and simultaneous random-access digital readout. (a) System diagram, and (b) pixel circuitry.

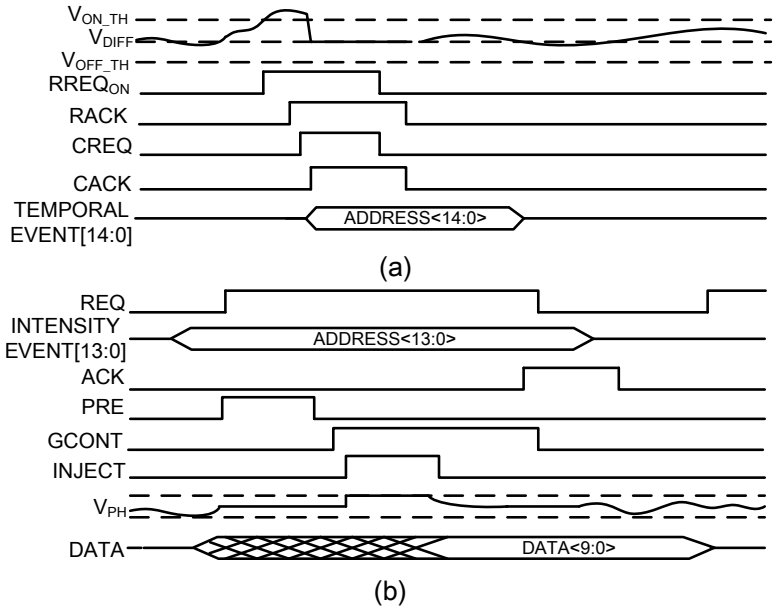


Figure 2.2: Timing diagram of (a) asynchronous temporal contrast event detection and (b) random-access intensity readout.

and preamplification stages for higher overall gain while reducing capacitive ratio and hence area. The current design implements two-stage amplification of the logarithmic phototransduction, with offset compensation in the amplifiers through current-domain correlated double sampling (CDS), and further voltage-domain CDS in the comparators for on/off event detection.

The first amplifier stage encompassing the pMOS transistor load of the photocurrent contributes $1+C_1/C_2$ gain. A source following internal to this amplifier buffers the high-impedance photosensitive node prior to the $C_1||C_2$ capacitive load. The subsequent inverting amplifier stage provides additional gain with factor C_1/C_2. The sensitivity of the two-stage amplified voltage output V_{DIFF} to relative change in pixel photo current I_{ph} is thus given by:

$$dV_{\mathrm{DIFF}} = \frac{C_1}{C_2}(1+\frac{C_1}{C_2})\frac{V_T}{\kappa}\frac{dI_{ph}}{I_{ph}} \tag{2.1}$$

where V_T is the Boltzmann thermal voltage and κ is the back-gate efficiency coefficient of the pMOS transistor load. A modest capacitance ratio $C_1/C_2 = 2$ yields 6-fold ampli-

fication in the logarithmic photoresponse, with a 2.1 mV voltage step for a 1% change in light intensity at room temperature.

Correlated double sampling (CDS) in the second gain stage and subsequent on/off event comparators reduces sensitivity to voltage offset due to transistor mismatch at low change threshold values, while further allowing to dynamically reference the signal to the instantaneous intensity value upon the preset phase (PRE). On/off events are registered through address-event representation (AER) logic and routed off-chip through fully asynchronous arbitration circuits. Self-timed event coding and registration is illustrated in Fig. 2.2 (a).

In addition, a current-domain divisive-normalizing form of CDS provides compensation for transistor mismatch and temperature variations in intensity readout. The current-domain divisive normalization is obtained by CDS subtraction of the pMOS logarithmic photoreceptor voltage response between signal and reference phases. To provide the reference, a single global current reference I_{INJECT} is steered into the pMOS load of the selected pixel, bypassing the photocurrent, upon activation of INJECT. The voltage difference is further CDS-sampled in the ADC. The timing control for current-domain CDS, with reference current injection following sampling of the photosignal, is illustrated in Fig. 2.2 (b). The signal GCONT in Fig. 2.2 (b) temporarily shunts the $1+C_1/C_2$ gain of the first stage in Fig. 2.1 (b) to unity, preventing the dynamic range of the photodiode current from saturating the readout and ADC. The resulting voltage difference across CDS signal and reference phases is thus

$$V_{\mathrm{DIFF,CDS}} = -\frac{C_1}{C_2}\frac{V_T}{\kappa}\log\frac{I_{ph}}{I_{\mathrm{INJECT}}} \tag{2.2}$$

independent of offsets in pixel amplification and readout source follower stages.

Global adaptation of the event detection thresholds to uniform temporal variations in light intensity, such as necessary under fluorescent and incandescent in-room lighting, is performed through tracking of the common-mode of intensity signal V_{GLOB} node as a reference to dynamically adjusting the thresholds with unity gain. The V_{GLOB} global intensity signal is obtained by unity-gain capacitive coupling to the V_{DIFF} nodes across all pixels as shown in Fig. 2.1. The buffered V_{GLOB} node is also used for external monitoring of the common-mode intensity signal.

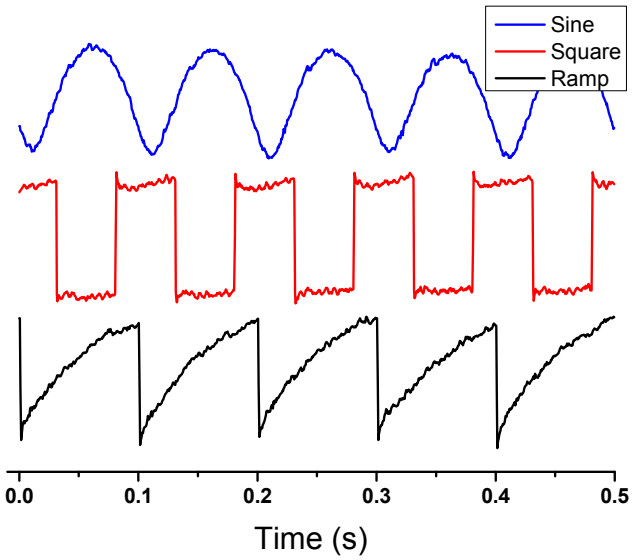


Figure 2.3: Common-mode global voltage output V_{GLOB} under uniform lighting with sine, square and ramp generated intensity profiles at 90% modulation depth. Voltage encodes the logarithm of intensity.

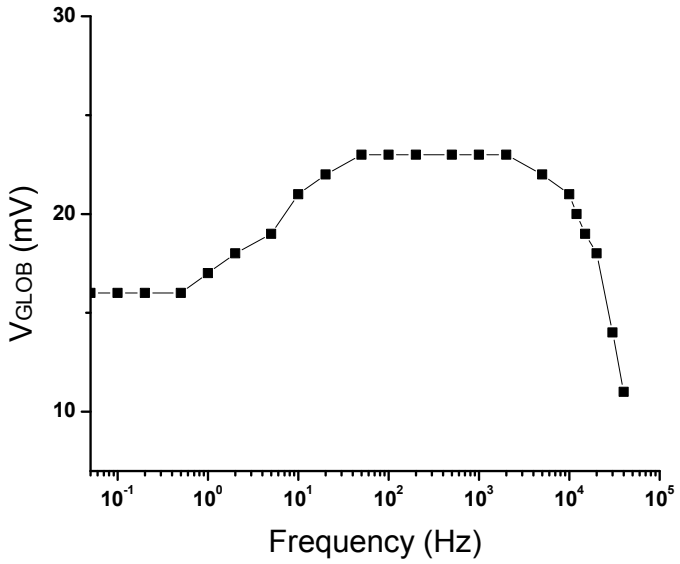


Figure 2.4: Frequency response of common-mode global voltage output V_{GLOB} under variable frequency uniform lighting at 90% modulation depth.

2.3 Experimental Results

Fig. 2.3 shows the V_{GLOB} common-mode intensity output under varying uniform illumination stimuli controlled with an LED array in front of the bare imager, and Fig. 2.4 shows the frequency response.

Fig. 2.5 (a) demonstrates the intra-scene dynamic range of the imager, mounted with a 5.7 mm effective focal length, F/1.6 lens. The image shows a bright light bulb of approximately 700 lux and an LED of approximately 40 lux visible within the same scene. Current-domain CDS enables an intra-scene dynamic range up to 75dB in the ADC output.

The change detection event-driven operation of the imager is illustrated in Fig. 2.6. A printed pattern of vertical stripes was used as stimulus shown in Fig. 2.6 (a) and moved in front of imager. In absence of movement, no events are registered except for sparse noise events in Fig. 2.6 (b). During left or right motion of the stripe stimulus in Fig. 2.6 (c) and (d), registered "on" and "off" events align at the leading and trailing edges as expected.

Fig. 2.7 shows the IC micrograph and pixel layout. Die size is 5mm x 5mm including pads and pixel size is 33um x 33um with 22.8% fill factor.

2.4 Conclusions

Table 2.1 summarizes the measured performance of the chip in relation to the state of the art. The 128×128 pixel with on-chip asynchronous 10-bit ADC offers random-access digital readout having 75 dB intra-scene dynamic range, operating at 12.6 mW power at 8.3 Mevents/s, or 1.52 nJ per pixel event detection and readout.

Chapter 2 is largely a reprint of material that was accepted to 2014 Biomedical Circuis and Systems Conference : J. Park, S. Ha, C. Kim, S. Joshi, T. Yu, W. Ma and G. Cauwenberghs, "A 12.6 mW 8.3 Mevents/s Contrast Detection 128×128 Imager with 75 dB Intra-Scene DR Asynchronous Random-Access Digital Readout", IEEE Biomedical Circuits and Systems Conference (BioCAS 2014), Oct 2014. The author is the primary author and investigator of this paper.

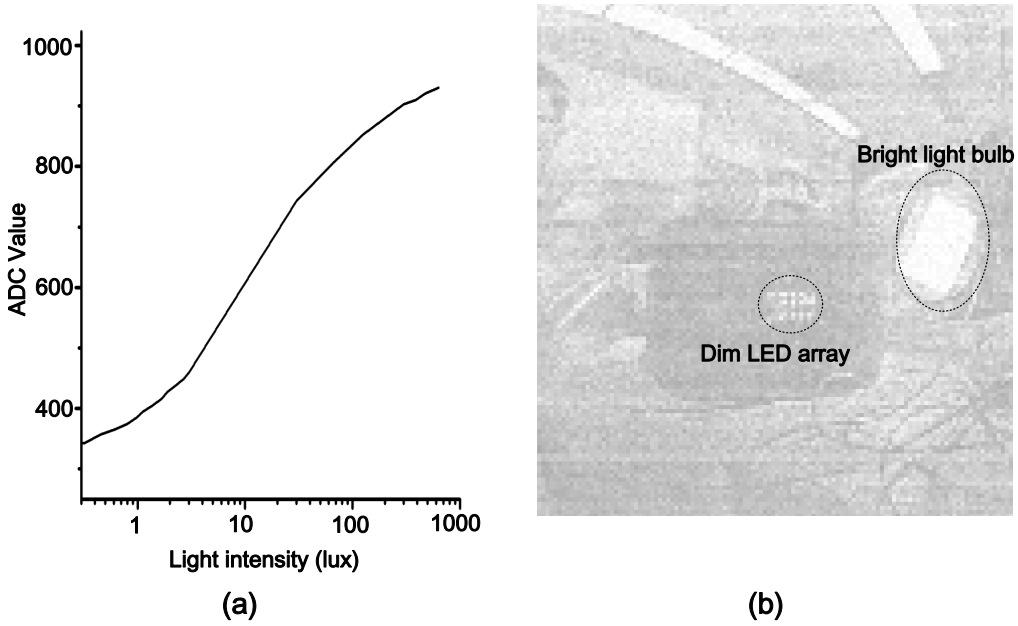


Figure 2.5: (a) Intra-scene dynamic range and (b) sample image of frame-scanned intensity readout.

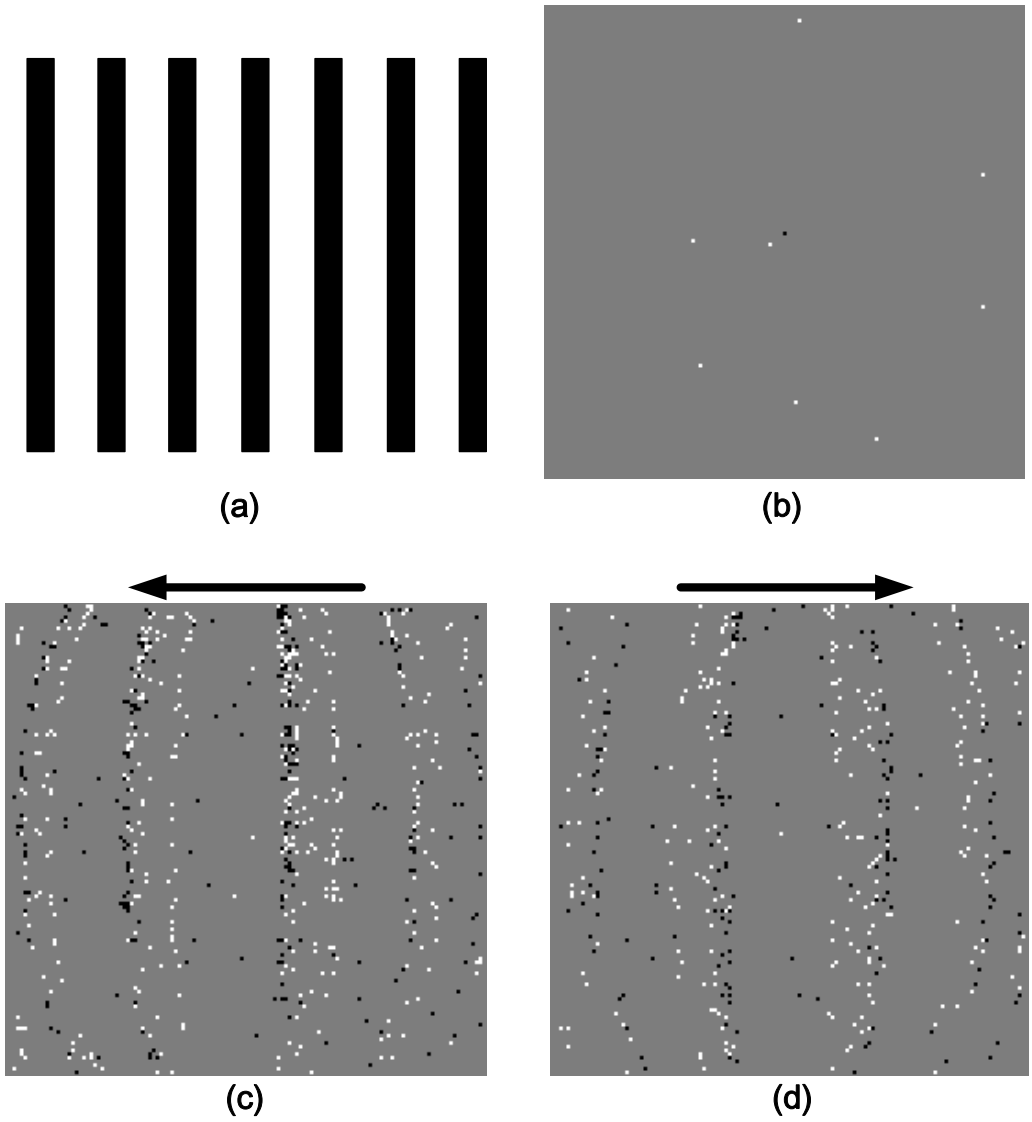


Figure 2.6: Temporal contrast change threshold detection. (a) Striped print pattern presented to the imager under indoors lighting (5 lux). (b) Received "on" and "off" events (shown as white and black dots, respectively, over gray background) to static presentation of the striped print pattern over a 30 ms time window. (c) Same, during leftward motion of the print ($-10\,^{\circ}$/s), and (d) during rightward motion ($+10\,^{\circ}$/s).

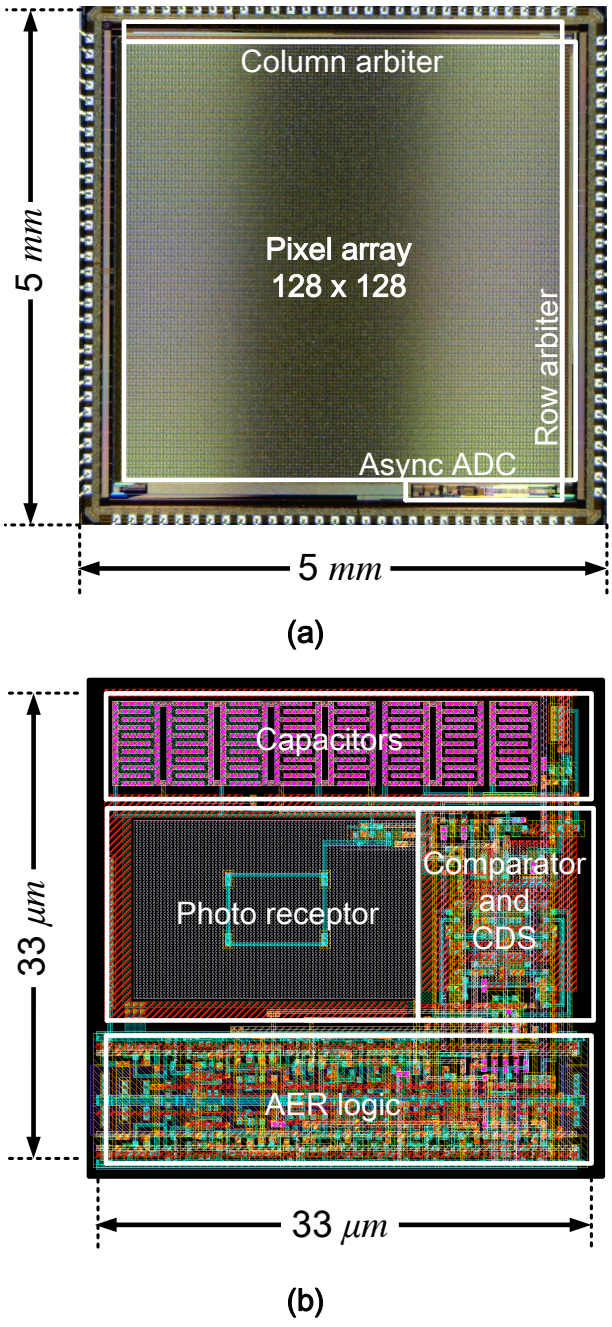


Figure 2.7: (a) Chip micrograph and (b) pixel layout.

Table 2.1: Imager Characteristics and Performance Summary

Reference	[44]	[68]	[73]	[14]	[43]	[6]	This work
Technology	0.35 μm	0.18 μm 6M 1P	0.35 μm 4M 2P	0.5 μm 3M 2P	0.35 μm 4M 2P	0.18 μm 6M 1P	0.18 μm 6M 1P
Resolution	128×128	304×240	128×128	90×90	128×128	240×180	128×128
Chip area (mm^2)	6×6.3	9.9×8.2	4.9×4.9	3×3	5.5×5.6	5×5	5×5
Pixel area (μm^2)	40×40	30×30	30×31	25.2×25.2	35×35	18.5×18.5	33×33
Fill factor (%)	8.1	10-20	10.5	17	8.7	22	22.8
Temporal contrast sensitivity (%)	15	13	1.5	2.1	10	12	8
FPN (%)	2.1	-	0.9	0.5 - 1.5	4.0	1	1.9
Dynamic range	120 dB	120 dB	120 dB	51 dB	$>$ 100 dB	120dB DVS 57dB APS	120 dB (75 dB intra-scene)
Peak event rate (Meps)	1	N/A	20	N/A	20	30	8.3
Intensity readout	N/A	Synchronous TDC (off-chip)	N/A	Single slope ADC (on-chip)	N/A	ADC (off-chip)	Asynchronous ADC (on-chip)
ADC resolution (bits)	N/A	8	N/A	6	N/A	8	10
Power consumption	24 mW	50-175 mW	4 mW @ 100 keps	4.2 mW @ 30 fps	132-231 mW	7.4-13.5 mW	12.6 mW @ 8.3 Mevents/s

Chapter 3

A 65,536 Neuron Asynchronous Micro-Pipelined Integrate-and-Fire Array Transceiver

3.1 Introduction

Here, we presents a 65,536-neuron integrate-and-fire array transceiver (IFAT) as a building block for large-scale hierarchical neuromorphic systems [64]. This chapter extends a previous report [63] with additional measurements and characterizations of the entire array of neurons. In Section 3.2, we describe circuit implementation. Conductance based synapse dynamics [83] motivating pulse width and amplitude modulation (PWAM), two-compartment integrate and fire neuron model, fully asynchronous address event routing and registration and a two-tier micro-pipelining scheme are described. In Section 3.3, we present measurement results. We demonstrated peak throughput enabled by a two-tier micro-pipelining circuit and power efficiency presented as synaptic event per energy. Also, we characterized representative neuron dynamics and variability over neuron array core. Finally, Section 3.4 summarizes related and prior works and concludes with a discussion on IFAT.

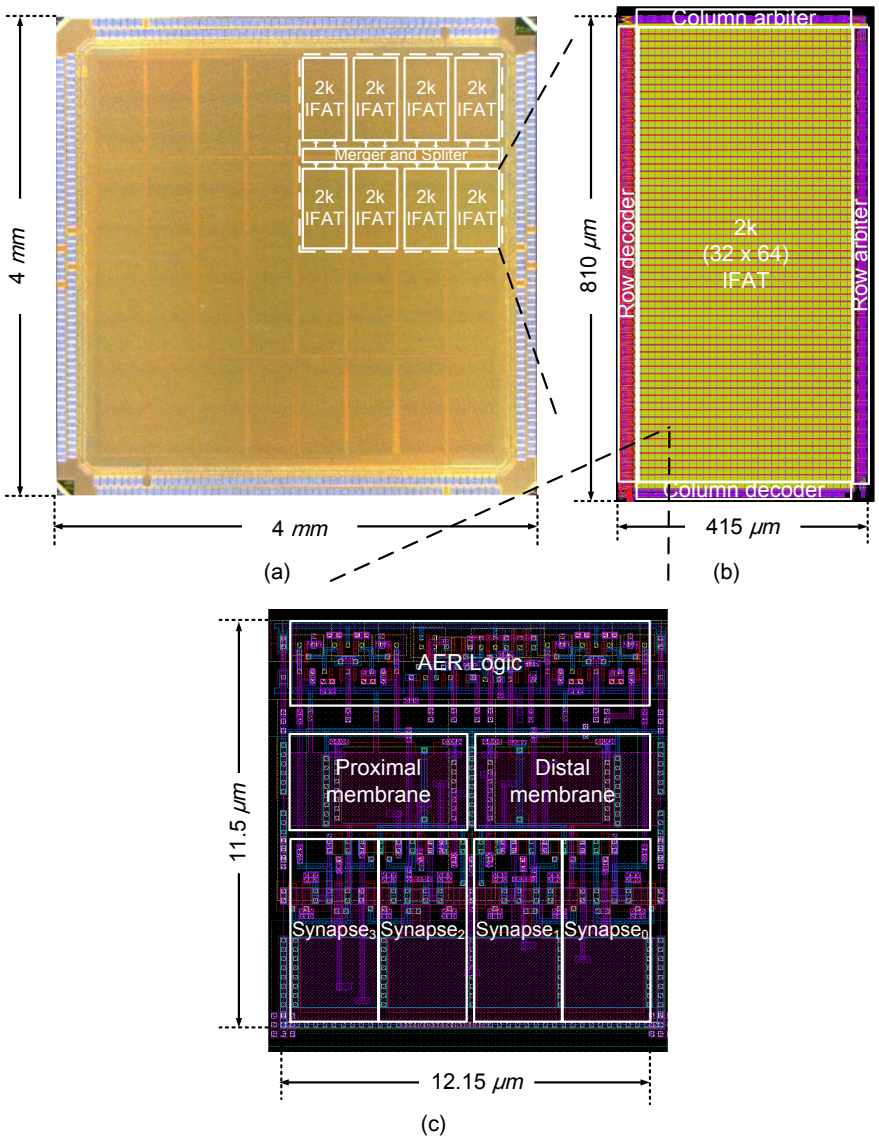


Figure 3.1: (a) Chip micrograph. One quadrant is indicated containing eight 2k-neuron IFAT core arrays. (b) 2k-neuron IFAT core and (c) two-compartment neuron cell layout.

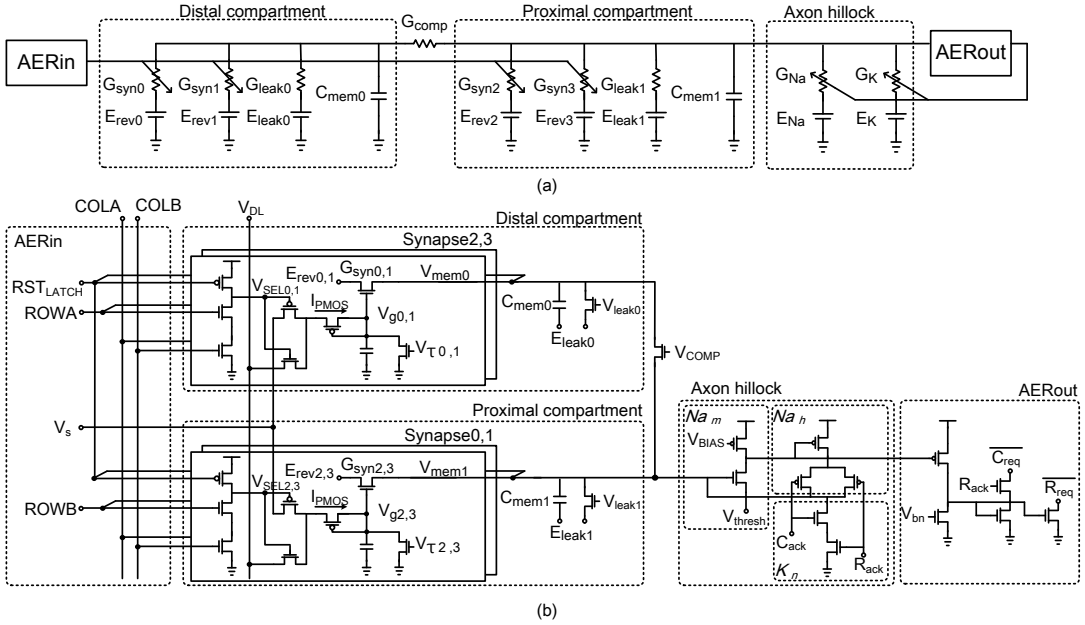


Figure 3.2: (a) Two compartmental leaky integrated-and-fire neuron model with conductance-based synapse. (b) Implemented two-compartment conductance based neuron circuit. A three-transistor dynamic latch holds V_{SEL} to select one active synapse element on the row while a pulse width modulated input at voltage V_S activates the synapse. Proximal and distal conductively leaky membrane compartments, each with two synapse circuits, are conductively coupled. An axon hillock circuit [80] generates and registers action potential output events resetting the proximal compartment membrane voltage V_{mem}.

3.2 Implementation

The 4×4 mm^2 IFAT chip was fabricated in a 90-nm CMOS process. With 436 staggered I/O pads, it was packaged in a 35×35 mm^2 body with 1.27-mm pitch Fine Ball Grid Array (FBGA). Each two-compartment neuron occupies 12.15×11.5 μm^2. The chip micrograph and layouts of the 2k neuron IFAT core and the neuron cell are shown in Fig 3.1.

3.2.1 Two-compartment Integrated-and-fire Neuron Model

The neuron is modeling a two-compartment leaky integrated-and-fire neuron, shown in Fig. 3.2 (a), as following equation.

$$C_{mem1}\frac{dV_{mem1}}{dt} = I_{fb} + \sum_{j=2,3} G_{syn,j}(E_{rev,j} - V_{mem1}) + G_{leak1}(E_{leak1} - V_{mem1}) + G_{comp}(V_{mem0} - V_{mem1}) \quad (3.1)$$

where C_{mem1} is proximal membrane capacitance, V_{mem0} and V_{mem1} are the distal and proximal membrane voltages, respectively, I_{fb} is the nonlinear positive feedback current due to the spiking mechanism, G_{syn} is synapse conductance, E_{rev} is reversal potential voltage, G_{leak} is leak conductance, E_{leak} is leak voltage, and G_{comp} is compartment conductance. The distal membrane voltage V_{mem0} follows similar dynamics.

The neuron circuit is shown in Fig. 3.2 (b) implementing a two-compartment conductance-based integrate-and-fire neuron. Each compartment is tied to two conductance based synapse circuits with programmable reversal potentials E_{rev} and time constants V_τ. The incoming event selects one of four synapse circuits through pairwise complements ROWA, ROWB and COLA, COLB. V_{SEL} holds the active low selection of one active synapse across the row, driving its pMOS diode-connected input with source voltage V_s to increment synaptic conductance in the log-domain, implementing a linear dynamical synapse with variable time constant set by V_τ [84]. After a pulse width Δt, V_s returns to ground, and RST_{LATCH} is activated to release V_{SEL} passive high, readying the row for activation of the next synaptic input event. Separate proximal and distal membrane compartments integrate currents from the synaptic, leak, and coupling conductances in continuous time. An AER self-timed axon hillock circuit [80] fires an action potential when the proximal membrane voltage reaches a threshold, registering a neural event on the output AER bus and resetting the membrane potential.

3.2.2 Conductance-based Synapse

Fig. 3.3 shows single transistor implementation of conductance based synapse presented in [83]. It is formulated from the drain current of nMOS transistor operating

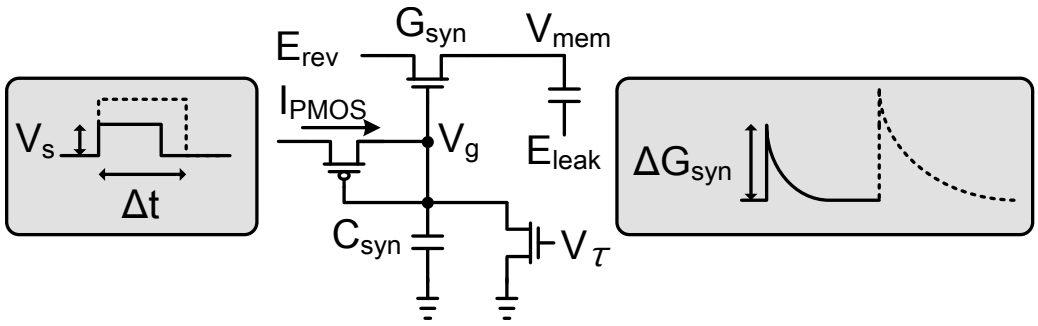


Figure 3.3: Implementation of a synapse with single-transistor log-domain conductance [83]. The synapse conductance update ΔG_{syn} is determined by the input pulse width Δt and its amplitude V_s according to Eq. (3.6).

in sub-threshold regime as follows:

$$I_d = I_0 e^{\frac{\kappa V_g}{V_T}} (e^{\frac{-V_s}{V_T}} - e^{\frac{-V_d}{V_T}}) \tag{3.2}$$

where I_0 is the transistor's dark current, V_g is the gate voltage, V_d is the drain voltage, V_s is the source voltage, κ is the back gate parameter, and V_T is the thermal voltage. It can be transformed to 'log-domain' or 'pseudo-voltage domain', with definition of a pseudo-voltage and pseudo-conductance [23].

$$I = G_{syn}(E_{rev} - V_{mem}) \tag{3.3}$$

where pseudo parameters of conductance $G_{syn} = \frac{I_0}{V_T} e^{\frac{\kappa V_g}{V_T}}$, pseudo parameters of reverse potential $E_{rev} = -V_T e^{(-\frac{V_d}{V_T})}$, and pseudo parameters of membrane potential $V_{mem} = -V_T e^{(-\frac{V_s}{V_T})}$.

From the pseudo parameter of conductance, we can derive synaptic conductance update with respect to time domain.

$$\begin{aligned}\frac{d}{dt}G_{syn} &= \frac{I_n}{V_T}\frac{d}{dt}e^{\frac{\kappa V_g}{V_T}} \\ &= I_n\frac{\kappa}{V_T^2}e^{\frac{\kappa V_g}{V_T}}(\frac{d}{dt}V_g) \\ &= I_n\frac{\kappa}{V_T^2}e^{\frac{\kappa V_g}{V_T}}\frac{I_{pmos}}{C_{syn}}\end{aligned} \tag{3.4}$$

$$I_{pmos} = I_p e^{\frac{V_s}{V_T}} e^{-\frac{\kappa V_g}{V_T}} \tag{3.5}$$

$$\Delta G_{syn} = \frac{\kappa I_n I_p}{V_T^2 C_{syn}} e^{\frac{V_s}{V_T}} \Delta t \tag{3.6}$$

where I_n and I_p are the sub threshold pre-exponential current factor of nMOS and pMOS, and C_{syn} is synapse capacitor.

The synaptic strength is pulse width Δt and amplitude modulation V_S (PWAM) encoded, and the resulting step in synaptic conductance ΔG_{syn} is approximately given by:

$$\Delta G_{syn} \propto (1+\frac{W}{16})2^A \tag{3.7}$$

where:

1. W encodes relative pulse width over baseline, representing the mantissa as given in integer units (0..15) by the 4-bit LSBs of the 8-bit digital synaptic strength; and

2. A encodes pulse amplitude in the log-domain, representing the exponent in integer units (0..15) as given by the 4-bit MSBs of the digital synaptic strength.

3.2.3 Overall Architecture

The IFAT uses the address event representation (AER) [7, 18, 42, 49, 77] of neuron locations on 2-D arrays to route synaptic input events into each 2k-neuron IFAT core

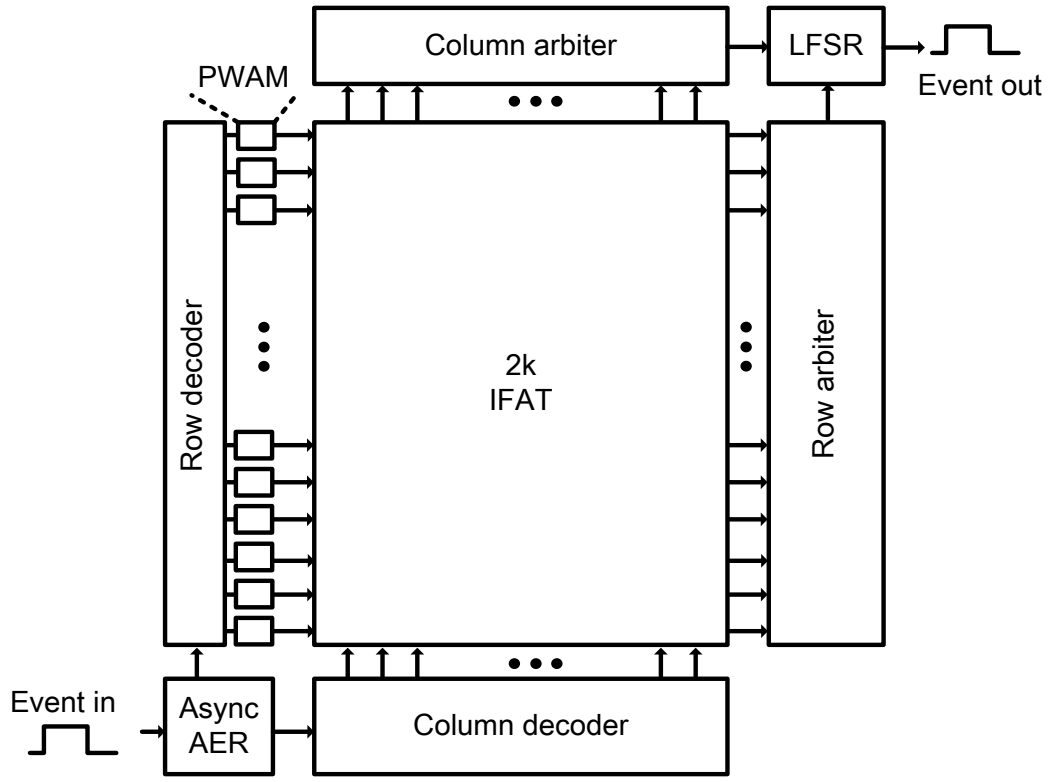


Figure 3.4: 2k-neuron Integrate-and-Fire Array Transceiver (IFAT) core with row and column decoders for input synaptic events, and row and column arbiters for output neural spike events arbitration. An event-triggered linear feedback shift register (LFSR) generates pseudo-random sequences in arbitrating multi-scale dyads of coincident output events (not presented here).

array, and register neural spike events out of each array, using separate input and output AER asynchronous digital buses. Each synaptic input event carries information on neuron address, synapse type, and synaptic strength. Previous pulse width modulation for synaptic strength implemented in synchronous AER logic incurred long wait times between subsequent events into each 2k-neuron core [83]. To mediate its limitation on throughput of the input event stream while further extending dynamic range of synaptic strength, an additional pipeline stage with pulse width and amplitude modulation (PWAM) is implemented for each row in the 2k-neuron core.

Fig. 3.4 shows the circuit implementation of the 2k-neuron integrate-and-fire transceiver (IFAT) core. Each 2k-neuron IFAT core includes an asynchronous AER communication circuit, PWAM circuits, decoders, and arbiters surrounding the array. Fully asynchronous communication with four-phase dual-rail encoding is implemented in input and output AER buses.

3.2.4 Four-phase Dual-rail Encoding Asynchronous Interface

In asynchronous designs, valid data transfer between sender and receiver needs to be guaranteed in any cases. While synchronous designs use a master system clock for data synchronization, asynchronous designs have a reliable data communication protocol mediated by "handshaking" with request and acknowledge signals. Among handshaking protocols [50], four-phase dual-rail encoding protocol is implemented in current chip. "Four-phase" means that it requires four phases of request and acknowledge sequences. "Dual-rail" means that each bit is encoded in two lines for delay-insensitive operation. This protocol requires C-elements (or Muller circuits) [59], each of which holds the output value until it receives the same value in both inputs. The circuit implementation, schematic symbol and truth table of C-elements are presented in Fig. 3.5 (a) and (b).

Circuit implementation of N-bit asynchronous pipeline stage with four-phase dual-rail encoding protocol is shown in Fig. 3.5 (c). Four-phase dual-rail protocol does not have explicit request signal but dual-bit line embed it. Each bit of data is encoded in two lines, TRUE and FALSE. TRUE bit represents actual bit value of the data and FALSE bit is complementary of it. If TRUE and FALSE have different values, these represent

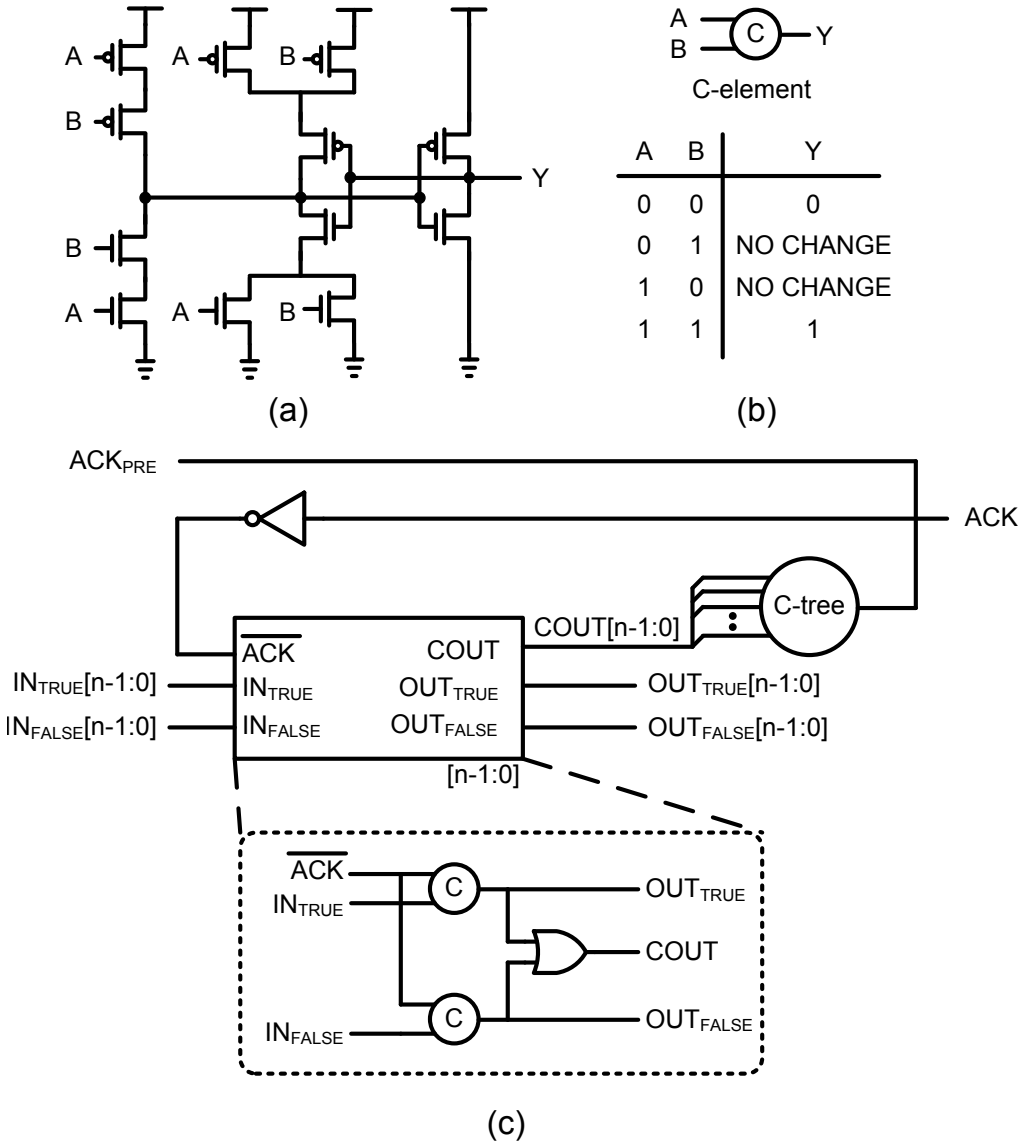


Figure 3.5: (a) Circuit implementation and (b) schematic symbol and truth table of C-element (or Muller circuit). (c) N-bit asynchronous pipeline stage. One bit representation in dual-rail encoding scheme with C-elements is shown in dashed box. Input data are latched when current stage is available, ACK is low. Completion tree, C-tree noted in figure, is tree of C-elements. It decides data packet is latched properly and acknowledge to previous stage.

valid bit value as in TRUE and a request signal to next stage. On the other hand, if both are same values, it means that bit lines are transitioning and are not representing valid data. Completion tree, C-tree (tree of C-elements) shown in Fig. 3.5 (d), validates all bit lines are properly latched and acknowledge to previous stage.

3.2.5 Asynchronous Splitter and Merger

64k neurons are arranged in four identical quadrants of 16k neurons, each quadrant containing eight 2k-neuron IFAT cores. 16k-neuron quadrant shares one digital input bus and one output bus. It requires asynchronous splitter to distribute and merger to combine address events properly on shared bus. An asynchronous splitter, implemented with chain of asynchronous pipeline stage shown in Fig. 3.5, broadcasts input synaptic event to each 2k neuron core. An asynchronous merger multiplexes neural spike outputs from each neuron core.

Input synaptic events are encoded as 24-bit address event, consisting of 3-bit of destination array, 13-bit of address and synapse type in 2k neuron core, and 8-bit of synapse strength. Asynchronous splitter decodes request signal from 3-bit MSB of input address events indicating destination array.

Output neuron spikes are encoded as 11-bit address event, representing a neuron address out of 2k neuron core, at 2k neuron core output stage. Asynchronous mergers are placed in diadic fashion, where two paths are sharing an output bus. One MSB is added on address event when it passes through each stage of merger. After event passes through three stages from each array to output bus, all three bits of MSB indicate one of arrays in a 16k-neuron quadrant. Therefore, an address event, which goes to micro controller, is a 14 bit address event representing one neuron out of 16k neurons. Fig. 3.6 shows an arbitration circuit and one stage of N-1 bit asynchronous merger implementation. Arbitration circuit (Fig. 3.6 (a)) consists of two cross-coupled NAND gates and arbitrates two simultaneous events from two paths. This selects request signal, which will be acknowledged. This selection represents one MSB that will be added on address event after event passes through this stage. One stage of N bit asynchronous merger is shown in Fig. 3.6 (b) and consists of an arbitration circuit and N bit asynchronous pipeline stage in Fig. 3.5 (b).

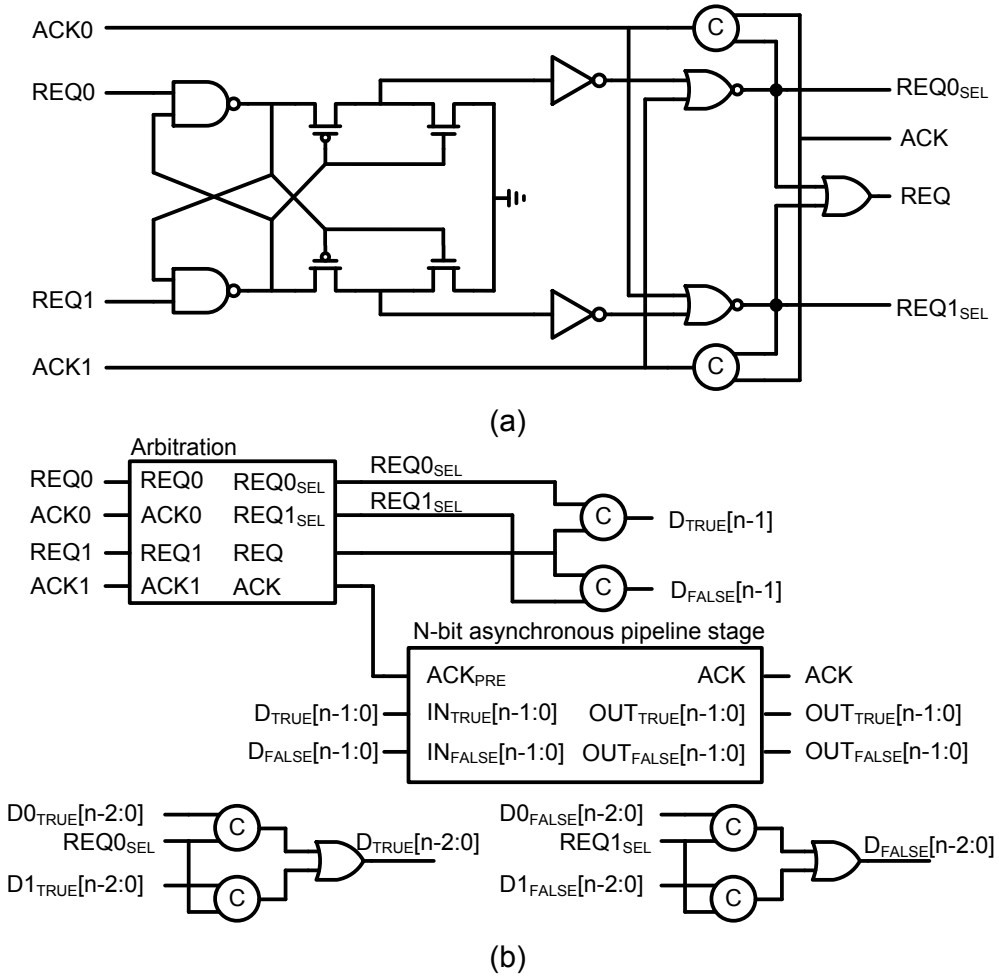


Figure 3.6: (a) Arbitration circuit consisting of two cross-coupled NAND-gates. (b) Asynchronous merger circuit consisting of arbitration circuit shown in (a) and N-bit asynchronous pipeline stage (shown in Fig. 3.5 (c))

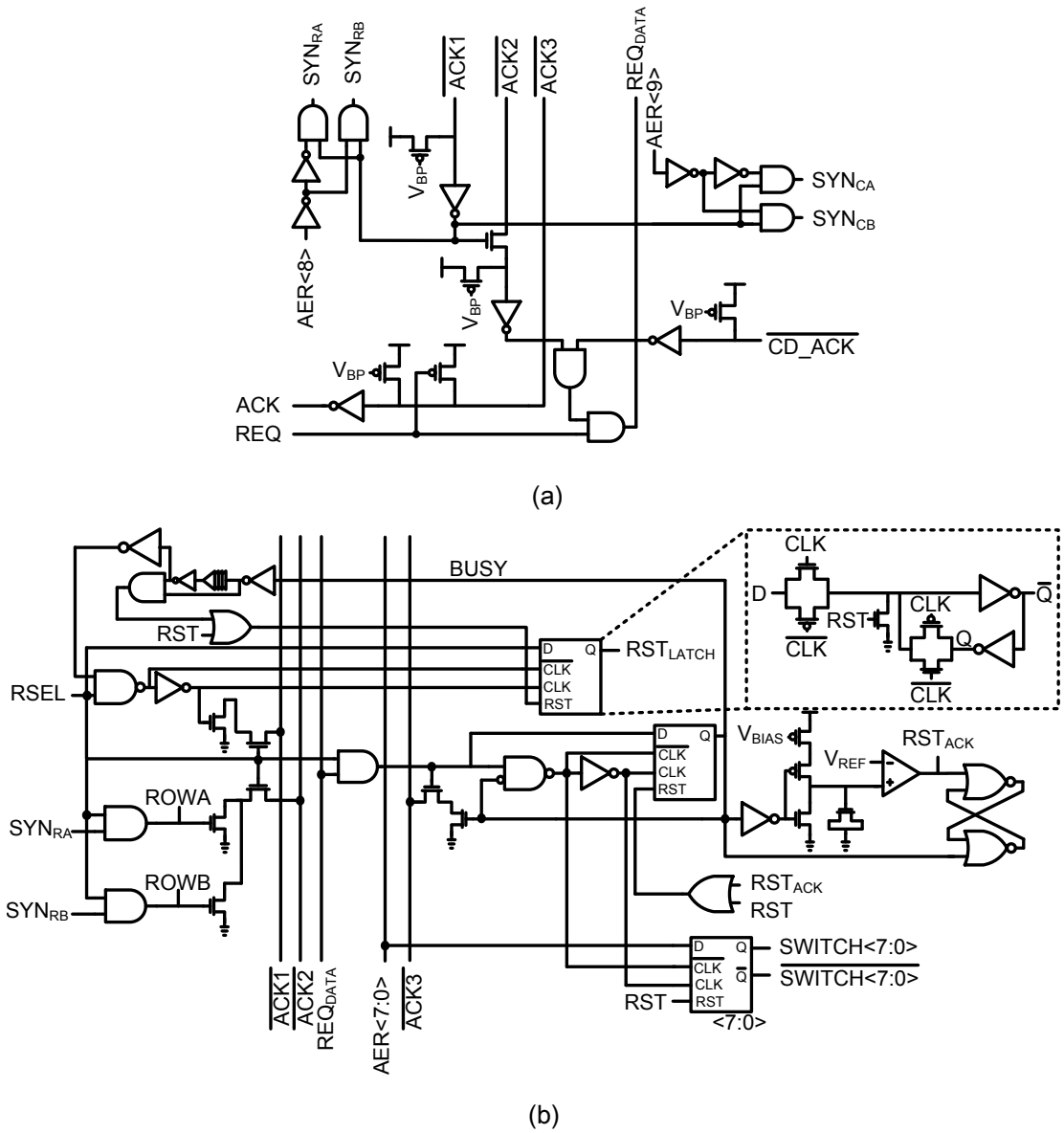


Figure 3.7: (a) Input asynchronous AER distribution network coordinating row-wise pulse width and amplitude modulation (PWAM) of synaptic strength, using 21-bit wide AER address, of which AER<9:8> determine synapse type. (b) Single-row PWAM circuit schematic. AER<7:0> contains the synaptic strength delivered with the REQ_{DATA} These bits are latched onto the SWITCH<7:0> bus. The 4 LSBs select the comparator reference voltage V_{REF} defining the pulse width of the synaptic stimulus, while the remaining 4 MSBs define its amplitude V_s

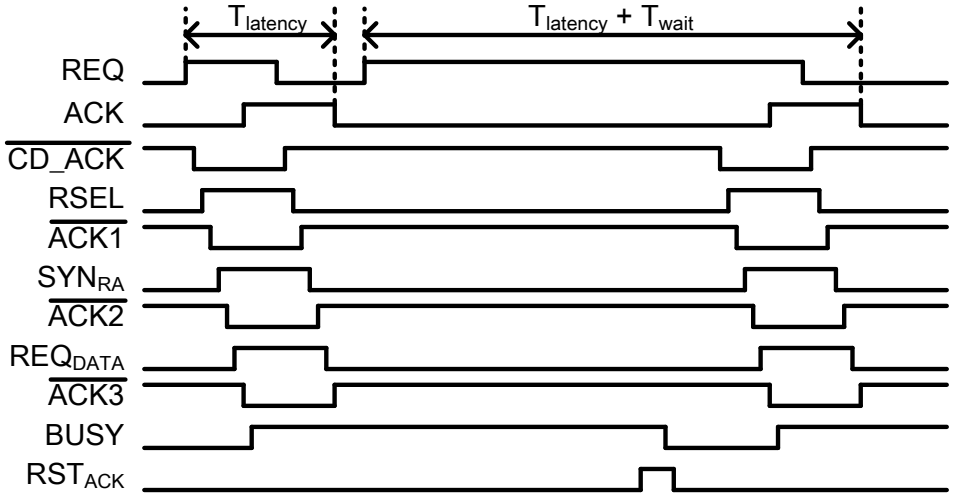


Figure 3.8: Timing diagram for AER input distribution (Fig. 3.7(a)) and pulse width and amplitude modulation (PWAM (Fig. 3.7(b)) circuits with two consecutive input events on the same row

3.2.6 Two-tier Micro Pipelining Scheme

The delivery of each address event packet is mediated by on-chip asynchronous request (REQ) and acknowledge (ACK) signals. Fig. 3.7 (a) and (b) show the asynchronous AER communication circuit and PWAM circuit enabling two-tier micro-pipelining of the input event stream to each neuron array. The first event is served directly with a latency $T_{latency}$ determined by the event handshaking between asynchronous AER circuit and the PWAM circuit described in Fig. 3.7 (a) and (b), respectively. If the first event is latched, the BUSY signal is enabled and next event is held until the BUSY signal is released by RST_{ACK}, waiting T_{wait} for completion of a synaptic pulse. The timing for two consecutive spike input events targeting neurons in the same row is shown in Fig. 3.8.

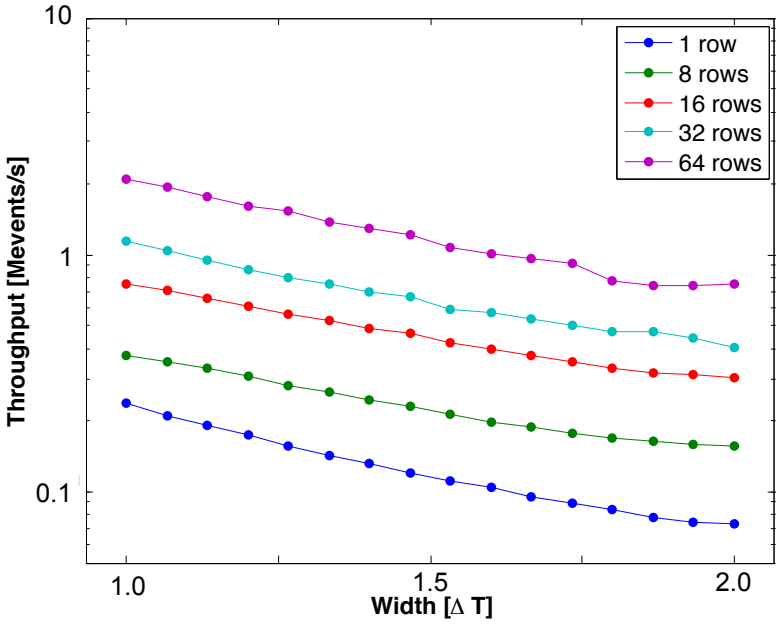


Figure 3.9: Measured throughput. Input spike streams address neurons in the same row and interleaved to multiple rows, from 8 to 64.

3.3 Measurement Results

3.3.1 Throughput

In the presented architecture, the throughput can be defined as follows:

$$Throughput = \frac{1}{\overline{T_{latency}} + \overline{T_{wait}}} \tag{3.8}$$

where $\overline{T_{latency}}$ is the average event handshaking latency and $\overline{T_{wait}}$ is the average waiting time in the case an incoming event addresses a neuron of the same row with the previous event as noted in Fig. 3.8. T_{wait} is proportional to $\Delta t / N_{interleave}$ where Δt is the input pulse width, and $N_{interleave}$ is the number of interleaving rows.

The measurement of the throughout is shown in Fig. 3.9. A spike input stream with maximum width and addressing the 32 neurons of a common row results in a 70.6 kevents/s throughput. As shown in Fig. 3.9, interleaving input spikes to multiple rows results in a higher throughput, as predicted by Eq. 3.8. When we interleave all

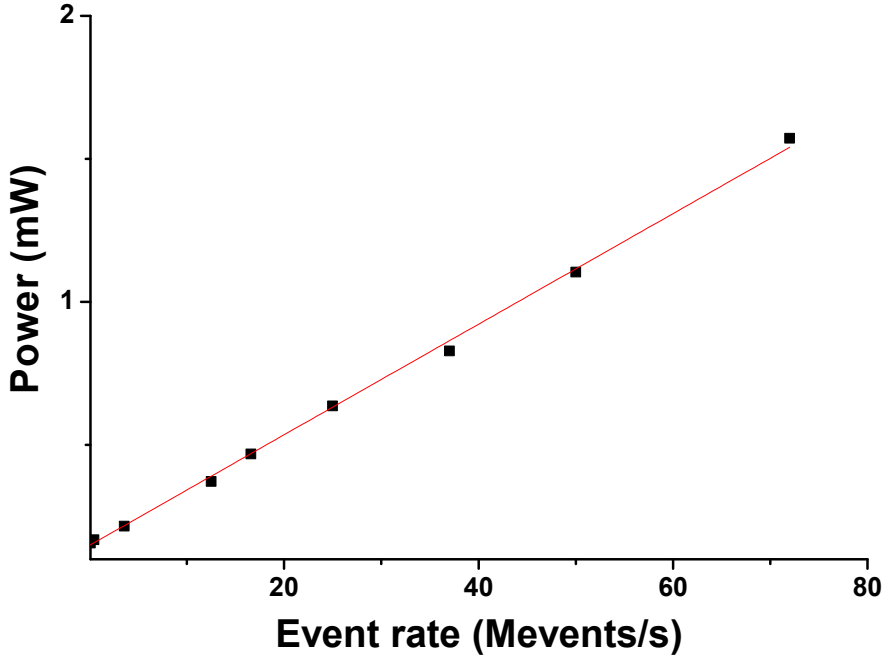


Figure 3.10: Measured input activity-dependent power consumption.

spike inputs to multiple rows the waiting time at the row pulse width modulation circuit is avoided, and we measure 18.2 Mevent/s per quadrant. The throughput for all four quadrants is thus 73 Mevents/s.

3.3.2 System-level Energy Efficiency

Each neuron in the brain projects to an average of 10,000 neurons. For this reason, the power consumption of biologically inspired neural network models is primarily governed by synaptic spike input. Fig. 3.10 shows power consumption as a function of synaptic input event rate. Power consumption increases linearly with the spike input event rate. We measured power consumption until the event rate reached the maximum throughput capability. At the maximum throughput of 73 Mevents/s, we measured a current draw of 1.31 mA from the 1.2 V analog supply, resulting in a total power dissipation of 1.572 mW, or an overall energy efficiency of 22 pJ/spike.

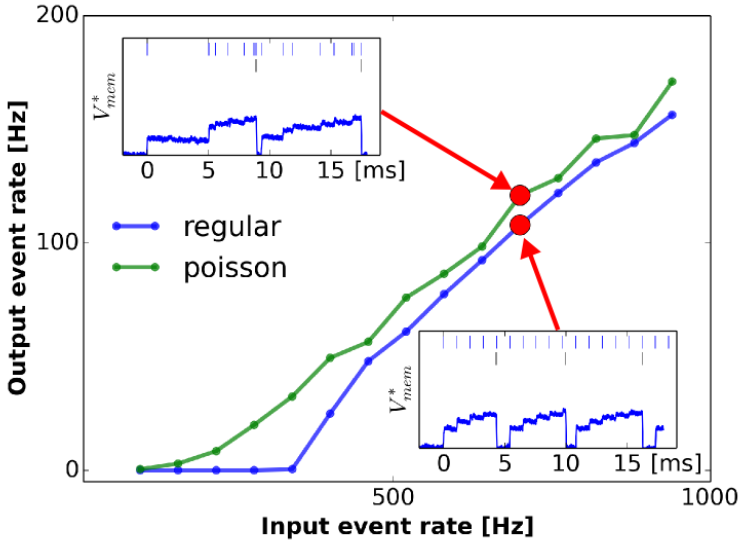


Figure 3.11: Neural activation function measured with inputs consisting of Poisson spike trains and regular spike trains. Insets: representative trace of the membrane potential for Poisson input (top left) and regular input (bottom right). Measured log-domain membrane potentials are shown exponentiated, indicated by the asterisk symbol. Input and output spikes are indicated by the top and middle rows of bars, respectively.

3.3.3 Neural Activation

We measured neural activation defined as the output event rate versus input event rate, using Poisson spike trains and regular spike trains. Fig. 3.11 shows the neural activation of one representative neuron. The shape of this activation function is threshold-linear and consistent with that of a leaky integrate and fire neuron model, where the threshold is caused by the leak. In the case of Poisson spike trains, the fluctuations in the input tend to smooth the activation function, as expected from studies of noisy integrate and fire neuron models [26].

3.3.4 Frequency Response

We measured input spike rate dependent response varying input spike rate from 500 Hz to 10,000 Hz. The interspike intervals of the input spike trains were generated using a Poisson process of constant mean rate. Fig. 3.12 shows the frequency response of one representative neuron varying input frequency (Fig. 3.12 (a)), and the gain defined

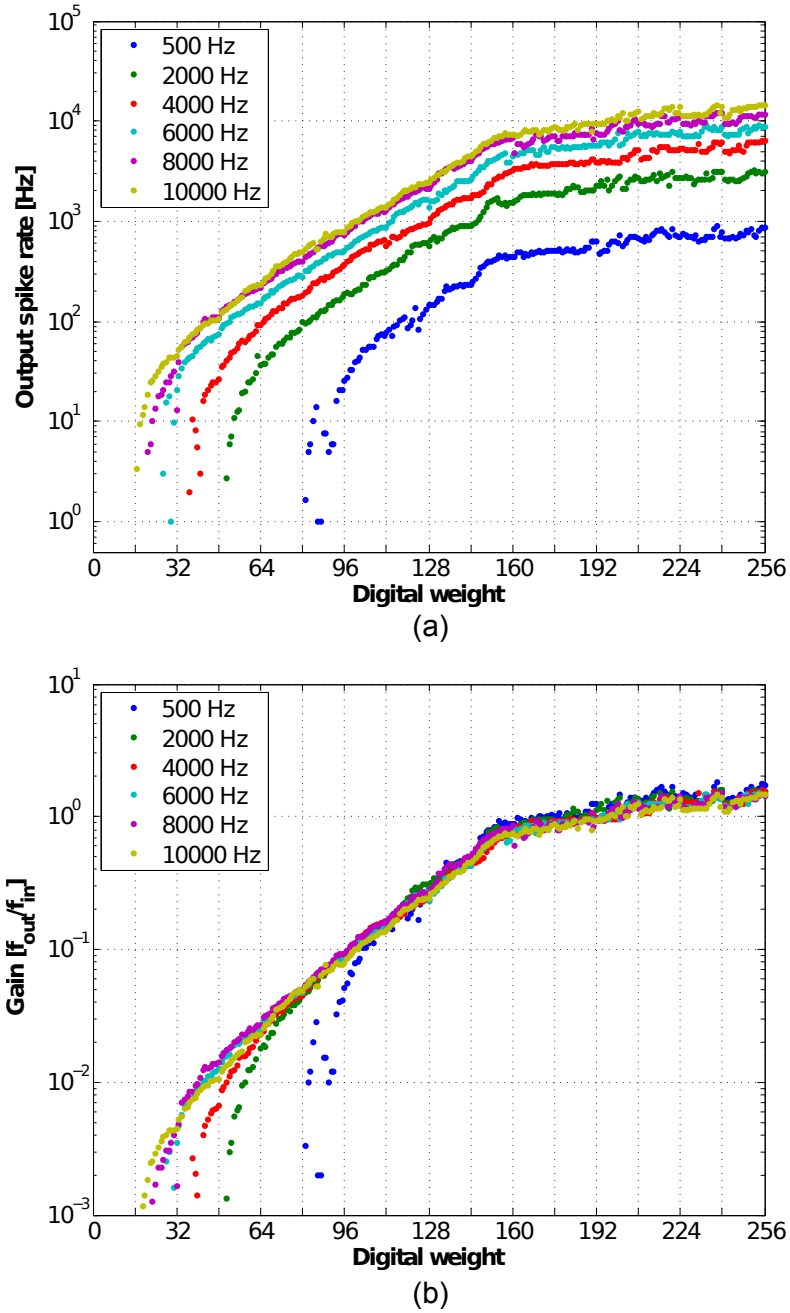


Figure 3.12: (a) Frequency responses measured with Poisson spike input trains of rates from 500 Hz to 10,000 Hz. (b) Measured gain defined as output spike rate over input spike rate.

as output spike rate over input spike rate (Fig. 3.12 (b)). Synapse time constant is shorter than membrane time constant so that input spike event integrates membrane potential. The neural output event rate saturates at high synaptic input strength because each input spike produces an output spike.

3.3.5 Neuron Response Variability

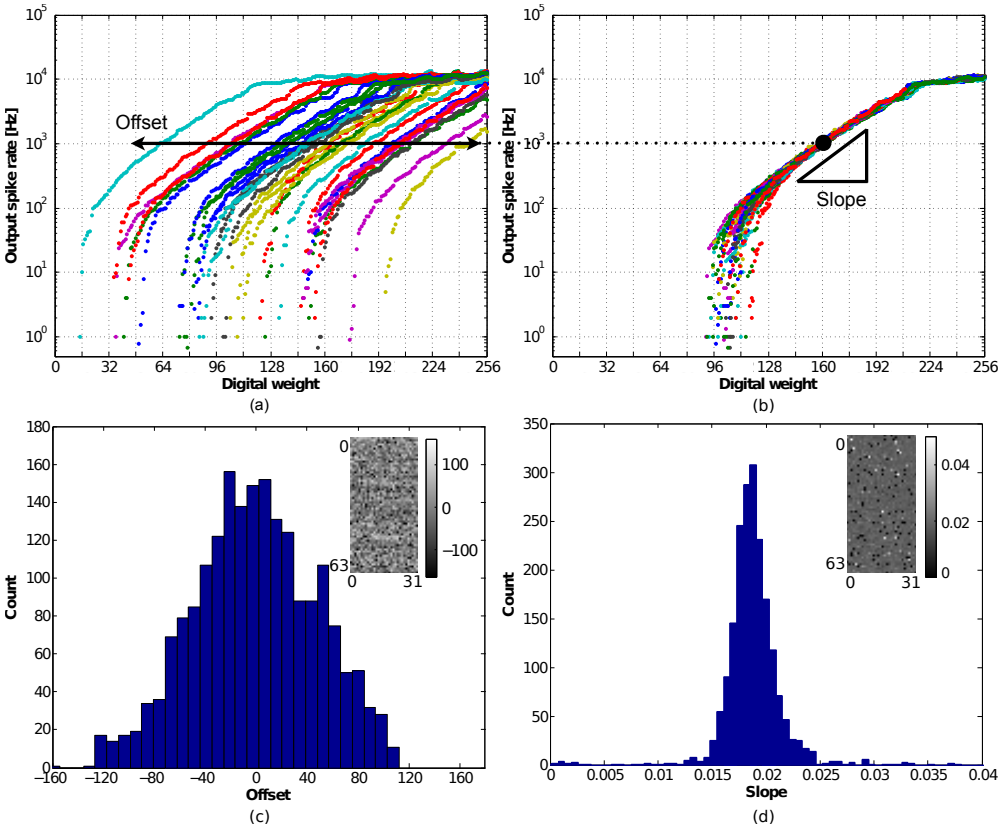


Figure 3.13: (a) Output spike rates as a function of 8-bit synaptic digital weight, which were measured from 32 neurons in one row with 10,000 Hz mean rate Poisson spike train in one second measurement. (b) Aligned neuron responses to mean of offset where output frequency gain is 0.1, while slope is defined as output spike rate increment in a decade per unit of digital weight. (c) Histogram of offsets measured from 2,048 neurons in an array. (d) Histogram of slope measured from 2,048 neurons in an array. Slope is distributed in mean of 0.0185 and standard deviation of 0.0068.

Due to transistor mismatches in transistor subthreshold operation regime, we expect to have variabilities of neuron response across a chip. Offset of neuron spike

is a major variability in neuron response caused by threshold transistor mismatch in an axon hillock circuit. While offset can be compensated by learning in address event domain [81], variation of response slope linearity is consideration.

Fig. 3.13 shows measured offset and slope of multiple neurons in a row and an array. Fig. 3.13 (a) shows output frequency responses measured from 32 neurons varying digital weight from 0 to 255 with input spike event trains of 10,000 Hz where interspike intervals were distributed in Poisson distribution. After shift lines to mean of offset, it is aligned to a mean of response shown in Fig. 3.13 (b). We defined a slope as a output spike rate increment decade per unit digital weight where neuron is in log-domain linear response regime. Fig. 3.13 (c) and (d) show histogram of offset and slope measured from an 2,048 neuron response in an array.

3.3.6 Linear Synapse Response Model

Current injection to the leaky integrate-and-fire neuron model is formulated as follows:

$$\begin{aligned} I_{inj} = C_{mem}\frac{dV_{mem}}{dt} &= g_{ext}(E_{ext} - V_{mem}) \\ &+ g_{inh}(E_{inh} - V_{mem}) \\ &+ g_{leak}(E_L - V_{mem}) \end{aligned} \tag{3.9}$$

where C_{mem} is membrane capacitance, V_{men} is membrane voltage, g_{ext} and g_{inh} are conductances of excitatory and inhibitory synapse, E_{ext} and E_{inh} are reversal potentials of excitatory and inhibitory synapse, g_{leak} is leak conductance, E_L is leak voltage, and V_{mem} is membrane voltage. To obtain simple neural response model we approximate the above terms as follows:

$$I_{inj} = g_{ext}E_{ext} + g_{inh}E_{inh} \tag{3.10}$$

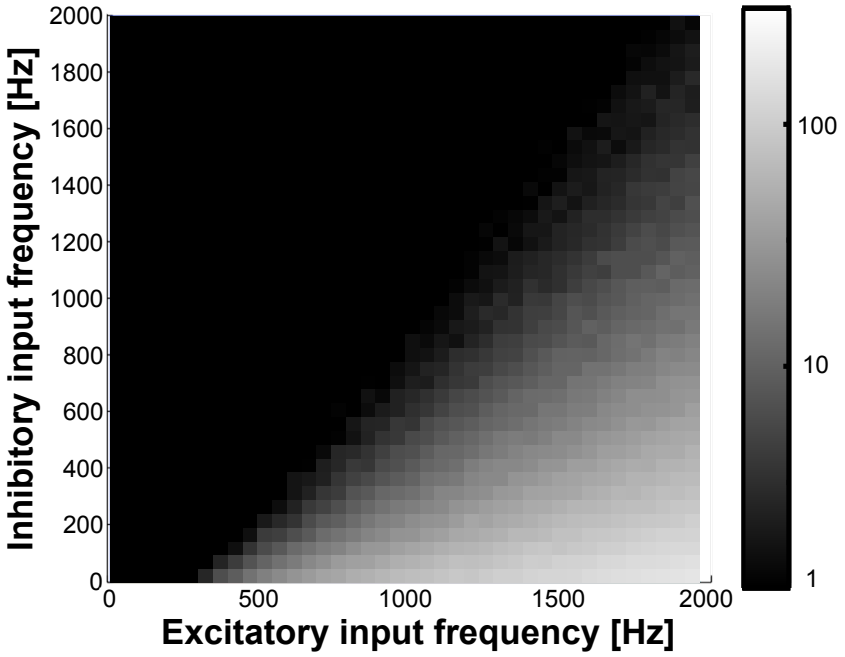


Figure 3.14: Measured output frequency varying excitatory and inhibitory input frequency from 0 to 2,000 at digital weight of 80.

As a first order approximation, we assumed that the conductance is equal to the total input spike train times a nominal synapse weight:

$$g_{syn} \propto \sum_{n} f_{in,n} w_n = f_{in,eff} w_{nom} \tag{3.11}$$

where g_{syn} is conductance of synapse, $f_{in,n}$ is frequency of nth input spike train, w_n is synapse weight of nth input spike train, $f_{in,eff}$ is sum of all input spike train frequency and w_{nom} is the nominal synapse weight of input spike train Given the first order approximation, the output frequency of the spike train is a sum of excitatory and inhibitory synaptic input spike trains with nominal weights

$$f_{out} = [f_{in,eff} w_{nom} = f_{ext,eff} w_{nom} - f_{inh,eff} w_{nom}]^{+} \tag{3.12}$$

Fig. 3.14 shows measured frequency output varying excitatory and inhibitory synapse input frequency from 0 Hz to 2,000 Hz at nominal digital weight 80. We used it as a model of neuron response for orientation tuning curve and boundary detection.

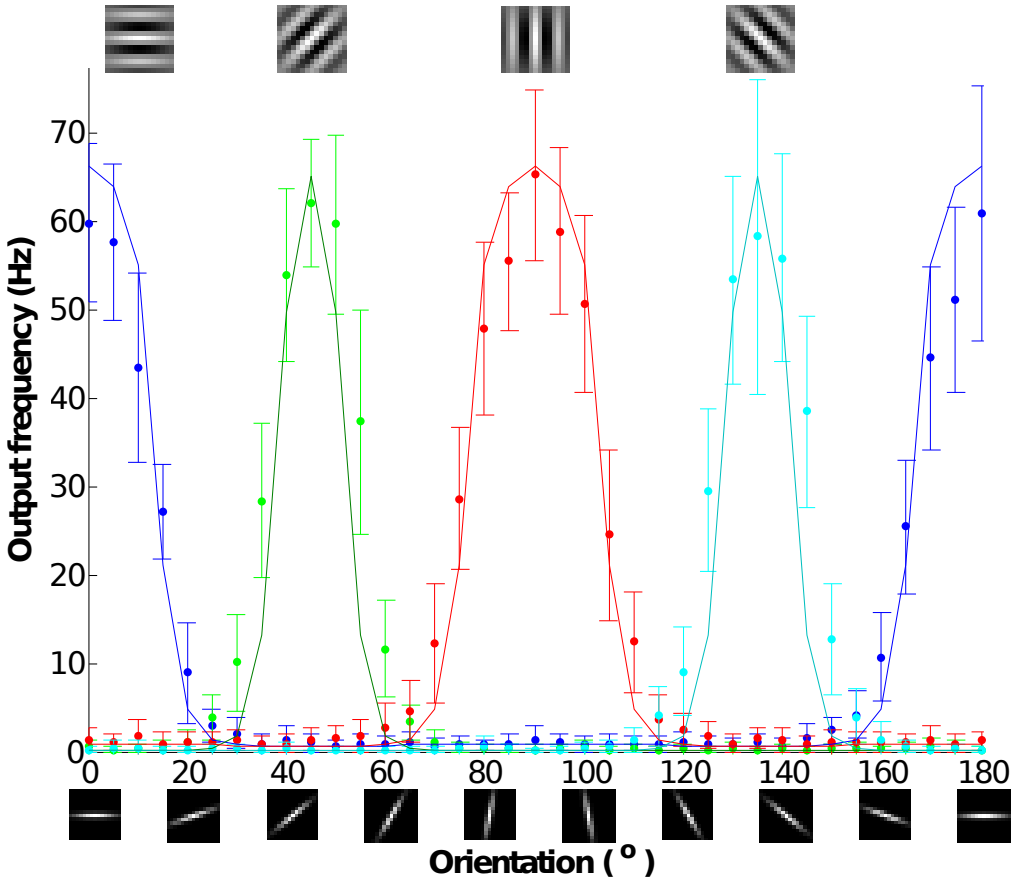


Figure 3.15: Tuning curve measurement results. The mean output frequency is plotted as a function of input bar stimulus orientation. Pixel size of bar stimuluses and orientation selective kernels are 15×15. When these are convoluted to compute tuning curve, pixel intensity of stimulus and orientation selective kernel represent synapse weight and synaptic input frequency respectively. Each data point is mean of 30 times measurement of 1 second stimulation each. Solid lines show simulation models from output frequency map shown in 3.14.

3.3.7 Orientation Tuning Curve

Orientation tuning curve is an output neural response defined as function of stimulus orientation convoluted to orientated filter. It is a typical measurement to characterize orientation selectivity in visual cortical neurons. Measured orientation tuning curves are shown in Fig. 3.15. We used 15×15 pixel bar stimulus rotating 5° from 0° to 180° and four orientations of Gabor kernels, 0°, 45°, 90°, and 135°. Stimulus pixel intensity and Gabor filter intensity are translated to input synaptic spike rate and strength respectively. Output spike rate is defined as follows:

$$f_{out} = [\sum_{i=1}^{15}\sum_{i=1}^{15} f_{in_{i,j}} w_{i,j}]^{+} \tag{3.13}$$

where i and j are index of pixel position, f_{out} is an output spike rate, f_{in} is a input synaptic spike rate and w is input synapse weight.

Each data point, shown in Fig. 3.15, is mean of 30 measurements and error bar represents one standard deviation. Each measurement runs one second input spike train. Solid lines show simulation model from output frequency mapping shown in Fig. 3.14 with first order approximation of leaky integrate-and-fire neuron model.

3.3.8 Boundary Detection

We show image boundary detection with a input image size of 113×75 pixels shown in Fig 3.16 (a) and 15×15 pixel kernels shown in Fig 3.16 (b) at first column. Same with orientation tuning curve measurement, stimulus and kernel pixel intensity are translated to input spike rate and input synapse strength respectively. Second column of Fig 3.16 (b) shows simulation models from the output frequency response map shown in Fig. 3.14. Measured results from IFAT are shown in Fig 3.16 at third column.

The neural engineering framework (NEF) is an increasingly popular tool for building generic dynamical systems using spiking neural networks [19]. Recently, the NEF demonstrated the performance of cognitively useful tasks using large-scale simulations of neurons It is of particular interest for analog VLSI implementations of neurons because it exploits the variability inherent to the circuits. The basic capability of the

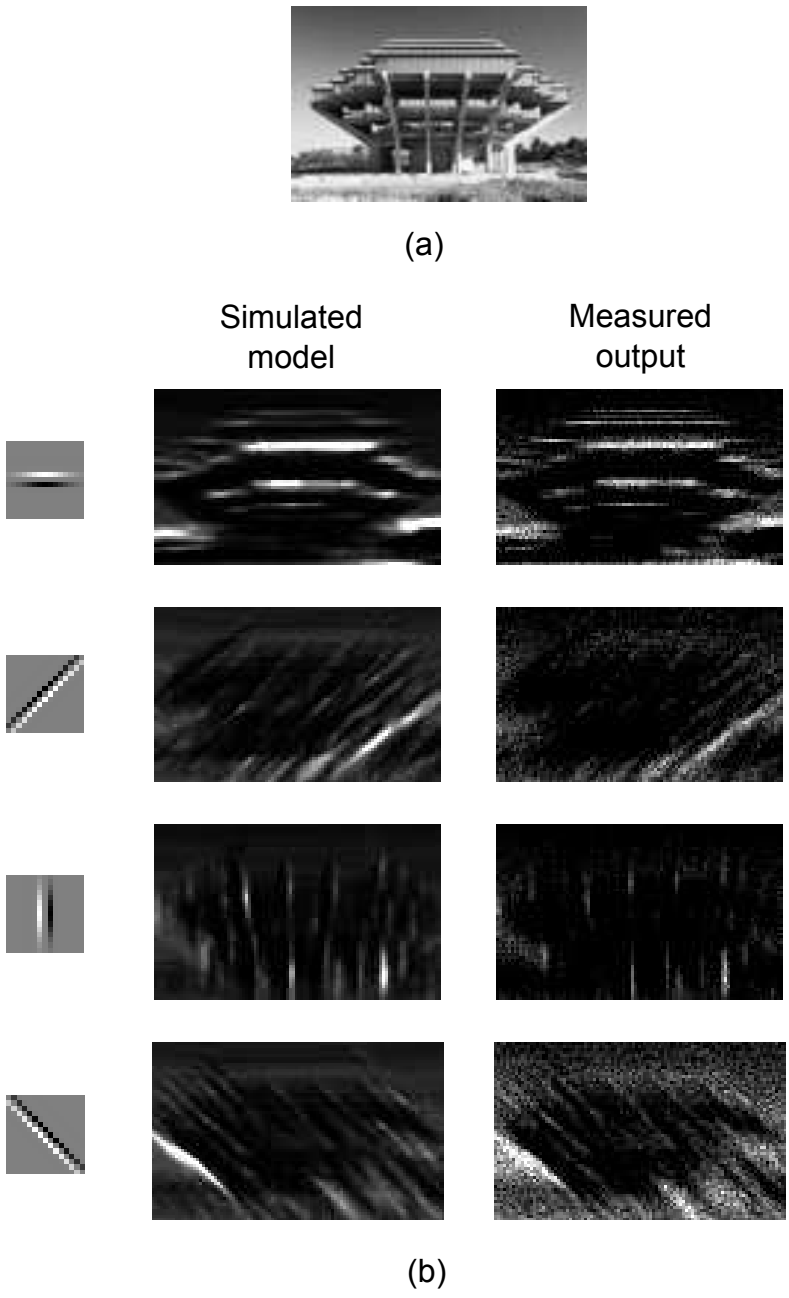


Figure 3.16: (a) 113×75 pixels input image (b) Image boundary detection results with simulated model from first order approximation of leaky integrated-and-fire neuron model and chip measurement outputs. 15×15 pixel patch convoluted with each degree of boundary detection kernels shown at first column.

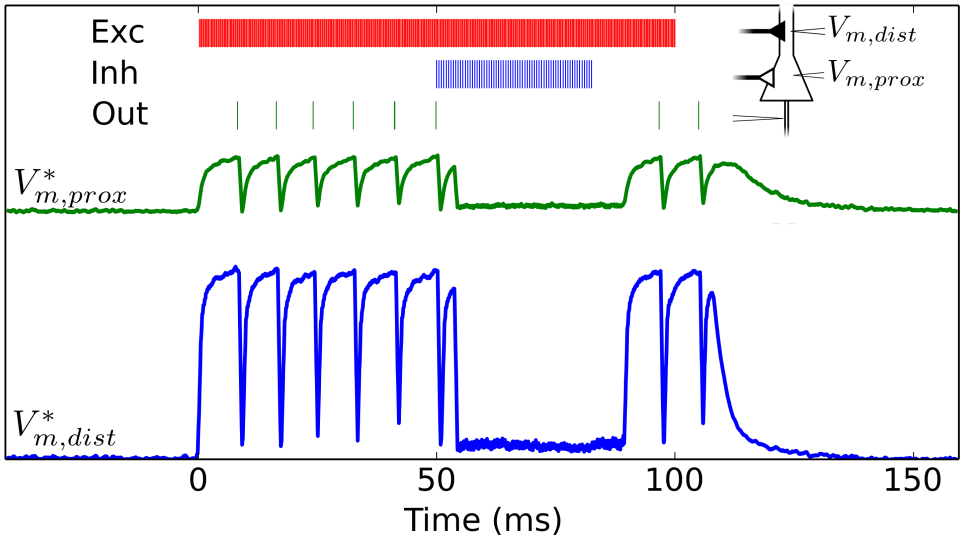


Figure 3.17: Measured interactions between the two compartments of the neuron. The distal compartment of the neuron is strongly excited, resulting in an excitatory input in the proximal compartment and the firing of the neuron. At 50ms, the proximal compartment is inhibited. This shunting inhibition blocks the effect of the upstream excitation.

NEF is to generate a spiking neural network obtained by linearly combining the firing rate transfer curves optimally, according to the desired encoding function. This combination usually requires configuring synaptic weights with high precision. Previous neuromorphic implementation have made use of synaptic stochasticity to overcome the requirement of high precision synaptic weights [15]. The current IFAT features a wide dynamical range and precision of the synaptic weights, making it possible to realize such networks without using synaptic stochasticity. Indeed, some of the steps used in the configuration of the edge detection neurons presented in this section are functionally equivalent to the ones used internally in the NEF.

3.3.9 Shunting Inhibition

The implemented two-compartmental neuron model is a distinguishing feature of this circuit compared to most existing implementations of integrate and fire neurons. The conductance between the compartments is configurable, allowing the distal and the proximal compartments of the neuron to interact more or less strongly. Fig. 3.17 il-

lustrates such an interaction in an example of shunting inhibition, where the excitatory synaptic current from the distal compartment is shunted by inhibition at reversal potential near rest in the proximal compartment.

3.4 Conclusions

We present a fully asynchronous 64k integrate-and-fire neuron array transceiver with 22-pJ/event energy efficiency and a two-tier pipeline circuit enabling 73-Mevents/s throughput.

Table 3.1 summarizes measured characteristics of the IFAT chip in relation to state of the art. Biophysical detail in compartmental conductance-based dynamics is afforded without compromise in area density and energy efficiency. Sustaining high efficiency in system-level interfacing of the IFAT chip for large-scale neuromorphic computing calls for future work in vertical integration of hierarchical address-event routing (HiAER) and synaptic routing tables (SRT) [64] using hybrid CMOS-memory technologies.

Chapter 3 is largely a reprint of material that was accepted to 2014 Biomedical Circuits and Systems Conference : J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, “A 65k-Neuron 73-Mevents/s 22-pJ/event Asynchronous Micro-Pipelined Integrate-and-Fire Array Transceiver”, IEEE Biomedical Circuits and Systems Conference (BioCAS 2014), Oct 2014. The author is the primary author and investigator of this paper.

Table 3.1: Related and Prior Works

Reference	[79]	[53]	[55]	[70]	[80]	[85]	[4]	This work
Technology (nm)	130	45 (SOI)	28	180	500	130	180	90
Die size (mm^2)	102	6	430	50	9	25	168	16
Neuron type	Digital	Digital	Digital	Analog	Analog	Analog	Analog	Analog
Total number of neurons	~5216[1]	256	1M	512	2k	65k	65k	65k
Neuron area (μm^2)	N/A[1]	3325	14.3[4]	1500	240	200	1800	140
Throughput (Mevents/s)	5	N/A[3]	N/A[3]	65	1	35	91	73
Energy (J/Spike)	8n	45p[2]	26p	N/A	645p	55p	31.2p	22p

[1] Software-instantiated leaky integrated and fire neuron
[2] Excluding static power
[3] Internal connectivity
[4] By multiplexing the neuron 256 times

Chapter 4

Hierarchical Address-Event Routing Architecture for Reconfigurable Large Scale Neuromorphic Systems

4.1 Introduction

This chapter focuses on hierarchical address-event routing (HiAER) as a multi-scale tree-based extension on AER synaptic routing for dynamically reconfigurable long-range synaptic connectivity in neuromorphic computing systems. The HiAER synaptic event routing infrastructure serves as a communication backbone to integrate-and-fire array transceivers (IFAT) [27, 80, 84] and other event-driven spiking neural network hardware systems, *e.g.,* [31, 47, 72, 78], the details of which are beyond the scope of the present chapter. Using results from queueing theory we previously showed that HiAER offers scalable synaptic event throughput, independent of neural network size, for given synaptic fan-out and nominal axonal delay, and without restriction on spatial range of synaptic connectivity [36]. Another distinguishing feature of HiAER is that synaptic connections code not only programmable synaptic strength (probability of presynaptic release and postsynaptic conductance) but also programmable axonal delay, implemented in the timing of events routed from source to destination. In Section 4.2, we describe fundamentals of HiAER and its edge-vertex-dual like mapping of flat arbi-

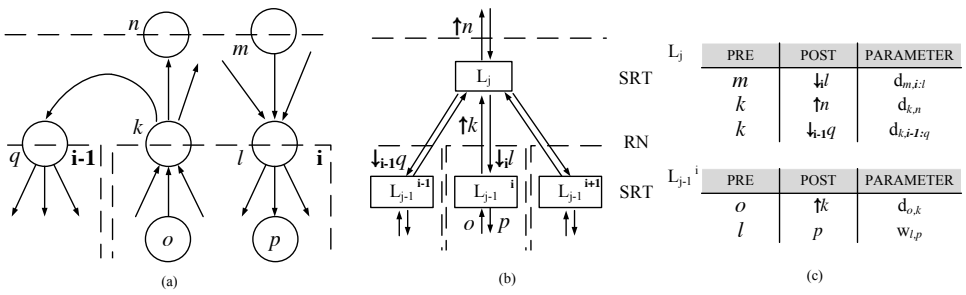


Figure 4.1: (a) Hierarchical neural network with ascending and descending neural projections. Physical neurons are represented by o and p, and inserted *relay neurons* (RN) interfacing across hierarchical partitions are denoted by q,k,l,n,m. Italic indices j and $j-1$ represent levels in the hierarchy, while boldface indices $\mathbf{i}$ and $\mathbf{i}-\mathbf{1}$ represent individual blocks within one level in the hierarchy. (b) The edge-vertex-dual of the hierarchical routing network. (c) Example of entries within the Synaptic Routing Table (SRT) shown in (b).

trary network topology onto hierarchically partitioned multi-scale local networks, with relay neurons interfacing between consecutive scales in the hierarchy. In Section 4.3, we describe implementation of the HiAER inter-bus routing node including memory interfaces to local routing tables and priority queue for timed event registration and delivery. Section 4.4 validates scaling properties of HiAER nominal throughput and latency in a field-programmable gate array (FPGA) based experimental platform on a custom printed circuit board (PCB) realizing two levels of HiAER each with branching factor of four. A nearly four-fold improvement in combined throughput and latency are demonstrated for HiAER across four routing nodes, in comparison to single-node AER as previously reported in [64]. Finally, Section 4.5 concludes with a discussion on HiAER advantages, limitations, and extensions.

4.2 Hierarchical Address Event Routing

4.2.1 Global Synaptic Connectivity and Axonal Spike Transmission

The efficient and scalable emulation of biological networks with VLSI learning systems requires abstraction of various biological details to ease the implementation and analysis of such networks. In biological neural networks neural spikes, which are elec-

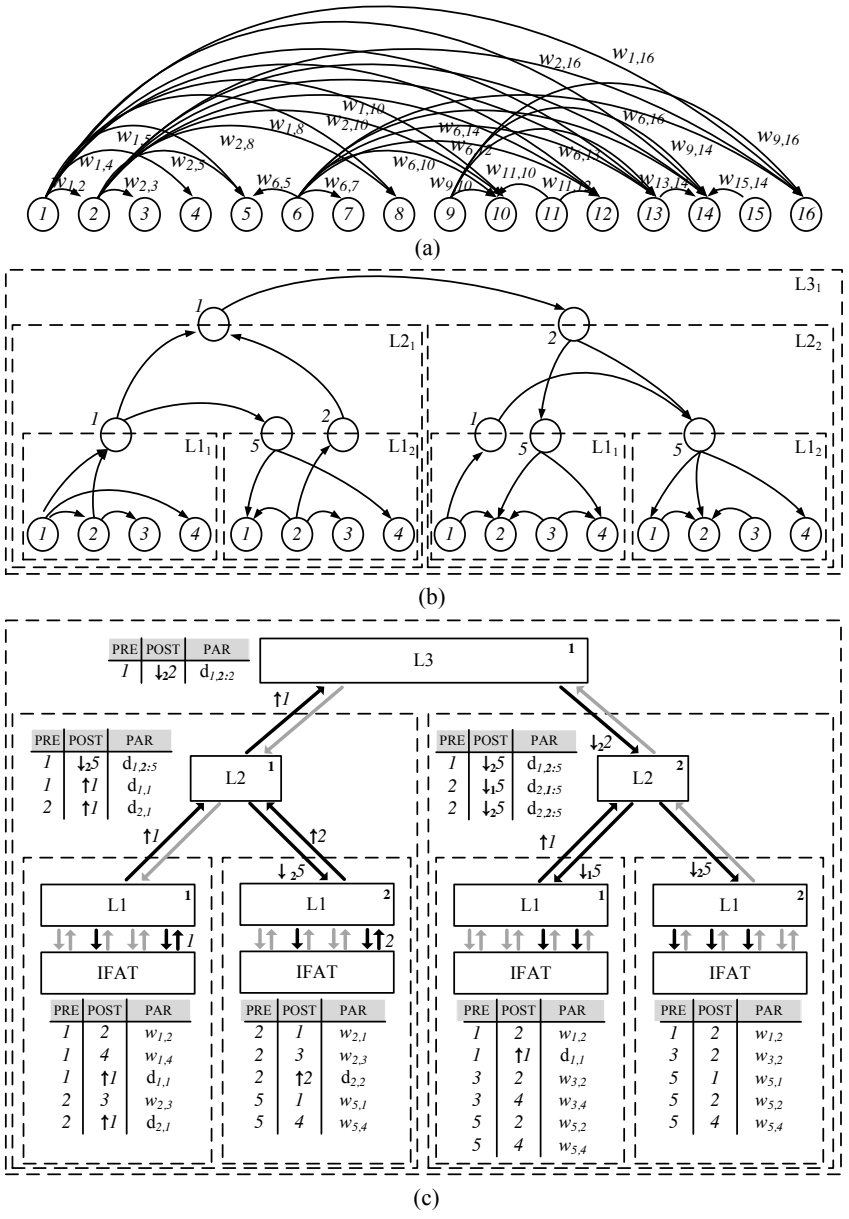


Figure 4.2: (a) Example network with 16 neurons and weighted synaptic connections. (b) Example partitioning into hierarchical neural network with ascending and descending projections through inserted relay neurons. (c) Corresponding edge-vertex-dual Hi-AER implementation with synaptic routing tables (SRT) at each level in the hierarchy.

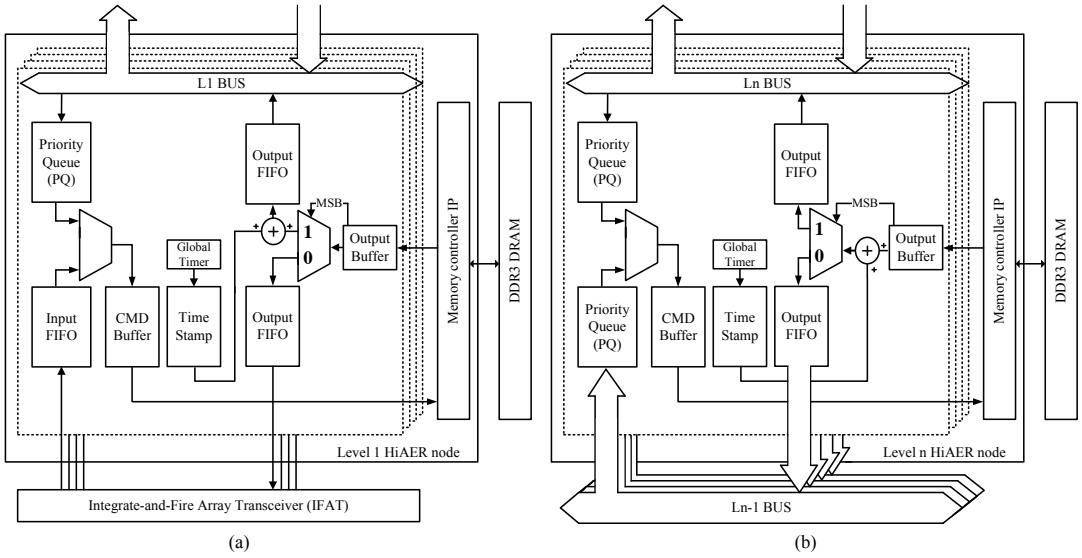


Figure 4.3: (a) Simplified system architecture of a HiAER node at Level 1 (leaf in the hierarchy), routing synaptic events through the Synaptic Routing Table (SRT) between physical neurons in the local Integrate-and-Fire Array Transceiver (IFAT) and relay neurons on the $L1$ bus. The SRT maps incoming events from any neuron onto outgoing events either to the final synaptic destination on the IFAT (along with synaptic strength w), or up the hierarchy through the $L1$ bus (along with timing information for axonal delay d). (b) Digital system architecture of a HiAER node at Level $n > 1$ (higher in the hierarchy), largely identical to Level 1 except for substitution of the IFAT with a $Ln-1$ bus, and of the $L1$ bus with a Ln bus. In the absence of physical neurons, events are transmitted only between relay neurons higher and/or lower in the hierarchy (along with timing information for axonal delay d).

trical pulses, originate in the axon hillock of a neuron, and propagate through the axon of the neuron via synapses of varying coupling strengths to one or several dendrites of receiving neurons, and within the receiving neurons to their cell bodies (soma), where a sufficiently high total amount of excitation by spikes gives rise to another spike. In the case of VLSI implementation where artificial neural arrays emulate such spiking neurons, we can uniquely identify each neuron in such an array by some address A, *e.g.,* its (x,y) coordinates within that array. Similarly, a combination of various synaptic properties can be grouped together and represented by the connection strength (such as postsynaptic conductance, or presynaptic release probability) w, and axonal transmission delay d for that synapse. Thus, the triplet (A,w,d) encodes each spike event electronically as a digital event.

4.2.2 Hierarchical Neural Network Topology

Although AER is capable of interconnecting neurons in a reconfigurable manner [27, 31, 47, 72, 78, 80, 84], the limited bandwidth of single-bus AER restricts the network size to thousands of neurons. Grid-based [39, 56, 62, 87] and tree-based [4, 21, 36, 54, 58, 71] extensions to AER have aimed at extending the bandwidth and spatial range of synaptic connectivity across multi-chip neural arrays in a scalable and efficient manner. Here we focus on hierarchical address-event routing (HiAER) as a multi-scale tree-based extension on single-bus AER to offer scalable synaptic event throughput without restriction on spatial range of synaptic connectivity [36]. Neurons communicate synaptic events, within and across neural arrays, over dedicated serial communication links. Depending on the destination, a spike address event may pass through several routing nodes, arranged in a tree like configuration, on its way from the presynaptic neuron to the postsynaptic neuron. At each routing node, an incoming address event may trigger multiple outgoing address events, enabling bundling of events to spatially co-located neurons. We term these entries within the synaptic routing table (SRT) relay neurons (RNs), thus connections from the neural array ascending between levels of hierarchy can be considered to be projections between relay neurons or projections between a neuron and relay neurons. Similarly, in descending a hierarchy these entries can be considered projections between relay neurons or between a relay neuron and the neural

array. The projections from incoming to multiple outgoing address events are stored in the SRT. The total transmission delay d is partitioned across various routing nodes in a hierarchical manner such that nodes higher up in the tree implement longer delays.

Fig. 4.1 defines key concepts and notations in this chapter. An example segment of a partitioned spiking neural network with directed synaptic connections between physical and relay neuron nodes in Fig. 4.1 (a) is transformed into an equivalent representation in Fig. 4.1 (b), with the corresponding SRT entries shown in Fig. 4.1 (c). The neural network segment in Fig. 4.1 (a) also indicates the hierarchy of connections across levels in the partitioning, *i.e.,* either between relay neurons on the same level (k to q), from a lower to a higher level (o to k, and k to n), or going from a higher to a lower level (m to l, and l to p). Within the context of our architecture in Fig. 4.1 (b), we collate all connections that belong to a neuron into SRT entries, thus creating a single entity to represent all its synapses. There is a SRT entry for each unique neuron or relay neuron within the hierarchy, identifying what is communicated by each event. Thus each link represents a unique neuron or relay neuron, while all synaptic connectivity information resides within SRT nodes. Topologically this transform is akin to the edge-to-vertex dual of a graph, where edges transform to vertices and vice versa. The SRT entries of the dual transformed network are shown in Fig. 4.1 (c). Note that only connections presynaptic to the neural array specify weight information, while those entries presynaptic to relay neurons specify axonal delay information. SRT entries also specify directional information shown by $\downarrow$ for entries descending the hierarchy and $\uparrow$ for those ascending. The same neuron can have multiple addresses, however it is unique within the its scope at any level of the hierarchy. This partitioning and grouping of messages ensures more efficient use of memory and bandwidth.

The above examples and description are extended to any level of hierarchy. Fig. 4.2 shows an extended example of conversion from a 16-neuron network (Fig.4.2(a)) through a partitioned network with three levels of hierarchy (Fig. 4.2 (b)) to its dual representation of hierarchical routing with SRT entries (Fig. 4.2 (c)). Arrows between routing nodes in Fig. 4.2 (c) represent the direction of AER communication, where solid arrows show active links communicating neuron spike events through each level of the HiAER hierarchy.

4.2.3 Distributed Axonal Delay

Axonal delay in action potential propagation, such as along neuronal fiber bundles in the white matter of cortex, plays an integral role in the functioning of the central nervous system [16], and has been the basis for models of neural computation based on coincidence in delay-based matched filtering of spike events [34]. A distinguishing feature of HiAER is that it explicitly accounts for relative timing in event transmission and delivery, providing a programmable axonal delay d for each individual synaptic destination A. Such explicit delay in the AER path provides a compact event-based digital alternative to previously proposed means to implement axonal delay in neuromorphic hardware, *e.g.,* [3, 75, 82].

Axonal delays depending on a variety of biological factors may range between tens of μs to hundreds of ms [16, 76]. In order to cover such wide range of time scales, an architecture with high temporal dynamic range is required. HiAER approaches this problem by partitioning delays in hierarchical fashion, in tandem with the partitioning of the network. Implemented axonal delays are distributed across HiAER routing nodes for each of the the relay neurons in the hierarchy. The net axonal delay d is thus the sum of incremental delays for all relay neurons in the path from presynaptic source to postsynaptic destination. Incremental delays are implemented by incrementing the deliver-by time-stamp of outgoing events, and by priority-queuing incoming events, not releasing them until the deliver-by time is reached, as elaborated in Section 4.3. One challenge with the implementation of delayed event queuing is that total queue occupancy grows linearly not only in event rate, but also in average delay, according to Little's Law [46]. Hierarchical partitioning of delays in HiAER allows to optimize for minimum overall queue occupancy by assigning largest incremental delays to relay neurons at highest levels, and proceeding with remaining incremental delay assignments down the hierarchy in greedy fashion, leaving smallest incremental delays at the lowest level (HiAER Level 1) where events fan out in greatest numbers to the local IFAT [36]. Such partitioning of axonal delay is consistent with the qualitative observation that longer axonal fiber bundles that interconnect more distant brain regions carry greater delays [76].

4.3 Hardware Implementation of HiAER

4.3.1 Routing Node System Architecture

The HiAER router, as shown in Fig. 4.3, arbitrates between input events, time-stamps a selected event, then accesses its entry in the synaptic routing table and places the entry on the bus *en route* to its destination. Leaf nodes in the tree, at Level 1, route local spike events to and from the IFAT as shown in Fig. 4.3 (a). Two event input paths feed into the Level 1 HiAER node: up from the local IFAT, and down from the $L1$ bus. Events are encoded differently depending on their source: events originating from IFAT contain the address of the neuron that spiked, whereas events from the $L1$ bus carry a deliver-by time-stamp and are kept in the priority queue until that time is reached. A more detailed explanation of the priority queue is provided in section 4.3.3. Time stamping is performed upon arrival of the arbitrated input event, loading the instantaneous global timer value onto the time stamp register. A local register copy of the the global timer, synchronized across all HiAER nodes, is globally reset and periodically incremented (every 1 ms). The event enters the SRT in DDR3 DRAM (third-generation double data rate dynamic random-access memory) through the CMD buffer and memory controller, returning a sequence of output events through the output buffer. The most significant bit (MSB) of the output event determines whether it is routed upward to the $L1$ bus, or downward to the IFAT. The content of the SRT and format of output events are described in Section 4.3.2.

Event routing at higher levels in the hierarchy proceeds in similar fashion, as illustrated in Fig. 4.3 (b) for the Level n HiAER node routing between $Ln-1$ and Ln buses. Differences with Level 1 routing arise due to need for a priority queue at both input paths from the higher (Ln) and lower level ($Ln-1$) of the hierarchy, enabling fine multi-level distributed control over the axonal delay parameter d. The format for entries in and events through the SRT is also different, as elaborated next.

4.3.2 Synaptic Routing Table

Synaptic routing tables (SRTs) specify all synaptic connectivity, leading a synaptic event from its source to its final destination through all levels of the hierarchy. In

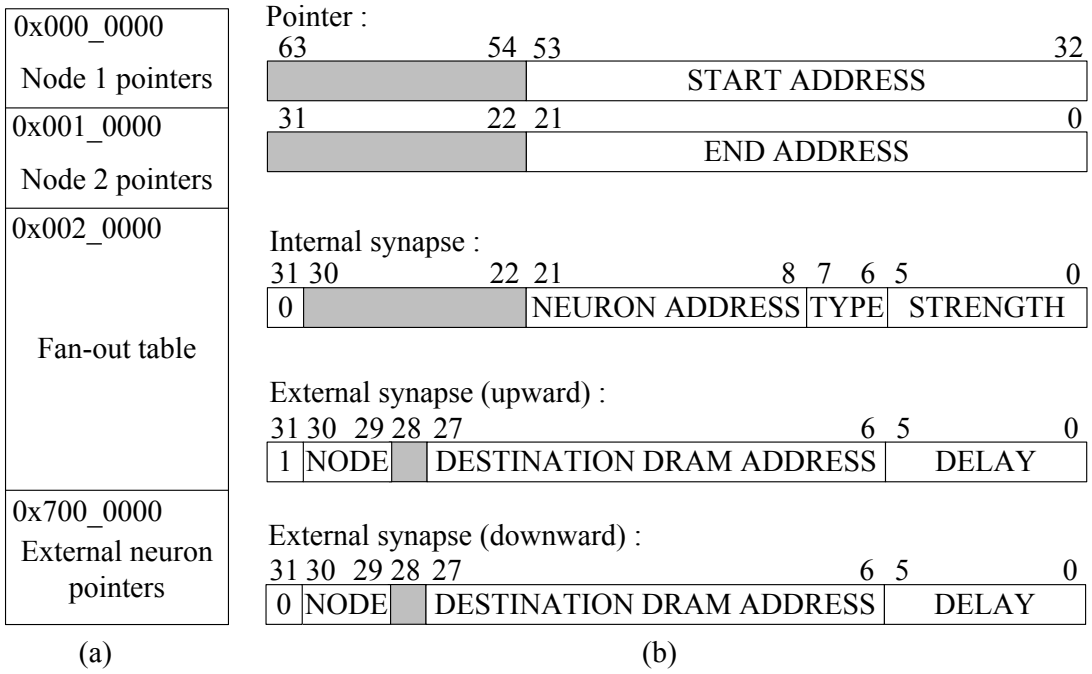


Figure 4.4: (a) Synaptic routing table (SRT) storing pointers and fan-out information in 2-Gbit DDR3 DRAM. The hexadecimal start addresses of DRAM partitions are shown. (b) SRT formats of internal and external neuron pointers, and internal and external types of synaptic fan-out events. *Top:* A 64-bit pointer contains 22-bit pointers to start and end addresses of the synapse events stored in the same DRAM. *Center:* Internal synapse event connecting to a local neuron, encoding 14-bit neuron address, 2-bit synapse type and 6-bit strength. *Bottom:* External synapse events connecting to lower ($Ln-1$) and higher (Ln) levels of HiAER nodes, with 2 bit node address, 22 bit address of the destination (relay neuron) and 6 bit delay.

addition, SRTs code the necessary information to distribute the axonal delay d across the path, and to deliver synaptic strength w (presynaptic release probability and postsynaptic conductance) on the final path segment at Level 1 to the destination in the local IFAT.

SRTs are implemented using two 2-Gbit DDR3 DRAM (Micron MT41J128M16) per Spartan-6 XC6SLX45T Xilinx FPGA, each interfacing through a dedicated bus and independent memory controller. The memory controller further provides multi-port access for sharing each physical DRAM with multiple data paths. In the current implementation we partitioned each HiAER node into four leaf nodes, with two nodes sharing one DRAM through the same memory controller.

Fig. 4.4 shows the memory partitioning and various formats of events stored in DRAM. For each input neuron a 64-bit pointer contains start and end addresses of synaptic fan-out entries in the same DRAM. The memory controller scans the data between start and end addresses to retrieve the information specifying each of the outgoing events in sequence, where each occupies two words (32 bits) in the fan-out table. Three types of outgoing events are distinguished: events serving internal synapses local to the IFAT (for Level 1 HiAER nodes only), and events leading to external synapses up or down through the HiAER hierarchy. The event type is marked by the most significant bit (MSB) of the 32-bit event in memory, which selects the path of the outgoing event up or down the HiAER hierarchy by the multiplexer shown in Fig. 4.3.

Internal synaptic events (MSB $= 0$ downward events at HiAER Level 1 in Fig.4.3 (a)) reach their final synaptic destination in the IFAT local to the HiAER node. The internal event contains the IFAT postsynaptic neuron address and synapse type A, and postsynaptic conductance w [85]. Other pertinent synaptic parameters, such as presynaptic release probability for stochastic synapses [27], may also be included in w. However, information on axonal delay d is excluded here, other than inherent delay in propagating the event through HiAER.

External synaptic events (MSB $= 1$ upward events, or MSB $= 0$ downward events at HiAER Level $n > 1$ in Fig. 4.3 (b)) connect to synaptic neurons in another node, whether neighboring or at another level in the hierarchy. External events code explicit delay timing information contributing incrementally to overall axonal delay d of

the chain of events from source to final synaptic destination. The outgoing event feeding through the output buffer in Fig. 4.3 is given a deliver-by time-stamp constructed as the sum of the 6-bit delay and the 10-bit time-stamp of the incoming event, prior to exiting the HiAER node.

The number of synapses per neuron is not constrained in hardware, other than the total number of synapses that can be stored in available memory, not occupied by (node 1, node 2, and external) pointer blocks indicated in Figure 4.4 (a). A memory capacity of 2-Gbit for every 2 HiAER nodes is chosen to accommodate a biologically realistic average synaptic fan-out of 1,024 at 16,384 (2^{16}) neurons per node and 32 bits per synapse.

4.3.3 Priority Queue

The priority queue (PQ) serves to hold incoming events, along with their deliver-by time-stamps, and release each event only once its time-stamp is reached by the global timer value. Hence the nominal incremental delay, in units of the global timer clock, is the 6-bit delay value as added to the sourcing event 10-bit time-stamp at the preceding HiAER node (see Section 4.3.2). This quantized value is a lower bound on the incremental delay actually implemented by the PQ. The slack in the timing (tightness of this lower bound) is given by propagation delays in the routing path, mainly from the PQ exit stage through the event arbiter and CMD buffer shown in Fig. 4.3. Other propagation delays in the path from source to destination (from the previous routing node's SRT and DRAM memory controller, output buffer, output FIFO, and the inter-node bus, to the current node's PQ entry stage) are inconsequential as long as their cumulative delay is smaller than the programmed incremental delay, since any smaller net delay is absorbed in the wait time in the PQ.

The PQ uses the global timer along with an adjustable unit-time step parameter determining the granularity of the implemented delay. The implemented 1 ms time step and 6-bit resolution in incremental delay support nominal single-node axonal delays up to 63 ms, with greater axonal delays achievable, if so desired, by nested routing across the hierarchy.

Fig. 4.5 shows the block diagram and state machine of the PQ and an example

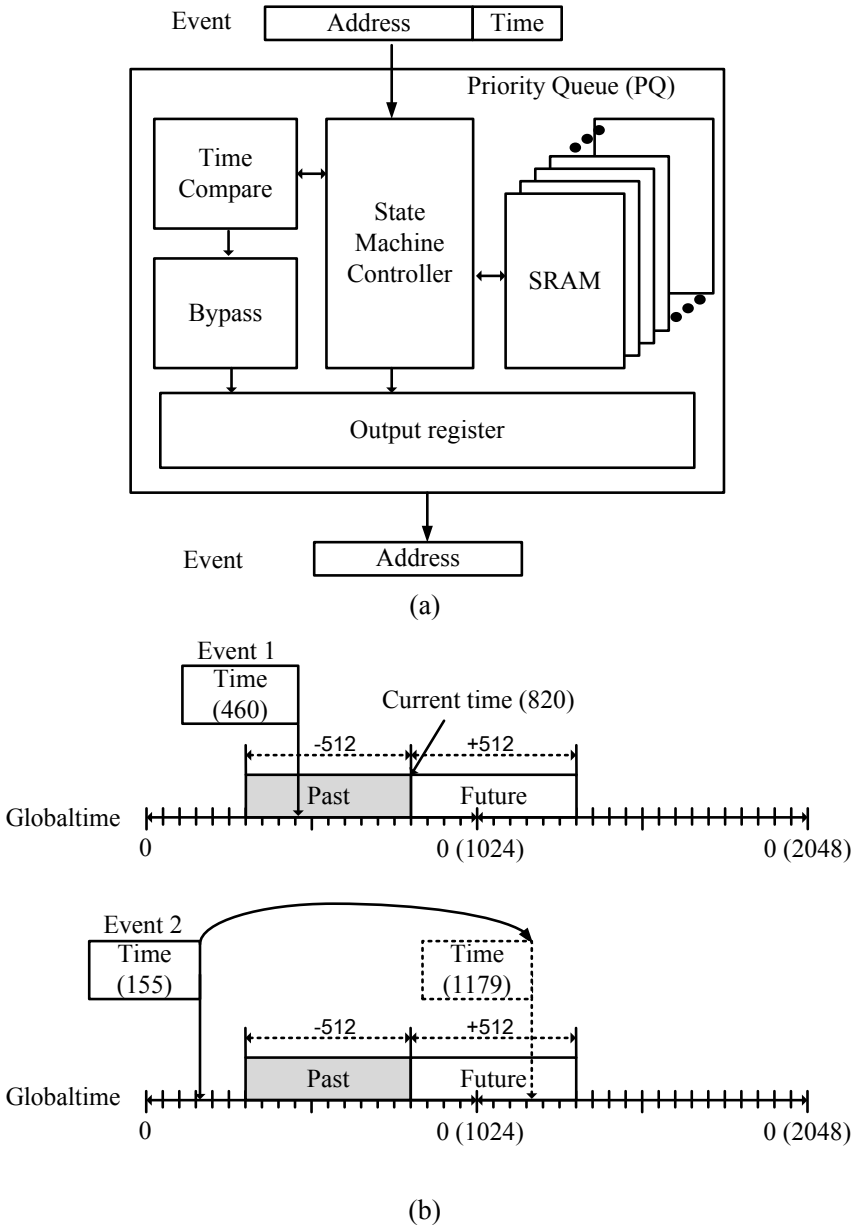


Figure 4.5: (a) System diagram implementing the priority queue (PQ). Incoming events and their deliver-by time-stamps are held in memory until their deliver-by time is reached by the current global time. (b) Examples illustrating temporal aliasing of the 10-bit event deliver-by time stamps over the horizon of the 10-bit current global time, distinguishing active future events from late past events.

illustrating the time comparison method used. The PQ consists of a time comparator, a state machine, output register and an SRAM module. The time comparator compares the current global time with deliver-by time-stamps that join incoming events. The state machine controls the PQ event flow depending on the status of incoming events and the time to the next scheduled event release in the queue. Incoming events, consisting of a 22-bit DRAM address and a 10-bit deliver-by time-stamp, are inserted (pushed) onto the queue in SRAM. Released events are removed (popped) from the queue, and reside in the output register until acknowledged by the next stage.

Due to finite bit-width of the global timer, improper time-aliasing can occur with events whose deliver-by times lie beyond half of the full digital timing window range. By convention we consider such aliased events as arriving too late, requiring immediate attention. Examples illustrating desired and improperly aliased operation are shown in Fig. 4.5 (b). The top event has a deliver-by time-stamp of 460 at a current time of 820, hence is considered as a missed past event and is expedited to the CMD buffer. In contrast, the bottom event time-stamped for 155, wrapping around to 1,179 ($= 2^{10} +$ 155), falls within $2^9 = 512$ cycles of the 820 current time, and enters the PQ wait table in the memory stack.

The finite state machine implementing the PQ, with state transitions driven by incoming events and time comparisons, is illustrated in Fig. 4.6. Incoming events (identified by EVENT_IN) trigger a time comparison, the result of which either directs the event to the output register (in case of a current or past event), or pushes it into the queue on the first available write pointer (in case of an active future event). The state machine also keeps track of the next event to be served using a NEXT_TIME variable, as the earliest of all stored time-stamps in the queue, and its read location. Whenever the global current time reaches the NEXT_TIME value, the event stored at the read pointer is popped from the queue and directed to the output register. After the pop, the PQ enters a search to update the read location and NEXT_TIME, circulating once through the queue from the current location for the earliest future time-stamp, while also popping any other event with the same deliver-by time as the present global-time. Otherwise, the state machine checks for vacant positions in the queue to fill any available among 16 write pointers.

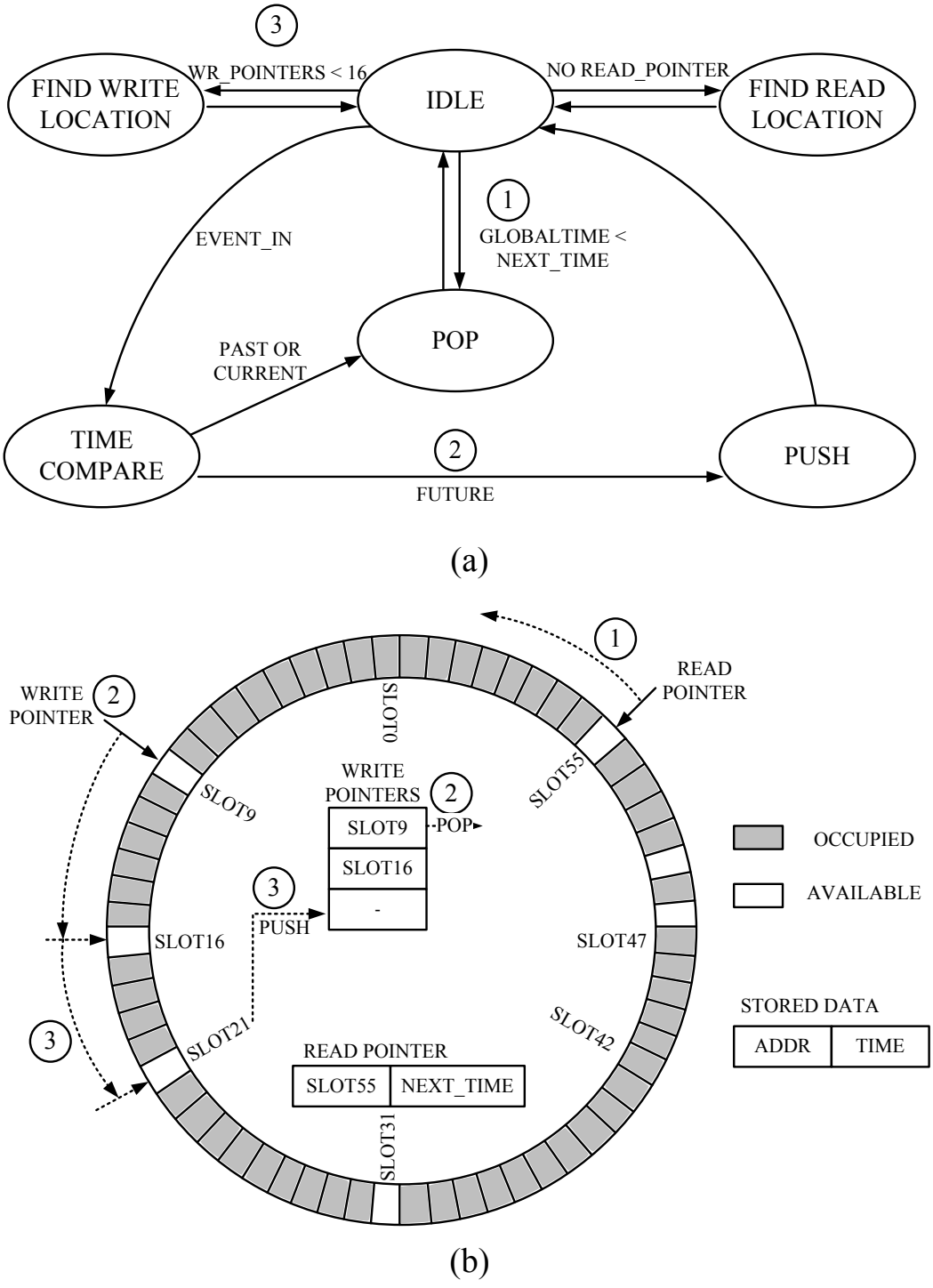


Figure 4.6: (a) Simplified state machine transition diagram of the PQ. (b) Illustration of PQ timing and memory operation.

4.3.4 Global Timer Synchronization

The global timer synchronizes event communication and tracking across the multi-chip architecture. Although one common crystal oscillator feeds all five FPGAs, their internal system clocks are desynchronized due to phase jitter in the their phase locked loops (PLLs). To remedy timing errors between nodes across the HiAER hierarchy, a global timer in the top-level FPGA emits periodic global time increment events synchronizing local timers in all lower level FPGAs. To prevent accumulation of error due to missed or spurious time increment events, additional timer reset events are globally sent for every 10-bit wrap-around of the top level global timer. These techniques combine to minimize the level of timing skew in the hierarchy.

4.4 Experimental Results

In this section, we present experimental results characterizing latency, throughput, and capacity of synaptic routing through HiAER realized in an FPGA-based prototype embedding two levels of hierarchy with 4-fold branching shown in Fig. 4.7. The HiAER tests are performed for different proof-of-concept configurations of network mappings and input spike rates, and range from a single communication node [64] to the full implemented hierarchy, demonstrating improvements in throughput and latency linear in the number of routing nodes.

4.4.1 HiAER-IFAT Realized Prototype

The hardware system in Fig. 4.7 integrates HiAER reconfigurable synaptic routing implemented using FPGAs and DRAM, with IFAT event-driven conductance-based continuous-time neural dynamics implemented in custom low-power mixed-signal very-large scale integrated circuits [84, 85].

Each quadruple set of HiAER Level 1 nodes (leaves in the hierarchy) shares one Xilinx Spartan 6 FPGA (XC6SLX45T), each sharing two 2 Gb DDR3 DRAMs (Micron MT41J128M16) for synaptic routing table (SRT) storage. Four such units are provided on the board, along with an extra unit serving four HiAER Level 2 nodes,

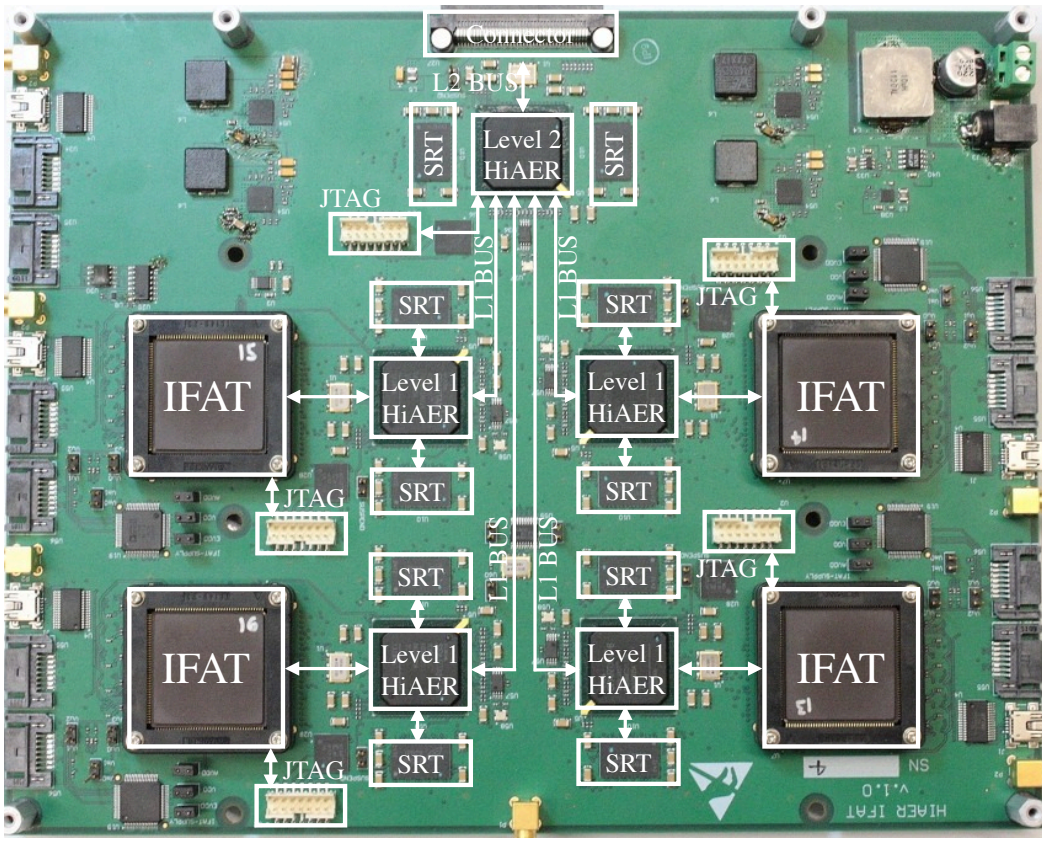


Figure 4.7: Hierarchical Address-Event Routing Integrate-and-Fire Array Transceiver (HiAER-IFAT) for scalable and reconfigurable spike-based neuromorphic computing. Board-level implementation of the HiAER-IFAT architecture with four Level 2 HiAER nodes, each with four Level 1 nodes connected to 2^{16} two-compartment analog Integrate-and-Fire Array Transceiver (IFAT) analog neuron arrays. Each quadruple set of nodes comprises one Spartan 6 FPGA, sharing two 2 Gb DDR3 DRAMs for synaptic routing tables (SRT).

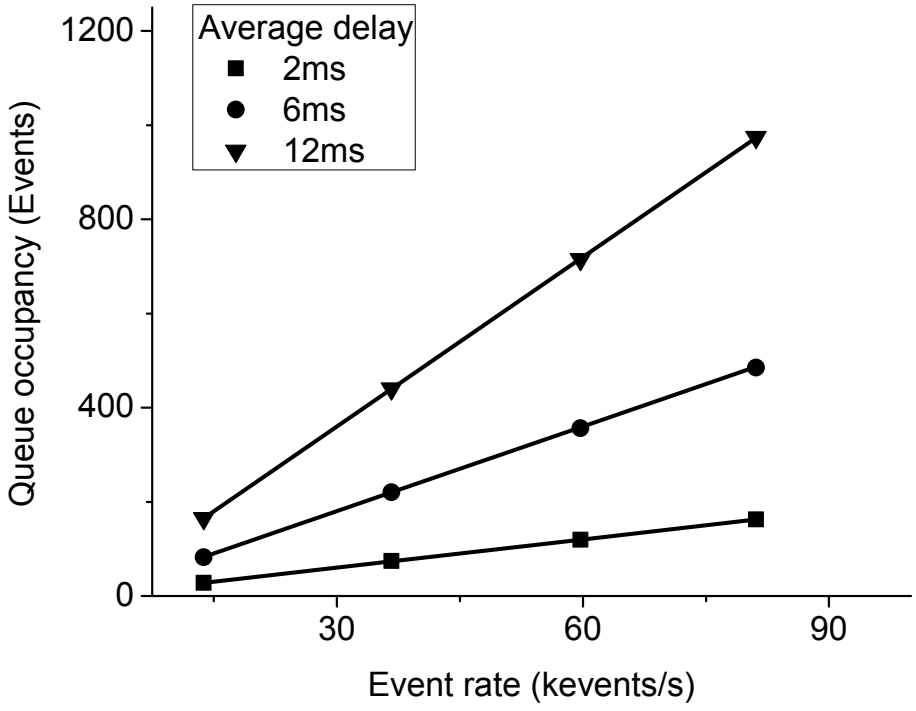
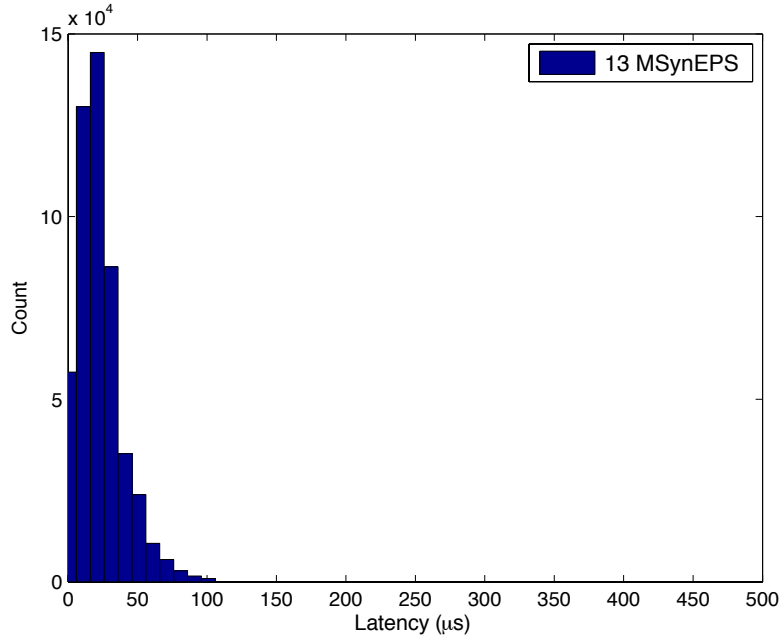
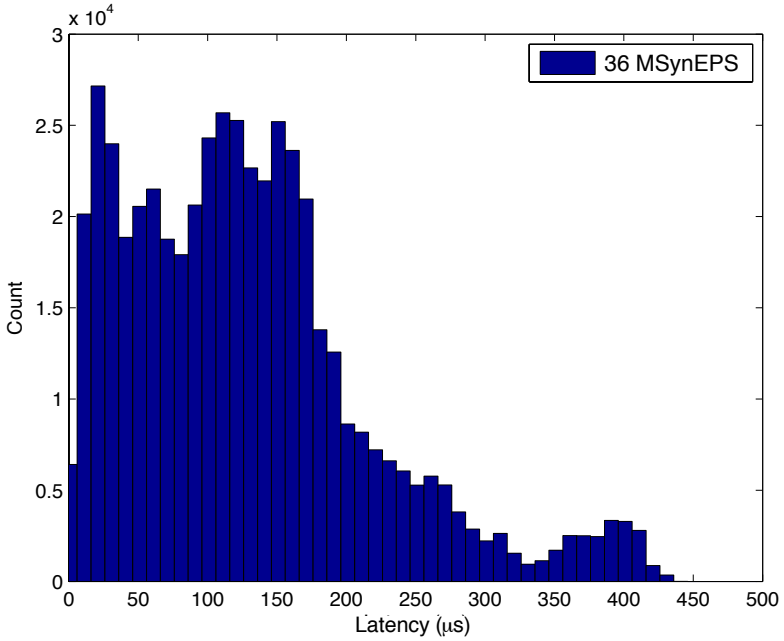


Figure 4.8: Measured data of average priority queue occupancy Q as a function of average event rate r and average axonal conduction delay d. Linear curves show the theoretical model according to Little's law $Q = rd$ [46] for reference.



(a)



(b)

Figure 4.9: Measured latency between presynaptic and postsynaptic events through the Synaptic Routing Table (SRT) at a Level 1 HiAER node (16k neurons), at sustained throughput of 1.3×10^7 synaptic events per second (a) and 3.6×10^7 synaptic events per second (b). The SRT was programmed with uniform 1,000 synaptic fan-out and zero nominal axonal conduction delays ($d = 0$), and the system clock was 150 MHz.

as indicated in Fig. 4.7. The nodes across the FPGAs are interconnected through $L1$ bus parallel communication links as shown. Each FPGA is also equipped with a local 200 MHz clock generator, an external clock input, and USB and JTAG ports for diagnostics and programming. An additional 200 MHz master clock generator can provide all $4+1$ HiAER nodes with a global clock. The system interfaces to the outside, at HiAER Level 3, through the $L2$ bus. Several boards can be combined to form a spike-based neuromorphic computer with more than 2^{18} (262,144) analog integrate-and-fire neurons and high-speed peripherals using different variants of address-event routing protocols, *e.g.,* [5, 20, 44, 88]. The data presented below are obtained by connecting the $L2$ bus of a single HiAER board over a USB 2.0 interface to a workstation.

Each IFAT chip contains four independent ports, each with 16k two-compartment integrate-and-fire neurons [85], and each assigned a single HiAER Level 1 node. The details of neural dynamics in IFAT are beyond of the scope of the present chapter, which focuses on efficiency and scaling in the realization of the HiAER synaptic routing independent of neural integration and spike generation. Indeed HiAER is applicable to a wide range of event-driven large-scale implementations of neural models, *e.g.,* [4, 21, 27, 31, 39, 47, 54, 56, 58, 62, 71, 72, 78, 80, 87].

4.4.2 Experimental Setup

To avoid timing distortion induced by latency of the USB interface between the HiAER $L2$ bus and the workstation, we implemented spike event generators and histogram recorders in FPGA on the board. Spike event generators at the Level 2 HiAER node produce neural event spike trains entering the $L1$ bus with interspike intervals drawn from a Poisson distribution parameterized in mean spike rate. Histogram recorders at each of the Level 1 HiAER nodes take the place of the local IFAT analog array, collecting statistics on time arrivals of received synaptic events while emulating the IFAT's asynchronous AER handshaking of the incoming events. Timing statistics are computed based on time-stamps of received events in relation to the current global timer value. Received events are binned accordingly, with their counts accumulated over a fixed number of trial events.

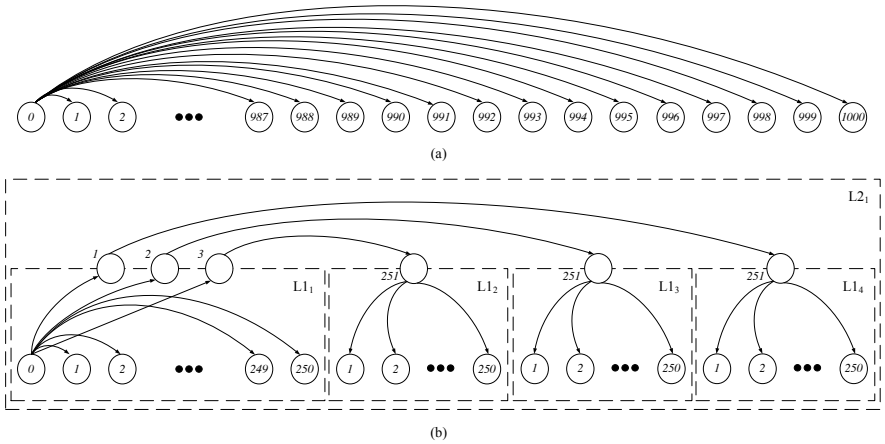


Figure 4.10: Example network partitioning of one presynaptic neuron connecting to 1,000 postsynaptic neurons (a) implemented in single-node flat hierarchy and (b) implemented across two levels of hierarchy partitioned into four HiAER nodes each with 250 postsynaptic neurons.

4.4.3 Priority Queue Analysis

Measured results from the priority queue (PQ) are shown in Fig. 4.8. The event generator was configured to produce Poisson spike trains of variable rate r, modeling varying loads of relay neuron events entering the HiAER node. The events were given Poisson distributed axonal delays d with mean delays of 2 ms, 6 ms, and 12 ms. Little's law [46] predicts the average queue occupancy Q under such conditions to be $Q = rd$ where r is the average incoming event rate and d is the average delay in the queue. Measured results of Q from recorded PQ occupancy data for varying input rate r and average axonal delay d are marked with symbols on the graph in Fig. 4.8, with intersecting straight lines indicating the theoretical fit following Little's law.

Table 4.1: FPGA Resource Usage for Priority Queue Implementation

Queue depth	1,024	2,048	4,096	8,192
Number of slice registers	670	1,195	2,243	4,343
Number of slice LUTs	654	1,234	2,382	4,810
Number of block RAM/FIFO	3	5	9	17

Table 4.1 shows FPGA resource usage for PQ implementation on the target device (Xilinx Spartan-6 XC6SLX45T) for varying queue depth, showing how the implemented PQ on the HiAER board with queue depth 1,024 scales to larger queue sizes,

trading performance for resource usage in approximately linear fashion, limited mainly by total SRAM capacity on the FPGA device.

4.4.4 Event Latency Through Single-node HiAER

Next we analyzed event latency for varying data rate of synaptic events through the Level 1 HiAER node. We again used Poisson event generators with variable spike rate, and measured event latency from histogram recorded data of time-stamp differences over 1 million synaptic events. We implemented an average synaptic fan-out of 1,000 in the SRT, generating on average 1,000 synaptic outgoing events per incoming neural spike event consistent with models of synaptic connectivity in the mammalian central nervous system [16, 76]. However, the axonal delay d was set to zero in order to emulate worst conditions for event throughput and latency: every event entering any PQ is late upon arrival and must exit immediately, accumulating latency in the process. In contrast, events with axonal delay d greater than accumulated propagation delays enter the PQ and resynchronize with the global timer exiting the PQ with near-zero latency. Hence the measured latency for zero axonal delay $d = 0$ should be taken as an upper bound on latency in the general case.

Fig. 4.9 shows measured latency of synaptic output events for two input event rates, indicating latencies below $100\,\mu$s at 1.3×10^7 synaptic events per second (SynEPS) throughput, and latencies below $450\,\mu$s at 3.6×10^7 SynEPS throughput.

4.4.5 Event Latency and Throughput Through Four Parallel HiAER Nodes

To validate improvements in latency and throughput owing to parallelism in hierarchical routing, we conducted experiments with flat and nested structured implementation of simple networks, with presynaptic neurons sharing a common set of 1,000 postsynaptic neurons as illustrated in the example of Fig. 4.10. A single-node implementation with flat hierarchy is shown in Fig. 4.10 (a). The same network is partitioned through three relay neurons into four HiAER nodes each with 250 postsynaptic neurons, shown in Fig. 4.10 (b). The effect of the network partitioning on event latency and thro-

ughput is illustrated in Fig. 4.11. As shown, four-fold partitioning diminishes the local fan-out requirement four-fold leading to approximately four-fold lower event latency, ranging between 0 and $\frac{1}{4}N\tau_{\text{SRT}}$, where $N = 1,000$ is the synaptic fan-out, and τ_{SRT} is the SRT latency per synapse. In addition, the resulting four-fold parallelism in local event routing leads to approximately four-fold increased event throughput across the network, relative to the single-node case. The maximum net synaptic event throughput (or synaptic channel capacity) across all 4 nodes is thus $4 \,/\, \tau_{\text{SRT}}$.

Fig. 4.12 shows measured latency between presynaptic and postsynaptic event through four Level 1 HiAER nodes. For these experiments we used the Poisson spike generator to route 50,000 neural events across the $L1$ bus at 1.23×10^5 events per second. All PQs were cleared of pre-existing events at start of each experiment in order to provide zero initial conditions in event latency. Four parallel HiAER nodes were used for both flat (locally connected) and hierarchical mapping, to equalize net synaptic event channel capacity across both cases. For the flat hierarchy in Fig. 4.10 (a), latency is measured from data collected by the histogram recorder on each of the four HiAER nodes with local 1,000 fan-out. For the four-node two-level hierarchy of Fig. 4.10 (b), latency was measured from data collected across four histogram recorders, one for each HiAER node, each with local 250 fan-out. A shorter tail and narrower distribution is observed in the case of four-fold hierarchical mapping, with worst-case latency of $125\,\mu$s, about a four-fold improvement over the case of flat mapping.

Measured event latency as a function of synaptic event rate, for flat and hierarchical mapping, averaged over all 50M synaptic events from empty PQ initial conditions, is shown in Fig. 4.13. At higher event rates, a four-fold reduction in latency for hierarchical mapping is consistently observed, even beyond the synaptic event channel capacity of 1.44×10^8 SynEPS (synaptic events per second) across the four parallel nodes.

4.5 Conclusions

We presented HiAER, and its efficient implementation in digital hardware, as a hierarchical scalable extension to synaptic address-event routing for large-scale spike-based neuromorphic systems with reconfigurable long-range synaptic connectivity, in

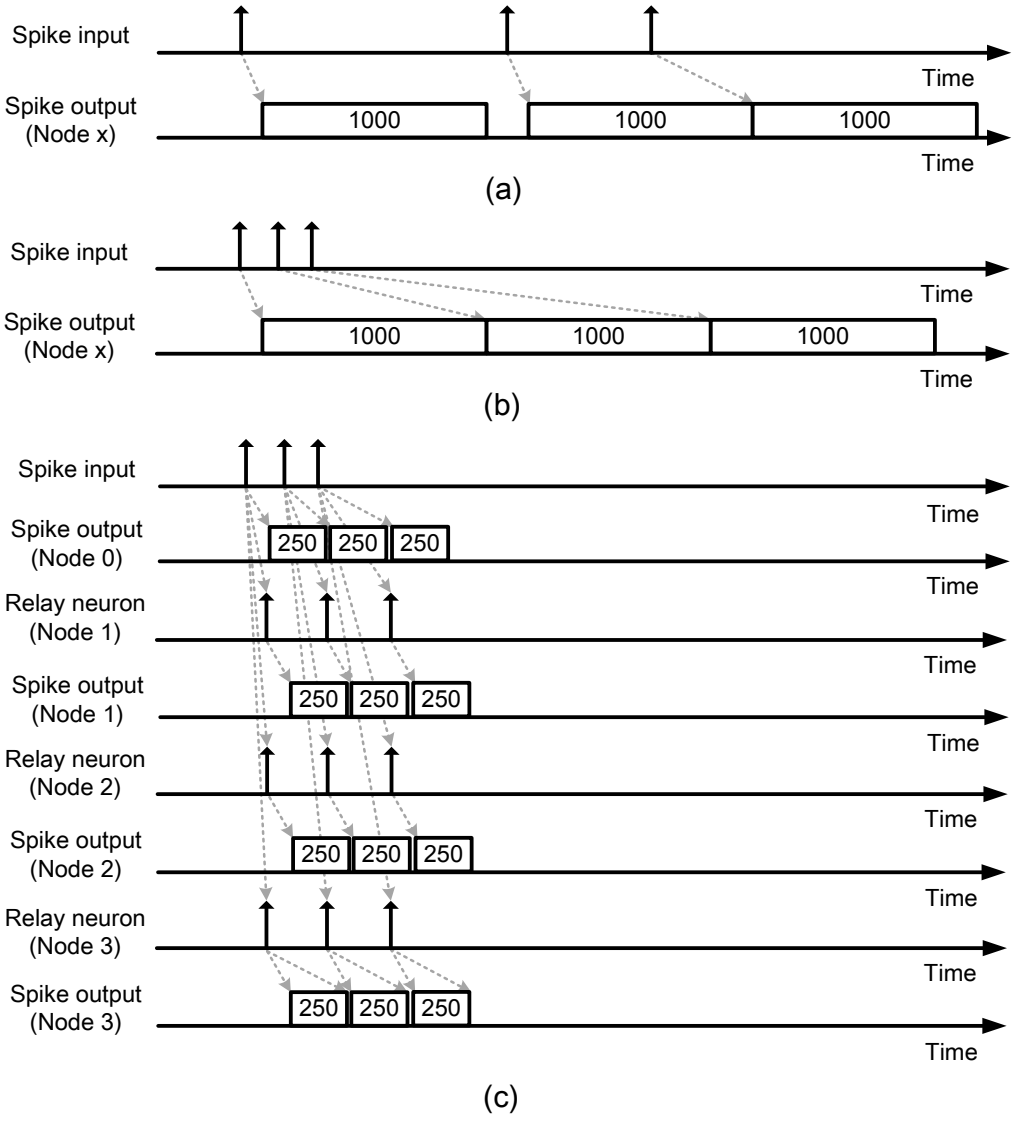
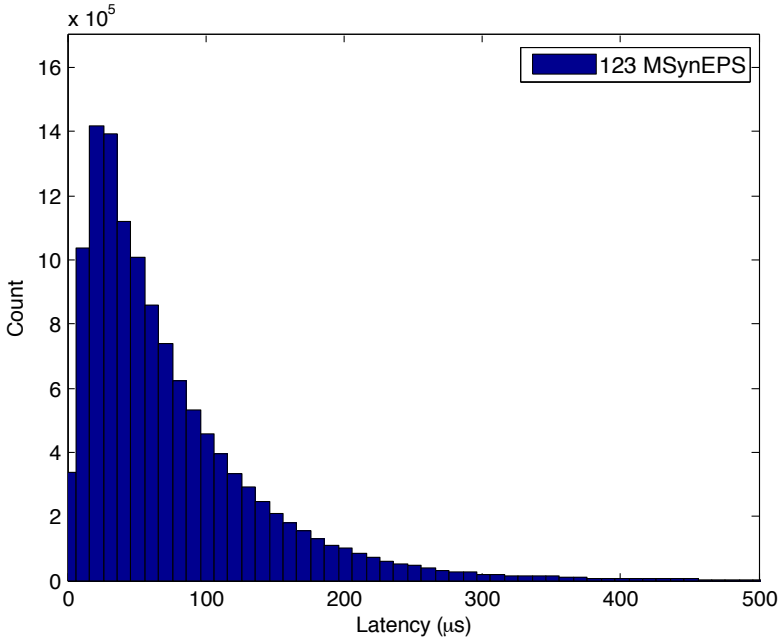
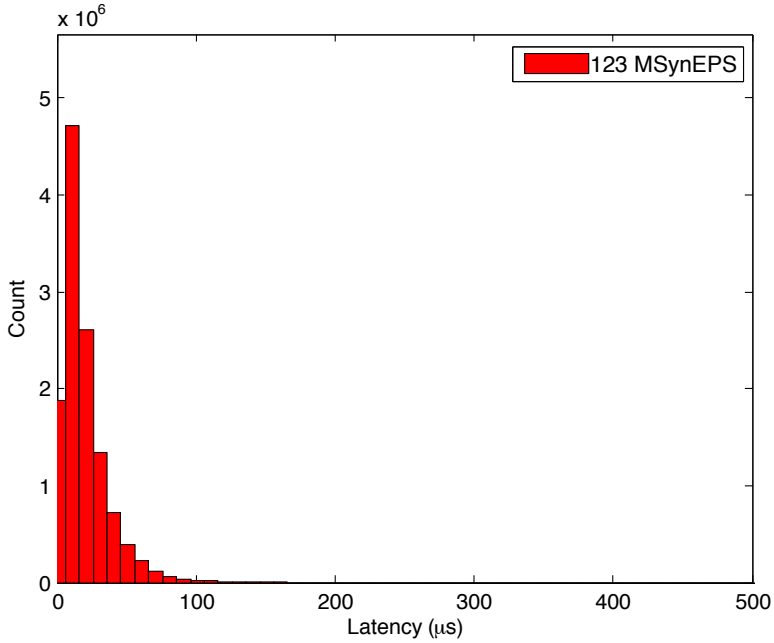


Figure 4.11: Effect of hierarchical network partitioning on event latency and throughput, for the example network in Fig. 4.10. (a) In the single-node flat hierarchy, event latency through the SRT at low neural spike input event rate ranges between 0 and $N\tau_{\text{SRT}}$, where $N = 1,000$ is the synaptic fan-out and τ_{SRT} is the SRT recall latency. (b) Neural spike input event rates greater than its capacity $1/N\tau_{\text{SRT}}$ result in progressively growing event latencies. (c) Partitioning of the network across four HiAER nodes, connected through three relay neurons, results into a four-fold decrease in local synaptic fan-out and, equivalently, event latency. The four-fold parallelism also supports a four-fold greater overall event throughput across the network.



(a)



(b)

Figure 4.12: Measured latency between presynaptic and postsynaptic events through four Synaptic Routing Table (SRT) nodes at the Level 1 HiAER (65 k neurons) in the HiAER-IFAT hierarchy, at sustained throughput of 1.23×10^8 synaptic events per second (SynEPS) with flat mapping (a) and four-fold hierarchical mapping (b) of the network in Fig. 4.10.

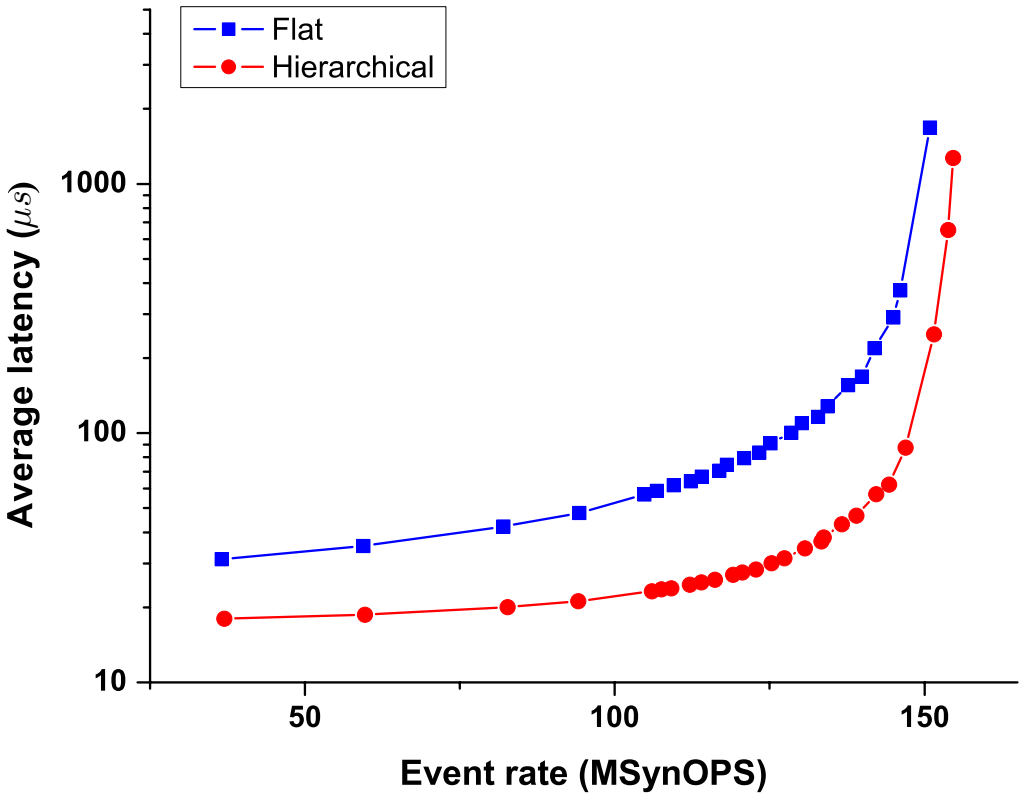


Figure 4.13: Average event latency measured as a function of synaptic event rate for flat and four-fold hierarchical partitioning of the network in Fig. 4.10.

which both strength and axonal delay for each implemented synapse are individually programmable. As a proof-of-concept, a two-level four-fold branching hierarchy with 262k two-compartment integrate-and-fire neurons, each fanning out to any other neurons with thousand synapses on average, was implemented on a custom PCB with 5 Xilinx Spartan-6 FPGAs, 10 DDR3 DRAMs, and 4 custom IFAT mixed-signal VLSI microchips. At the single-board level we demonstrated approximately linear scaling in throughput of global synaptic event routing at 36 MsynEPS per 16k-neuron node in the hierarchy. We also showed decreased event latency, from 83.6 μs for flat partitioning to 28.3 μs for 4-fold hierarchical partitioning owing to the corresponding reduction of local connectivity in the distributed network. Furthermore we showed average queue occupancy in the PQs consistent with Little's law, with 12 ms of average axonal delay at 8×10^4 events/s relay neuron event rate per HiAER routing node for the implemented 1,024 queue depth in FPGA SRAM.

Larger-size networks, in principle of unlimited size, may be obtained by cascading boards to extend the HiAER hierarchy to higher levels at net synaptic throughput scaling with the number of nodes across the hierarchy [36]. Hierarchical partitioning of axonal delay may further support temporal spike-based models of neural computation based on pattern matching in delayed spike coincidence detection [34] at virtually unlimited range of delays. Conversely, recently developed stochastic rate-based models with Monte Carlo Markov chain (MCMC) neural sampling from Boltzmann distributions in large-scale spiking networks with biophysical integrate-and-fire neurons [67] and their extensions to on-line learning spike-based Boltzmann machines [61] map directly onto the HiAER architecture as well.

The challenges in further scaling up hardware realizations of HiAER are multifold, calling for advances in:

Area and energy efficiency Measuring 20 cm $\times$ 25 cm and consuming 10 W of power at 720 MSynEPS net synaptic throughput across five FPGAs, the presented 262k neuron, 262M synapse implementation offers an area efficiency of 200 μm^2 per synapse and an energy efficiency of 14 nJ per synaptic event. Although a respectable feat of neuromorphic engineering, the realized efficiencies pale in comparison with the 10^{-3} μm^2 area

and 2 fJ energy per synapse for the human brain which counts roughly 10^{15} synapses, each activated on average at 10 Hz, within 0.002 m^3 volume and across 1 m^2 cortical surface area, and at 20 W of metabolic power consumption [10, 16, 22, 76]. The HiAER realized efficiencies are limited by DRAM memory cell density and read energy in serial access of SRTs, and by the FPGA general-purpose reconfigurable logic. Significant area and energy improvements can be expected from custom silicon integration of SRTs distributed across HiAER routing nodes, such as using wafer-scale integration [21, 58] or vertically stacked 3-D integration of CMOS and memory technologies [37, 40]. Further energy improvements may also result from direct asynchronous synthesis of all HiAER event routing, including PQ, FIFOs, and possibly DRAM memory controller. The advantage of asynchronous implementation, in the absence of any clock, is that power scales directly with event rate, except for static standby power [29, 53].

Efficient partitioning Efficient use of HiAER resources is critically dependent on efficient partitioning of the implemented network into a hierarchy of clusters that minimizes event traffic across routing nodes. The general problem of efficient hierarchical graph partitioning is well studied, and solutions formulated in various application domains, *e.g.,* [38] may be ported to hierarchical synaptic partitioning, in tandem with compilation and analysis tools for efficient mapping of the hierarchical neural and synaptic structure onto neuromorphic architecture [17, 36, 65]. In addition, anatomical and functional connectivity information gathered from connectomics [2, 35] may guide naturally efficient network partitioning inspired by the structural organization of the central nervous system.

Efficient learning Although not pursued here, HiAER may be extended with local mechanisms of spike-timing dependent plasticity (STDP) implemented directly in the address domain [81] to learn the HiAER long-range synaptic connectivity on-line from real-time data. STDP-based models of temporally asymmetric Hebbian unsupervised learning extend to other forms of spike-based learning such as reinforcement learning of distal reward using STDP-modulated dopamine signaling [33], and deep learning of multi-layered cortical representations using STDP event-driven contrastive divergence in spiking Boltzmann machines [61]. The advantage of HiAER for efficient hierarchical

event-driven implementation of STDP-based on-line learning is that all information on synaptic strength, regardless of global range in connectivity, resides only in local SRTs at the final destination (Level 1 HiAER) leaf nodes in the hierarchy, in direct proximity to both presynaptic and postsynaptic event streams. Thus local implementation of event-driven STDP at Level 1 HiAER SRTs may be sufficient to support more general implementation of complex non-local learning rules that take advantage of global nested network structure with the long-range and hierarchical connectivity provided by HiAER.

Chapter 4 is largely a reprint of material that was submitted to 2014 Transaction on Neural Networks and Learning Systems : J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical Address Event Routing for Reconfigurable Large-Scale Neuromorphic Systems", *IEEE Transaction on Neural Networks and Learning Systems.* The author is the primary author and investigator of this paper.

Chapter 5

Conclusions

This dissertation presented a low-power VLSI systems for event-driven sensory and neural processing combining high efficiency in general spatiotemporal signal event coding at the sensor front end, with programmable and adaptive signal specificity in parsing the event stream at the system level, relaxing resolution requirements in the front end circuits while reducing the data rate according to the information content of the signal.

In chapter 2, we presented a temporal contrast detection event-driven asynchronous 128×128 pixel image sensor with integrated 10-bit ADC intensity readout in $0.18\,\mu$m 6M 1P CMOS. Current-mode log-compressed readout with correlated double sampling (CDS) offset compensation provides random access to instantaneous pixel intensity over 75 dB intra-scene dynamic range. Temporal correlated double sampling in "on" and "off" transient event detection reduces capacitor sizing requirements within the pixel. Global threshold adaptation using common-mode capacitive coupling across the pixel array compensates for global scene lighting variations. Power consumption at 8.3 Mevents/s is 12.6 mW from a 1.8 V supply, or 1.52 nJ per pixel event detection and readout. Die size is $5\times5\,\text{mm}^2$ and pixel size is $33\times33\,\mu\text{m}^2$ with 22.8% fill factor fabricated in $0.18\,\mu$m 6M 1P CMOS.

In chapter 3, we presented a 65 k-neuron integrate-and-fire array transceiver (IFAT) for event-based neural computation. The internally analog, externally digital chip is fabricated on a $4\times4\,\text{mm}^2$ die in 90 nm CMOS and arranged in four quadrants of 16 k parallel addressable neurons. Each neuron circuit serves input spike events by

dynamically instantiating conductance-based synapses onto four local synapse circuits over two membrane compartments, and produces output spike events upon reaching a threshold in integration over one of the membrane compartments. Fully asynchronous input and output spike event data streams are mediated over the standard address event representation (AER) protocol. To support full event throughput at large synaptic fan-in, a two-tier micro-pipelining scheme parallelizes input events along neural array cores, and along rows of each core. Measured results show sustained peak synaptic event throughput of 18.2 Mevents/s per quadrant, at 19.2 pJ average energy per synaptic input event and 25 μW standby power.

In chapter 4, we presented a Hierarchical Address Event Routing (HiAER) architecture for scalable communication of neural and synaptic spike events between neuromorphic processors, implemented with 5 Xilinx Spartan-6 FPGAs and 4 custom analog neuromorphic ICs serving 262k neurons and 262M synapses. The architecture extends the single-bus address event representation (AER) protocol to a hierarchy of multiple nested buses, routing events across increasing scales of spatial distance. The HiAER protocol provides individually programmable axonal delay in addition to strength for each synapse, lending itself towards biologically plausible neural network architectures, and scales across a range of hierarchies suitable for multi-chip and multi-board systems in reconfigurable large scale neuromorphic systems. We show approximately linear scaling of net global synaptic event throughput with number of routing nodes in the network, at 3.6×10^7 synaptic events per second per 16k-neuron node in the hierarchy.

Concluding this dissertation, we outlook possible projects combining this thesis work in a integrated chip with hybrid fabrication technology will be shown in next section.

5.1 Outlook

5.1.1 3-D Neuromorphic Processor (HiAER-IFAT)

3-D neuromorphic processor is a fully reconfigurable massively parallel network of 65k biologically inspired, two-compartment spiking neurons and 65M dynamic synapses with arbitrary, global connection topology. 3-D packaging of the chips offers

unprecedented capabilities in large scale modeling of the nervous system, and in the scale and density of neuromorphic systems for tactical and commercial applications in pattern recognition and machine perception.

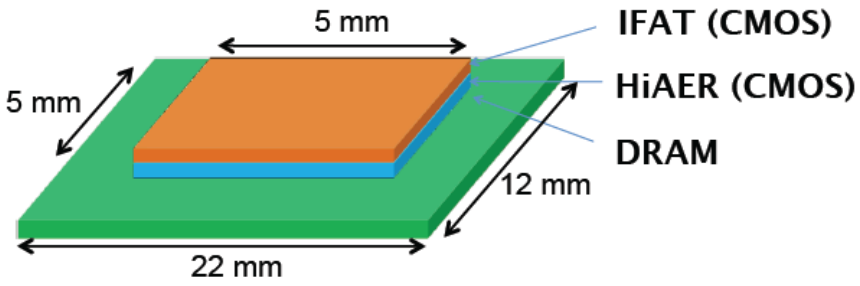
Fig. 5.1 is triple-stack 3D integrated neuromorphic processor. The top layer of the CMOS stack implements a mixed-signal array of 130k neuron compartments, with event-based addressing. Lateral connectivity between the neuron compartments implements multi compartment models of neural computation with up to 65k biologically realistic neurons. A digital CMOS process is ideally suited for implementing the neuromorphic mixed-signal VLSI neural circuits at high density and high energy efficiency.

The bottom layer of the CMOS stack implements registration, interpretation, and routing of neural and synaptic events to sustain high bandwidth of neural interconnectivity while ensuring arbitrary connectivity and synaptic plasticity of the network. The bottom layer communicates incoming and outgoing neural events to the top neural layer through high-bandwidth vertical interconnects. The bottom logic layer also makes use of high-bandwidth DRAM interconnects for storage and recall of neural connectivity and synaptic parameters.

5.1.2 3-D Neuromorphic Silicon Retina

3D neuromorphic silicon retina will be the first to emulate the detailed spatiotemporal dynamics of ganglion cell visual coding in the mammalian retina, while offering ultra-low power operation and high integration density as well as high fill factor. We will combine state-of-the art approaches in CMOS imaging and neuromorphic computing on a single 3-D integrated substrate through wafer stacking of three traditionally disparate CMOS technologies that are each tailored for optimal performance: high light-sensitivity in the photoreceptor array on the top layer in an optoelectronic CMOS process, low-power and high-density asynchronous neural event coding and communication in the middle layer in a deep-submicron CMOS process, and dedicated memory for reconfigurable spatiotemporal dynamics in the base layer in DRAM technology. Fig 5.2 is showing proposed 3D Imager.

Compared to conventional silicon retinas, the proposed design will have the following features and advantages:



- IFAT : Analog integrate-and-fire array transceiver
  - 65k analog continuous-time spiking neurons
  - Two compartments and four dynamical synapse types each
  - Asynchronous spike event I/O interface
- HiAER : Hierarchical address-event routing
  - Locally dense, globally sparse synaptic interconnectivity
- DRAM : Synaptic routing table
  - 65M digitally programmable synapses
  - Reconfigurable, arbitrary topology

Figure 5.1: Triple-stack 3D integrated neuromorphic processor

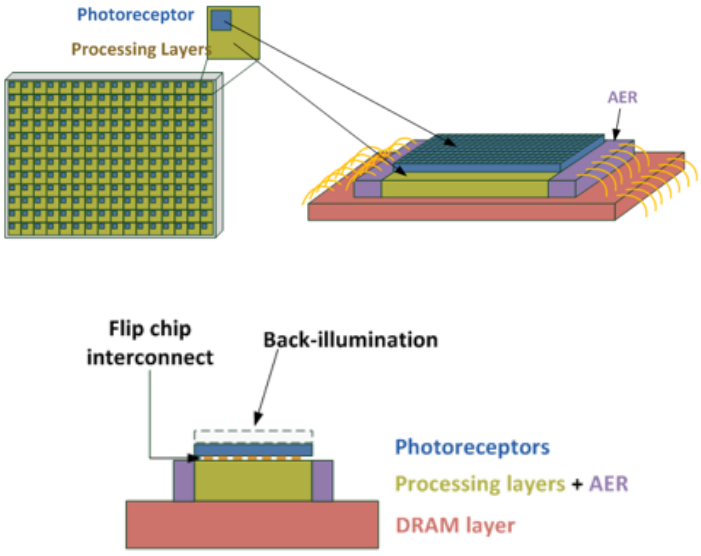


Figure 5.2: Implementation of 2D and 3D versions of the silicon retina chip

Precision-on-demand spatiotemporal resolution by asynchronous event-driven communication We adopt event-driven information processing, which is frame-less and asynchronous, so much more energy-efficient than frame-based processing.

Biologically realistic spatiotemporal filtering The proposed chip will implement at least four types of ganglion cells, covering both on/off sustain and on/off transient responses. These implemented models will account for the spatiotemporal dynamics of horizontal and amacrine cells in the retina

Low supply design Despite that virtually of all neuromorphic circuits operate in the sub-threshold voltage region, there has not a single chip implemented in ultra-low power supply such as 0.5V. The three layers of the proposed chip will be implemented with 0.5V power supply.

Chip-level light adaptability Modeling the iris in regulating the amount of light entering the retina, it propose chip-level light adapting scheme to enhance dynamic range and improve signal-to-noise ratio (SNR) simultaneously. Conventional approaches have

enhanced the dynamic range by linear-to-logarithmic conversion with a resulting loss of SNR in each pixel. By moving the burden of providing a wide dynamic range from each pixel to external control, a more area and power efficient design is possible for the pixel. Each pixel may further adapt its dynamic range, complementing the control from the chip-level light intensity detection circuit, which may detect light intensity level by measuring current consumption of the whole chip.

3-D structure with function-dedicated layers The retina chip will consist of three layer of stacking chips. The most top layer will be fabricated for high-performance in light detection, and will be back illuminated to connect to the middle layer by flip-chip bonding. The middle layer will be fabricated to achieve high-density integration and low power consumption. Digital control for address-event representation (AER) will be further integrated in the middle layer. The middle layer will connect by bonding wires to the bottom layer with DRAMs working memory

Appendix A

IFAT User Guide

A.1 IFAT Pin Definitions

Following tables list the pin definitions used in IFAT.

Table A.1: IFAT Power and Ground

Name	Description	Level (V)
DVDD	Power for I/O ring	1.8
VDD	Power for chip core	1.2
AVDD	Power for analog section	1.2
EVDD	Power for event arbitration module	1.2
GVDD	Power for guardring	1.8
VSS	Ground	0

Table A.2: IFAT Analog Pin Definitions

Name	Description
VBIAS_UNITYBUFFER	Voltage bias governing the gate voltage of unity gain buffer current source
VBIAS_SF	Voltage bias governing the gate voltage of source follower for analog output probe
VAMP_LOW	Lowest voltage reference for voltage divider selecting amplitude of PWAM
VAMP_HIGH	Highest voltage reference for voltage divider selecting amplitude of PWAM
VWIDTH_BIAS_LOW	Lowest voltage reference for voltage divider selecting width of PWAM
VWIDTH_BIAS_HIGH	Highest voltage reference for voltage divider selecting width of PWAM
VINPUT_CURRENT_BIAS	Voltage bias governing the gate voltage of current source for width modulation in PWAM
VWIDTH_BIAS	Voltage bias governing the gate voltage of comparator current source
VTRANSCAP_BODY	Body bias voltage of transcap transistor
VBP	pMOS voltage bias governing the gate voltage of the pMOS in the feedback loop in the row and column arbiters
VBN	nMOS voltage bias governing the gate voltage of the nMOS at the input to the NAND gates of the buffered outputs and feedback signals in the row and column arbiters
SYNAPSE_DRIVE_LOW	Lowest voltage level of synaptic driving voltage level when it is not driven
Continued on next page	

Table A.2 IFAT Analog Pin Definitions, Continued.

Name	Description
VCAPBIAS	Voltage bias governing the gate voltage of transcap transistor
VTHRESH	Threshold level for spiking neuron action potentials
VSPIKE	Maximum voltage level for neuron action potentials
VSS_SYNAPSE	Lowest voltage level of synapse variable
GCOMP	Signal at the gate of a single transistor implementing the conductance connection between neuron compartments
VPDN	nMOS voltage bias governing the gate voltage of the row and column request lines in the neurons
VBIAS	pMOS voltage bias governing the gate voltage of the pMOS current load at the input of the neurons
VRESET	Reset level for spiking neuron action potentials
VPUP	pMOS voltage bias governing the gate voltage of the pMOS at the input of the row and column arbiters
VPUP_REQ	pMOS voltage bias governing the gate voltage of the column and row request inputs
EREV<3:0>	Reversal potential value for synapse 0-3
VTAU<3:0>	Tau voltage value governing the membrane dynamics profile for synapse 0-3
ELEAK	Reversal potential leakage parameter for each neuron compartment
GLEAK_PROXIMAL	Signal at the gate of a single transistor implementing the conductance leakage from proximal compartment
GLEAK_DISTAL	Signal at the gate of a single transistor implementing the conductance leakage from distal compartment
V_U_PROBE<3:0>	For Vu, synapse 0-3, voltage buffered through source follower

Continued on next page

Table A.2 IFAT Analog Pin Definitions, Continued.

Name	Description
VMEM PROXIMAL PROBE	For proximal compartment membrane voltage, voltage buffered through source follower
VMEM DISTAL PROBE	For distal compartment membrane voltage, voltage buffered through source follower

Table A.3: IFAT Digital Pin Definitions

Name	Description
INPUT_DATA_ACK	Acknowledge signal for address event from system to chip
INPUT_DATA_REQ	Request signal for address event from system to chip
INPUT_DATA<23:0>	24 bit address event from system to chip
OUTPUT_DATA_ACK	Acknowledge signal for address event from chip to system
OUTPUT_DATA_REQ	Request signal for address event from chip to system
OUTPUT_DATA<13:0>	14 bit address event from chip to system
PULSE_GEN_RST	Active high reset signal for pulse width and amplitude modulation (PWAM) modules
RESETLFSR	Active high reset signal for LFSR
RESET_IFAT	Active high reset signal for the IFATs
ASYNC_RST	Active high reset signal for asynchronous modules
SHIFT_REG_RST	Active high reset signal for the shift registers
SHIFT_REG_D	SPI input for serial chain register selecting a neuron to probe buffered analog output
COL_SHIFT_REG_CLK	Clock for serial chain of column select register
ROW_SHIFT_REG_CLK	Clock for serial chain of row select register
ARRAY_SHIFT_REG_CLK	Clock for serial chain of array select register

A.2 IFAT Pinout Table

This section includes the pinout information table for the IFAT packaging.

Table A.4: IFAT Pin Name

Number	Pin Name	Pin Number
1	DVDDL	G5
2	VSS	A1
3	VDDL	J5
4	PORT0_OUTPUT_DATA<11>	E2
5	PORT0_OUTPUT_DATA<12>	D4
6	PORT0_OUTPUT_DATA<13>	E1
7	VSS	R15
8	VDDL	K5
9	PORT0_ASYNC_RST	D1
10	PORT0_RESET_IFAT	F2
11	PORT0_RESETLFSR	E4
12	PORT0_PULSE_GEN_RST	F1
13	PORT0_ROW_SHIFT_REG_CLK	E3
14	PORT0_COL_SHIFT_REG_CLK	G2
15	PORT0_INPUT_DATA<0>	F4
16	PORT0_INPUT_DATA<1>	G1
17	PORT0_INPUT_DATA<2>	F3
18	PORT0_INPUT_DATA<3>	H2
19	PORT0_INPUT_DATA<4>	G4
20	PORT0_INPUT_DATA<5>	H1
21	PORT0_INPUT_DATA<6>	G3
22	PORT0_INPUT_DATA<7>	J2

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
23	PORT0_INPUT_DATA<8>	H4
24	PORT0_INPUT_DATA<9>	J1
25	PORT0_INPUT_DATA<10>	H3
26	PORT0_INPUT_DATA<11>	K2
27	PORT0_INPUT_DATA<12>	J4
28	PORT0_INPUT_DATA<13>	K1
29	PORT0_INPUT_DATA<14>	J3
30	PORT0_INPUT_DATA<15>	L2
31	DVDDL	H5
32	VSS	B1
33	VDDL	L5
34	VSS	L15
35	PORT0_INPUT_DATA<16>	K4
36	PORT0_INPUT_DATA<17>	L1
37	PORT0_INPUT_DATA<18>	K3
38	PORT0_INPUT_DATA<19>	M2
39	PORT0_INPUT_DATA<20>	L4
40	PORT0_INPUT_DATA<21>	M1
41	PORT0_INPUT_DATA<22>	L3
42	PORT0_INPUT_DATA<23>	N1
43	PORT0_INPUT_DATA_REQ	M4
44	PORT0_INPUT_DATA_ACK	N2
45	PORT0_SHIFT_REG_D	M3
46	PORT0_SHIFT_REG_RST	P1
47	PORT0_ARRAY_SHIFT_REG_CLK	N3
48	EVDD0	M5
49	VSS	N13
50	AVDDLU	N5

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
51	PORT0_VBIAS_SF	P2
52	PORT0_VBIAS_UNITYBUFFER	N4
53	GVDD	P5
54	VSS	P13
55	GVDD	R5
56	VSS	R13
57	PORT2_VBIAS_UNITYBUFFER	R1
58	PORT2_VBIAS_SF	P4
59	AVDDLB	U5
60	VSS	T13
61	EVDD2	T5
62	PORT2_ARRAY_SHIFT_REG_CLK	R2
63	PORT2_SHIFT_REG_RST	P3
64	PORT2_SHIFT_REG_D	T1
65	PORT2_INPUT_DATA_ACK	R3
66	PORT2_INPUT_DATA_REQ	T2
67	PORT2_INPUT_DATA<23>	R4
68	PORT2_INPUT_DATA<22>	U1
69	PORT2_INPUT_DATA<21>	T3
70	PORT2_INPUT_DATA<20>	U2
71	PORT2_INPUT_DATA<19>	T4
72	PORT2_INPUT_DATA<18>	V1
73	PORT2_INPUT_DATA<17>	U3
74	PORT2_INPUT_DATA<16>	V2
75	VSS	N15
76	VDDLB	V5
77	VSS	AE1
78	DVDDLB	AA5

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
79	PORT2_INPUT_DATA<15>	U4
80	PORT2_INPUT_DATA<14>	W1
81	PORT2_INPUT_DATA<13>	V3
82	PORT2_INPUT_DATA<12>	W2
83	PORT2_INPUT_DATA<11>	V4
84	PORT2_INPUT_DATA<10>	Y1
85	PORT2_INPUT_DATA<9>	W3
86	PORT2_INPUT_DATA<8>	Y2
87	PORT2_INPUT_DATA<7>	W4
88	PORT2_INPUT_DATA<6>	AA1
89	PORT2_INPUT_DATA<5>	Y3
90	PORT2_INPUT_DATA<4>	AA2
91	PORT2_INPUT_DATA<3>	Y4
92	PORT2_INPUT_DATA<2>	AB1
93	PORT2_INPUT_DATA<1>	AA3
94	PORT2_INPUT_DATA<0>	AB2
95	PORT2_COL_SHIFT_REG_CLK	AA4
96	PORT2_ROW_SHIFT_REG_CLK	AC1
97	PORT2_PULSE_GEN_RST	AB3
98	PORT2_RESETLFSR	AC2
99	PORT2_RESET_IFAT	AB4
100	PORT2_ASYNC_RST	AD1
101	VDDLB	W5
102	VSS	T15
103	PORT2_OUTPUT_DATA<13>	AC3
104	PORT2_OUTPUT_DATA<12>	AD2
105	PORT2_OUTPUT_DATA<11>	AC4
106	VDDLB	Y5

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
107	VSS	AF1
108	DVDDLB	AB5
109	VSS	L16
110	PORT2_OUTPUT_DATA<10>	AC5
111	PORT2_OUTPUT_DATA<9>	AD4
112	PORT2_OUTPUT_DATA<8>	AE3
113	PORT2_OUTPUT_DATA<7>	AD6
114	PORT2_OUTPUT_DATA<6>	AE4
115	PORT2_OUTPUT_DATA<5>	AC6
116	PORT2_OUTPUT_DATA<4>	AD5
117	PORT2_OUTPUT_DATA<3>	AB6
118	PORT2_OUTPUT_DATA<2>	AE5
119	PORT2_OUTPUT_DATA<1>	AD7
120	PORT2_OUTPUT_DATA<0>	AF2
121	PORT2_OUTPUT_DATA_REQ	AC7
122	PORT2_OUTPUT_DATA_ACK	AF3
123	VSS	R11
124	AVDDB	AA22
125	VSS	T11
126	PORT2_VTRANSCAP_BODY	AB7
127	PORT2_VBP	AE6
128	PORT2_VBN	AD8
129	PORT2_SYNAPSE_DRIVE_LOW	AF4
130	PORT2_VCAPBIAS	AC8
131	PORT2_ELEAK	AF5
132	PORT2_VTHRESH	AB8
133	PORT2_GLEAK_DISTAL	AE7
134	PORT2_VSPIKE	AD9

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
135	PORT2_E_REV<1>	AF6
136	PORT2_V_TAU<2>	AC9
137	PORT2_V_TAU<0>	AE8
138	PORT2_VSS_SYNAPSE	AB9
139	VSS	L12
140	PORT2_VAMP_LOW	AD10
141	PORT2_VAMP_HIGH	AF7
142	PORT2_VWIDTH_BIAS_LOW	AC10
143	PORT2_VWIDTH_BIAS_HIGH	AE10
144	PORT2_VINPUT_CURRENT_BIAS	AB10
145	PORT2_VWIDTH_BIAS	AE9
146	PORT2_V_U_PROBE<1>	AD11
147	PORT2_V_U_PROBE<0>	AF8
148	PORT2_VMEM_PROXIMAL_PROBE	AC11
149	PORT2_VMEM_DISTAL_PROBE	AE11
150	PORT2_V_U_PROBE<3>	AB11
151	PORT2_V_U_PROBE<2>	AF9
152	PORT2_GCOMP	AD12
153	PORT2_E_REV<0>	AF10
154	PORT2_E_REV<3>	AC12
155	PORT2_V_TAU<3>	AF11
156	PORT2_V_TAU<1>	AB12
157	PORT2_VPDN	AE12
158	PORT2_VBIAS	AD13
159	PORT2_VRESET	AF12
160	PORT2_E_REV<2>	AC13
161	PORT2_GLEAK_PROXIMAL	AF13
162	PORT2_VPUP	AB13

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
163	PORT2_VPUP_REQ	AE13
164	PORT3_VPUP_REQ	AB14
165	PORT3_VPUP	AE14
166	PORT3_GLEAK_PROXIMAL	AC14
167	PORT3_E_REV<2>	AF14
168	PORT3_VRESET	AD14
169	PORT3_VBIAS	AF15
170	PORT3_VPDN	AB15
171	PORT3_V_TAU<1>	AE15
172	PORT3_V_TAU<3>	AC15
173	PORT3_E_REV<3>	AF16
174	PORT3_E_REV<0>	AD15
175	PORT3_GCOMP	AF17
176	PORT3_V_U_PROBE<2>	AB16
177	PORT3_V_U_PROBE<3>	AE16
178	PORT3_VMEM_DISTAL_PROBE	AC16
179	PORT3_VMEM_PROXIMAL_PROBE	AF18
180	PORT3_V_U_PROBE<0>	AD16
181	PORT3_V_U_PROBE<1>	AE17
182	PORT3_VWIDTH_BIAS	AB17
183	PORT3_VINPUT_CURRENT_BIAS	AF19
184	PORT3_VWIDTH_BIAS_HIGH	AC17
185	PORT3_VWIDTH_BIAS_LOW	AE18
186	PORT3_VAMP_HIGH	AD17
187	PORT3_VAMP_LOW	AF20
188	VSS	M12
189	PORT3_VSS_SYNAPSE	AF21
190	PORT3_V_TAU<0>	AB18

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
191	PORT3_V_TAU<2>	AE19
192	PORT3_E_REV<1>	AC18
193	PORT3_VSPIKE	AF22
194	PORT3_GLEAK_DISTAL	AD18
195	PORT3_VTHRESH	AE20
196	PORT3_ELEAK	AB19
197	PORT3_VCAPBIAS	AF23
198	PORT3_SYNAPSE_DRIVE_LOW	AC19
199	PORT3_VBN	AF24
200	PORT3_VBP	AD19
201	PORT3_VTRANSCAP_BODY	AE21
202	VSS	N12
203	AVDDB	AB22
204	VSS	P12
205	PORT3_OUTPUT_DATA_ACK	AB20
206	PORT3_OUTPUT_DATA_REQ	AE22
207	PORT3_OUTPUT_DATA<0>	AC20
208	PORT3_OUTPUT_DATA<1>	AE23
209	PORT3_OUTPUT_DATA<2>	AD20
210	PORT3_OUTPUT_DATA<3>	AD22
211	PORT3_OUTPUT_DATA<4>	AC21
212	PORT3_OUTPUT_DATA<5>	AC22
213	PORT3_OUTPUT_DATA<6>	AB21
214	PORT3_OUTPUT_DATA<7>	AE24
215	PORT3_OUTPUT_DATA<8>	AD21
216	PORT3_OUTPUT_DATA<9>	AF25
217	PORT3_OUTPUT_DATA<10>	AD23
218	VSS	M16

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
219	DVDDRB	W22
220	VSS	N11
221	VDDRB	T22
222	PORT3_OUTPUT_DATA<11>	AC25
223	PORT3_OUTPUT_DATA<12>	AD24
224	PORT3_OUTUPT_DATA<13>	AC26
225	VSS	N16
226	VDDRB	U22
227	PORT3_ASYNC_RST	AD25
228	PORT3_RESET_IFAT	AB25
229	PORT3_RESETLFSR	AC23
230	PORT3_PULSE_GEN_RST	AB26
231	PORT3_ROW_SHIFT_REG_CLK	AC24
232	PORT3_COL_SHIFT_REG_CLK	AA25
233	PORT3_INPUT_DATA<0>	AB23
234	PORT3_INPUT_DATA<1>	AA26
235	PORT3_INPUT_DATA<2>	AB24
236	PORT3_INPUT_DATA<3>	Y25
237	PORT3_INPUT_DATA<4>	AA23
238	PORT3_INPUT_DATA<5>	Y26
239	PORT3_INPUT_DATA<6>	AA24
240	PORT3_INPUT_DATA<7>	W25
241	PORT3_INPUT_DATA<8>	Y23
242	PORT3_INPUT_DATA<9>	W26
243	PORT3_INPUT_DATA<10>	Y24
244	PORT3_INPUT_DATA<11>	V25
245	PORT3_INPUT_DATA<12>	W23
246	PORT3_INPUT_DATA<13>	V26

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
247	PORT3_INPUT_DATA<14>	W24
248	PORT3_INPUT_DATA<15>	U25
249	DVDDRB	Y22
250	VSS	P11
251	VDDRB	V22
252	VSS	P15
253	PORT3_INPUT_DATA<16>	V23
254	PORT3_INPUT_DATA<17>	U26
255	PORT3_INPUT_DATA<18>	V24
256	PORT3_INPUT_DATA<19>	T25
257	PORT3_INPUT_DATA<20>	U23
258	PORT3_INPUT_DATA<21>	T26
259	PORT3_INPUT_DATA<22>	U24
260	PORT3_INPUT_DATA<23>	R25
261	PORT3_INPUT_DATA_REQ	T23
262	PORT3_INPUT_DATA_ACK	R26
263	PORT3_SHIFT_REG_D	T24
264	PORT3_SHIFT_REG_RST	P26
265	PORT3_ARRAY_SHIFT_REG_CLK	R23
266	EVDD3	R22
267	VSS	R12
268	AVDDRB	P22
269	PORT3_VBIAS_SF	P25
270	PORT3_VBIAS_UNITYBUFFER	R24
271	VSS	T12
272	GVDD	M22
273	VSS	L13
274	GVDD	N22

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
275	PORT1_VBIAS_UNITYBUFFER	N25
276	PORT1_VBIAS_SF	P23
277	AVDDRU	K22
278	VSS	M13
279	EVDD1	L22
280	PORT1_ARRAY_SHIFT_REG_CLK	N26
281	PORT1_SHIFT_REG_RST	P24
282	PORT1_SHIFT_REG_D	M26
283	PORT1_INPUT_DATA_ACK	N23
284	PORT1_INPUT_DATA_REQ	M25
285	PORT1_INPUT_DATA<23>	N24
286	PORT1_INPUT_DATA<22>	L26
287	PORT1_INPUT_DATA<21>	M24
288	PORT1_INPUT_DATA<20>	L25
289	PORT1_INPUT_DATA<19>	M23
290	PORT1_INPUT_DATA<18>	K26
291	PORT1_INPUT_DATA<17>	L24
292	PORT1_INPUT_DATA<16>	K25
293	VSS	M15
294	VDDR	E22
295	VSS	L11
296	DVDDR	H22
297	PORT1_INPUT_DATA<15>	L23
298	PORT1_INPUT_DATA<14>	J26
299	PORT1_INPUT_DATA<13>	K24
300	PORT1_INPUT_DATA<12>	J25
301	PORT1_INPUT_DATA<11>	K23
302	PORT1_INPUT_DATA<10>	H26

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
303	PORT1_INPUT_DATA<9>	J24
304	PORT1_INPUT_DATA<8>	H25
305	PORT1_INPUT_DATA<7>	J23
306	PORT1_INPUT_DATA<6>	G26
307	PORT1_INPUT_DATA<5>	H24
308	PORT1_INPUT_DATA<4>	G25
309	PORT1_INPUT_DATA<3>	H23
310	PORT1_INPUT_DATA<2>	F26
311	PORT1_INPUT_DATA<1>	G24
312	PORT1_INPUT_DATA<0>	F25
313	PORT1_COL_SHIFT_REG_CLK	G23
314	PORT1_ROW_SHIFT_REG_CLK	E26
315	PORT1_PULSE_GEN_RST	F24
316	PORT1_RESETLFSR	E25
317	PORT1_RESET_IFAT	F23
318	PORT1_ASYNC_RST	D26
319	VDDR	F22
320	VSS	P16
321	PORT1_OUTPUT_DATA<13>	E24
322	PORT1_OUTPUT_DATA<12>	D25
323	PORT1_OUTPUT_DATA<11>	E23
324	VDDR	G22
325	VSS	M11
326	DVDDR	J22
327	VSS	R16
328	PORT1_OUTPUT_DATA<10>	B24
329	PORT1_OUTPUT_DATA<9>	C21
330	PORT1_OUTPUT_DATA<8>	C23

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
331	PORT1_OUTPUT_DATA<7>	D21
332	PORT1_OUTPUT_DATA<6>	D22
333	PORT1_OUTPUT_DATA<5>	E21
334	PORT1_OUTPUT_DATA<4>	B23
335	PORT1_OUTPUT_DATA<3>	C20
336	PORT1_OUTPUT_DATA<2>	A25
337	PORT1_OUTPUT_DATA<1>	D20
338	PORT1_OUTPUT_DATA<0>	A24
339	PORT1_OUTPUT_DATA_REQ	E20
340	PORT1_OUTPUT_DATA_ACK	C22
341	VSS	L14
342	AVDDT	E5
343	VSS	M14
344	PORT1_VTRANSCAP_BODY	C19
345	PORT1_VBP	B22
346	PORT1_VBN	D19
347	PORT1_SYNAPSE_DRIVE_LOW	A23
348	PORT1_VCAPBIAS	E19
349	PORT1_ELEAK	A22
350	PORT1_VTHRESH	C18
351	PORT1_GLEAK_DISTAL	B21
352	PORT1_VSPIKE	D18
353	PORT1_E_REV<1>	A21
354	PORT1_V_TAU<2>	E18
355	PORT1_V_TAU<0>	B20
356	PORT1_VSS_SYNAPSE	C17
357	VSS	N14
358	PORT1_VAMP_LOW	D17

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
359	PORT1_VAMP_HIGH	B19
360	PORT1_VWIDTH_BIAS_LOW	E17
361	PORT1_VWIDTH_BIAS_HIGH	A20
362	PORT1_VINPUT_CURRENT_BIAS	C16
363	PORT1_VWIDTH_BIAS	B18
364	PORT1_V_U_PROBE<1>	D16
365	PORT1_V_U_PROBE<0>	A19
366	PORT1_VMEM_PROXIMAL_PROBE	E16
367	PORT1_VMEM_DISTAL_PROBE	A18
368	PORT1_V_U_PROBE<3>	C15
369	PORT1_V_U_PROBE<2>	B17
370	PORT1_GCOMP	D15
371	PORT1_E_REV<0>	A17
372	PORT1_E_REV<3>	E15
373	PORT1_V_TAU<3>	B16
374	PORT1_V_TAU<1>	C14
375	PORT1_VPDN	A16
376	PORT1_VBIAS	D14
377	PORT1_VRESET	B15
378	PORT1_E_REV<2>	E14
379	PORT1_GLEAK_PROXIMAL	A15
380	PORT1_VPUP	E13
381	PORT1_VPUP_REQ	A14
382	PORT0_VPUP_REQ	D13
383	PORT0_VPUP	B14
384	PORT0_GLEAK_PROXIMAL	C13
385	PORT0_E_REV<2>	B13
386	PORT0_VRESET	E12

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
387	PORT0_VBIAS	A13
388	PORT0_VPDN	D12
389	PORT0_V_TAU<1>	A12
390	PORT0_V_TAU<3>	C12
391	PORT0_E_REV<3>	B12
392	PORT0_E_REV<0>	E11
393	PORT0_GCOMP	A11
394	PORT0_V_U_PROBE<2>	D11
395	PORT0_V_U_PROBE<3>	B11
396	PORT0_VMEM_DISTAL_PROBE	C11
397	PORT0_VMEM_PROXIMAL_PROBE	A10
398	PORT0_V_U_PROBE<0>	E10
399	PORT0_V_U_PROBE<1>	A9
400	PORT0_VWIDTH_BIAS	D10
401	PORT0_VINPUT_CURRENT_BIAS	B10
402	PORT0_VWIDTH_BIAS_HIGH	C10
403	PORT0_VWIDTH_BIAS_LOW	A8
404	PORT0_VAMP_HIGH	E9
405	PORT0_VAMP_LOW	B9
406	VSS	P14
407	PORT0_VSS_SYNAPSE	A7
408	PORT0_V_TAU<0>	D9
409	PORT0_V_TAU<2>	B8
410	PORT0_E_REV<1>	C9
411	PORT0_VSPIKE	B7
412	PORT0_GLEAK_DISTAL	E8
413	PORT0_VTHRESH	A6
414	PORT0_ELEAK	D8

Continued on next page

Table A.4 IFAT Pin Name, Continued.

Number	Pin Name	Pin Number
415	PORT0_VCAPBIAS	B6
416	PORT0_SYNAPSE_DRIVE_LOW	C8
417	PORT0_VBN	A5
418	PORT0_VBP	E7
419	PORT0_VTRANSCAP_BODY	A4
420	VSS	R14
421	AVDDT	F5
422	VSS	T14
423	PORT0_OUTPUT_DATA_ACK	D7
424	PORT0_OUTPUT_DATA_REQ	A3
425	PORT0_OUTPUT_DATA<0>	C7
426	PORT0_OUTPUT_DATA<1>	B5
427	PORT0_OUTPUT_DATA<2>	E6
428	PORT0_OUTPUT_DATA<3>	C5
429	PORT0_OUTPUT_DATA<4>	D6
430	PORT0_OUTPUT_DATA<5>	B4
431	PORT0_OUTPUT_DATA<6>	C6
432	PORT0_OUTPUT_DATA<7>	B25
433	PORT0_OUTPUT_DATA<8>	D5
434	PORT0_OUTPUT_DATA<9>	D2
435	PORT0_OUTPUT_DATA<10>	C1
436	VSS	T16

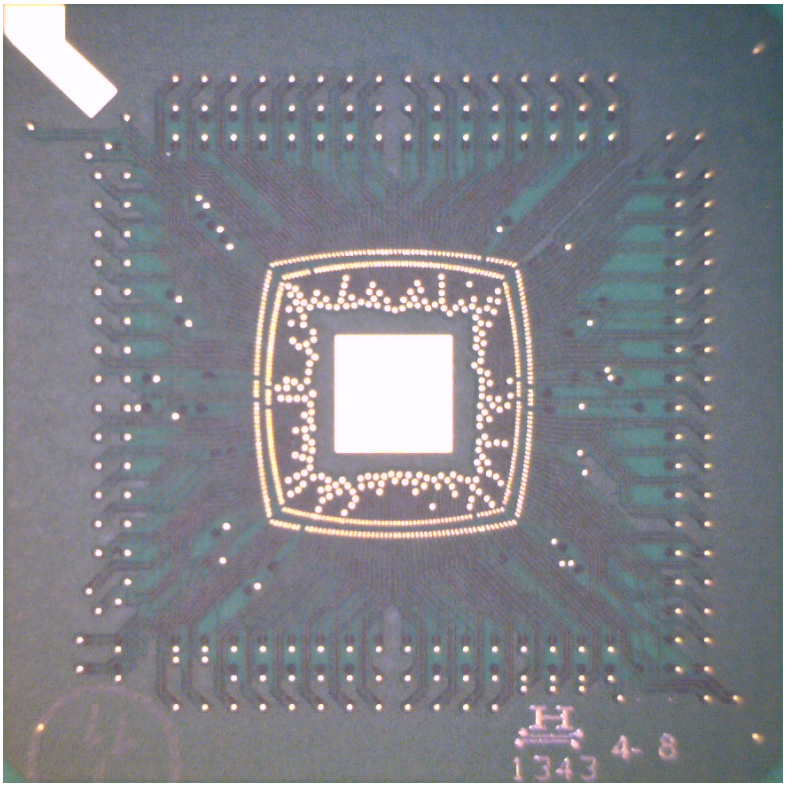


Figure A.1: IFAT packaging substrate

A.3 Substrate Design

Substrate material is Megtron 6 and plating finish in terms of ENEPIG process as shown in Tab. A.5. Designed substrate is, shown in Fig. A.1, fabricated in 35×35 mm^2 4 layers PCB. Packaging ball map is shown in Fig. A.2.

Table A.5: ENEPIG Process

Material	Thickness (μm)
Electroless nickel	100-150
Immersion gold	1-3
Electroless palladium	3-6

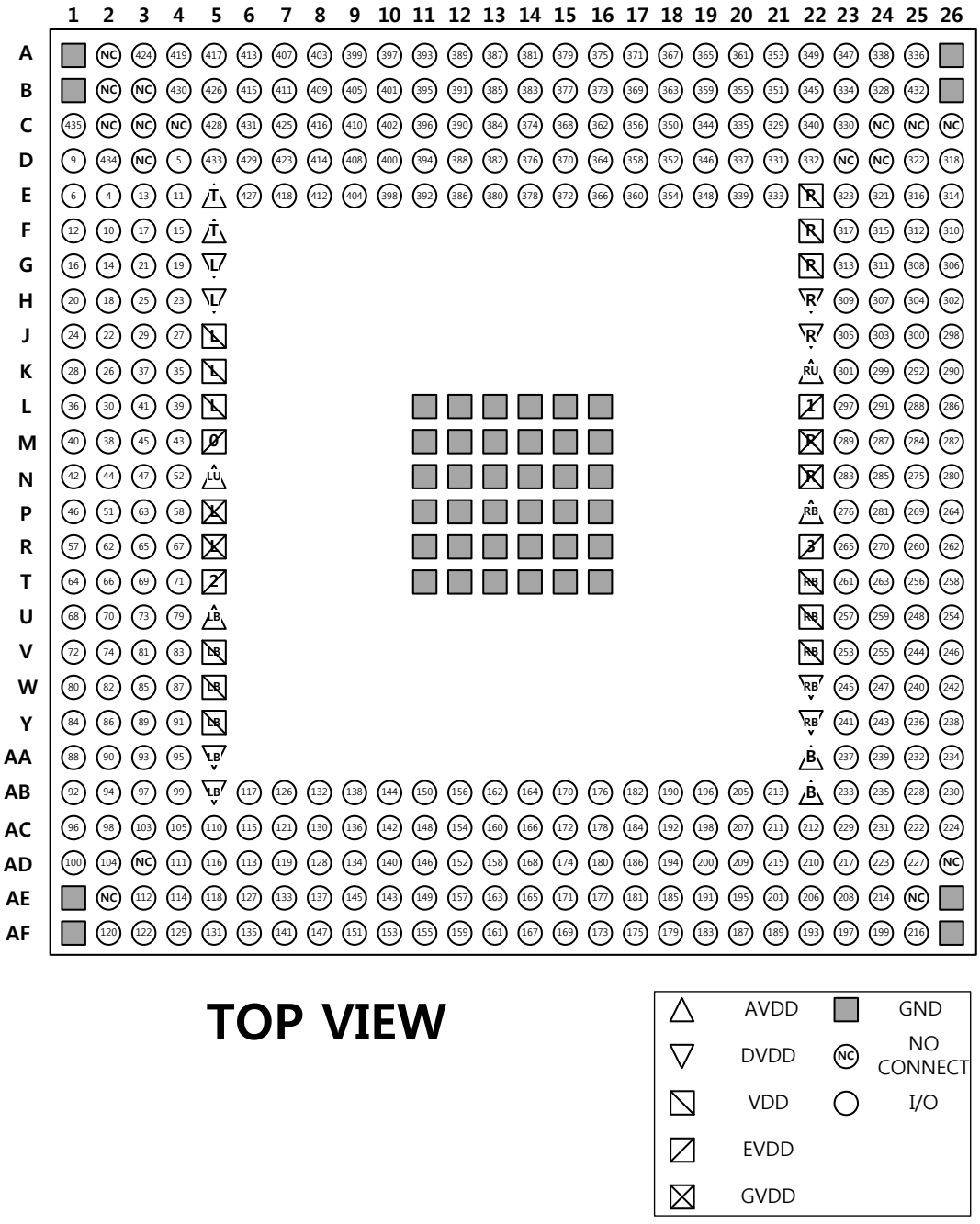


Figure A.2: IFAT pinout diagram.

Appendix B

Imager User Guide

B.1 Imager Pin Definitions

Following tables list the pin definitions used in imager.

B.1.1 Imager Power and Ground

Table B.1: Imager Power and Ground

Pin Name	Direction	Description	Level (V)
VDDIO	Power	Voltage level for driving I/O pad	1.8
VDD INJECT	Power	Voltage level for current injection	1.8
VDD ADC	Power	Voltage level for driving asynchronous ADC	1.8
VDD ARB	Power	Voltage level for arbitration logic	1.8
VDD CORE	Power	Voltage level for driving core logic	1.8
VSSIO	Ground	Ground level for I/O pad	0
VSS	Ground	Ground level for entire chip	0

B.1.2 Imager Digital Input and Output

Table B.2: Imager ADC Digital Input and Output.

Pin Name	Direction	Description	Recommend Initial Value
DATA_IN <14:0>	input	Input address event to access pixel for intensity read-out <13:7> Column, <6:0> Row	-
REQ_IN	input	Handshake request signal	-
ACK_IN	output	Handshake acknowledge signal	-
ADC DATA OUT <9:0>	output	10bit asynchronous ADC output encoded in Gray code	-
SDI_ADC	input	1bit SPI input for serial chain register controlling ADC configuration	-
SCK_ADC	input	Clock for SPI	-
SLOAD ADC	input	Active high SPI enable signal	-
CLOCK GEN INIT	input	Initialize clock generation chain on asynchronous input path	-
PRE_BUF	output	PRE signal generated from clock generation chain. We can probe for debugging purpose.	-
INJECT BUF	output	INJECT signal generated from clock generation chain. We can probe it for debugging purpose.	-

Continued on next page

Table B.2 Imager ADC Digital Input and Output, Continued.

Pin Name	Direction	Description	Recommend Initial Value
ADC ENABLE BUF	output	ADC_ENABLE signal generated from clock generation chain. We can probe for debugging purpose.	-
PROBE	input	Enable folded cascode amplifier for buffering probe signals.	0
OVER VIN	input	Enable over writing VIN node with signal input from EXT_VIN pin.	0
OVER AMP OUT	input	Enable over writing AMP_OUT node with signal input from EXT_AMP_OUT	0
INJECT EXT	input	External INJECT signal for driving ADC externally.	0
CDS PRE EXT	input	External CDS_PRE signal for driving ADC externally.	0
EN_EXT	input	External EN signal for driving ADC externally.	0
LATCH EXT	input	External LATCH signal for driving ADC	0

Table B.3: Imager Temporal Event Digital Input and Output

Pin Name	Direction	Description
DATA <14:0>	output	ON/OFF pixel spike event output encoded in address event <14:8> Row, <7> Sign(0:ON, 1:OFF), <6:0> Column
REQ_IN	output	Handshake request signal
ACK_IN	input	Handshake acknowledge signal
RSTARRAY	input	Active high signal for resetting pixel array
RST_LFSR	input	Active high signal for resetting LFSR
PIXEL RST_B	input	Active high signal for pixel reset
GLOB ADAPT RST_B	input	Active low signal for resetting DC operating voltage of global adaptation node

Table B.4: Imager Bias Calibration SPI

Pin Name	Direction	Description
SDI	input	1bit SPI input for serial chain register controlling bias calibration
SCK	input	Clock for bias calibration SPI register
RESET	input	Reset bias calibration SPI register
DATA_EN	input	Load register value to bias calibration SPI register
SDO	output	Serial chain output from bias calibration SPI register

B.1.3 Imager Analog Input and Output

Table B.5: Imager Analog for Core

Pin Name	Direction	Description	Recommend Initial Value
NBIAS COMP	input	Bias current for comparator in pixel	50n
VOFF_TH	input	Global voltage bias governing the reference voltage implementing off-event threshold	800 mV
VPRE	input	Global voltage bias governing the reference voltage implementing reset level of pixel comparator	900 mV
VON_TH	input	Global voltage bias governing the reference voltage implementing on-event threshold	1 V
PBIAS	input	Bias current for inverting amplifier in pixel	100n
VC_HIGH	input	Voltage bias governing the gate voltage of the cascode bias block (High)	900 mV
VC_LOW	input	Voltage bias governing the gate voltage of the cascode bias block (Low)	700 mV
VC_BIAS	input	Current for injecting current to pixel for CDS	100 nA
PIXEL ANALOG OUTPUT	output	Analog output value of addressed pixel	-
SF_BIAS	input	Bias current for source follower in pixel array	5u

Continued on next page

Table B.5 Imager Analog for Core, Continued.

Pin Name	Direction	Description	Recommend Initial Value
NBIAS HYST	input	Bias current for comparator in clock generation chain implementing clock pulse with hysteresis	100 nA
NBIAS	input	Bias current for comparator in clock generation chain implementing clock pulse	120 nA
CLOCK GEN VREF	input	Voltage reference for comparator in clock generation chain	900 mV
IBIAS PMOS CLK1	output	Current defining duration of PRE signal pulse width with RC time constant comparing CLOCK_GEN_VREF	200 nA
IBIAS PMOS CLK2	output	Current defining duration of INJECT signal pulse width with RC time constant comparing CLOCK_GEN_VREF	200 nA
VBP	input	Global pMOS voltage bias governing the gate voltage of the pMOS in the feedback loop in the row and column arbiters, the buffered outputs from the row and column arbiters, and the pMOS in async input control logic for weak pull up	900 mV
VPUP	input	Global pMOS voltage bias governing the gate voltage of the pMOS at the input of the row and column arbiters	900 mV

Continued on next page

Table B.5 Imager Analog for Core, Continued.

Pin Name	Direction	Description	Recommend Initial Value
VPUPREQ	input	Global pMOS voltage bias governing the gate voltage of the column and row request inputs	900 mV
NBIAS_SF	input	Bias current for source follower in pixel	50n

B.2 Imager Pinout Table

Table B.6: Imager Pin Name

Number	Pin Name	Type
1	NBIAS_COMP	ANALOG
2	VOFF_TH	ANALOG
3	VPRE	ANALOG
4	VON_TH	ANALOG
5	PBIAS	ANALOG
6	VC_HIGH	ANALOG
7	VC_LOW	ANALOG
8	VC_BIAS	ANALOG
9	PIXEL_ANALOG_OUTPUT	ANALOG
10	SF_BIAS	ANALOG
11	NBIAS_HYST	ANALOG
12	NBIAS	ANALOG
13	CLOCK_GEN_VREF	ANALOG
14	IBIAS_PMOS_CLK1	ANALOG
15	IBIAS_PMOS_CLK2	ANALOG

Continued on next page

Table B.6 Imager Pin Name, Continued.

Number	Pin Name	Type
16	VSS	POWER/GND
17	VDD_INJECT	POWER/GND
18	VDD_INJECT	POWER/GND
19	VSSIO	POWER/GND
20	VDDIO	POWER/GND
21	SDI	DIGITAL
22	SCK	DIGITAL
23	RESET	DIGITAL
24	DATA_EN	DIGITAL
25	PIXEL_RST_B	DIGITAL
26	SDO	DIGITAL
27	PRE_BUF	DIGITAL
28	INJECT_BUF	DIGITAL
29	ADC_ENABLE_BUF	DIGITAL
30	ACK_IN	DIGITAL
31	REQ_IN	DIGITAL
32	CLOCK_GEN_INIT	DIGITAL
33	DATA_IN<0>	DIGITAL
34	DATA_IN<1>	DIGITAL
35	DATA_IN<2>	DIGITAL
36	DATA_IN<3>	DIGITAL
37	DATA_IN<4>	DIGITAL
38	DATA_IN<5>	DIGITAL
39	DATA_IN<6>	DIGITAL
40	DATA_IN<7>	DIGITAL
41	DATA_IN<8>	DIGITAL
42	DATA_IN<9>	DIGITAL
43	DATA_IN<10>	DIGITAL

Continued on next page

Table B.6 Imager Pin Name, Continued.

Number	Pin Name	Type
44	DATA_IN<11>	DIGITAL
45	DATA_IN<12>	DIGITAL
46	DATA_IN<13>	DIGITAL
47	VSS	POWER/GND
48	VDD_ADC	POWER/GND
49	VDD_ADC	POWER/GND
50	VSS	POWER/GND
51	VIN_EXT	ANALOG
52	REF_EXT	ANALOG
53	IBIAS_PROBE	ANALOG
54	VCM	ANALOG
55	IBIAS_AMP	ANALOG
56	PBIAS_COMP	ANALOG
57	VHI	ANALOG
58	VLOW	ANALOG
59	EXT_IN	ANALOG
60	EXT_AMP_OUT	ANALOG
61	PROBE_VIN	ANALOG
62	PROBE_AMP_OUT	ANALOG
63	GLOB_ADAPT_OUT	ANALOG
64	GLOB_ADAPT_BIAS	ANALOG
65	GLOB_ADAPT_REF	ANALOG
66	VSSIO	POWER/GND
67	VDDIO	POWER/GND
68	GLOB_ADAPT_RST_B	DIGITAL
69	ADC_DATA_OUT<0>	DIGITAL
70	ADC_DATA_OUT<1>	DIGITAL
71	ADC_DATA_OUT<2>	DIGITAL

Continued on next page

Table B.6 Imager Pin Name, Continued.

Number	Pin Name	Type
72	ADC_DATA_OUT<3>	DIGITAL
73	ADC_DATA_OUT<4>	DIGITAL
74	ADC_DATA_OUT<5>	DIGITAL
75	ADC_DATA_OUT<6>	DIGITAL
76	ADC_DATA_OUT<7>	DIGITAL
77	ADC_DATA_OUT<8>	DIGITAL
78	ADC_DATA_OUT<9>	DIGITAL
79	PROBE	DIGITAL
80	OVER_VIN	DIGITAL
81	OVER_AMP_OUT	DIGITAL
82	SLOAD_ADC	DIGITAL
83	SDI_ADC	DIGITAL
84	SCK_ADC	DIGITAL
85	INJECT_EXT	DIGITAL
86	CDS_PRE_EXT	DIGITAL
87	EN_EXT	DIGITAL
88	LATCH_EXT	DIGITAL
89	VSSIO	POWER/GND
90	VDDIO	POWER/GND
91	DATA<14>	DIGITAL
92	REQ_IN	DIGITAL
93	DATA<13>	DIGITAL
94	DATA<12>	DIGITAL
95	DATA<11>	DIGITAL
96	DATA<10>	DIGITAL
97	DATA<9>	DIGITAL
98	DATA<8>	DIGITAL
99	DATA<7>	DIGITAL

Continued on next page

Table B.6 Imager Pin Name, Continued.

Number	Pin Name	Type
100	DATA<6>	DIGITAL
101	DATA<5>	DIGITAL
102	DATA<4>	DIGITAL
103	DATA<3>	DIGITAL
104	DATA<2>	DIGITAL
105	DATA<1>	DIGITAL
106	DATA<0>	DIGITAL
107	ACK_IN	DIGITAL
108	RSTARRAY	DIGITAL
109	RST_LFSR	DIGITAL
110	VSS	POWER/GND
111	VDD_ARB	POWER/GND
112	VDD_ARB	POWER/GND
113	VSS	POWER/GND
114	VDD_CORE	POWER/GND
115	VDD_CORE	POWER/GND
116	VSS	POWER/GND
117	VBP	ANALOG
118	VPUP	ANALOG
119	VPUPREQ	ANALOG
120	NBIAS_SF	ANALOG

B.3 Chip Packaging

Fig. B.1 shows 5×5 mm^2 imager chip packaged in Thin Quad Flat Pack (TQFP) package with 128 leads, 14×14 mm^2 body and 0.4 mm pitch.

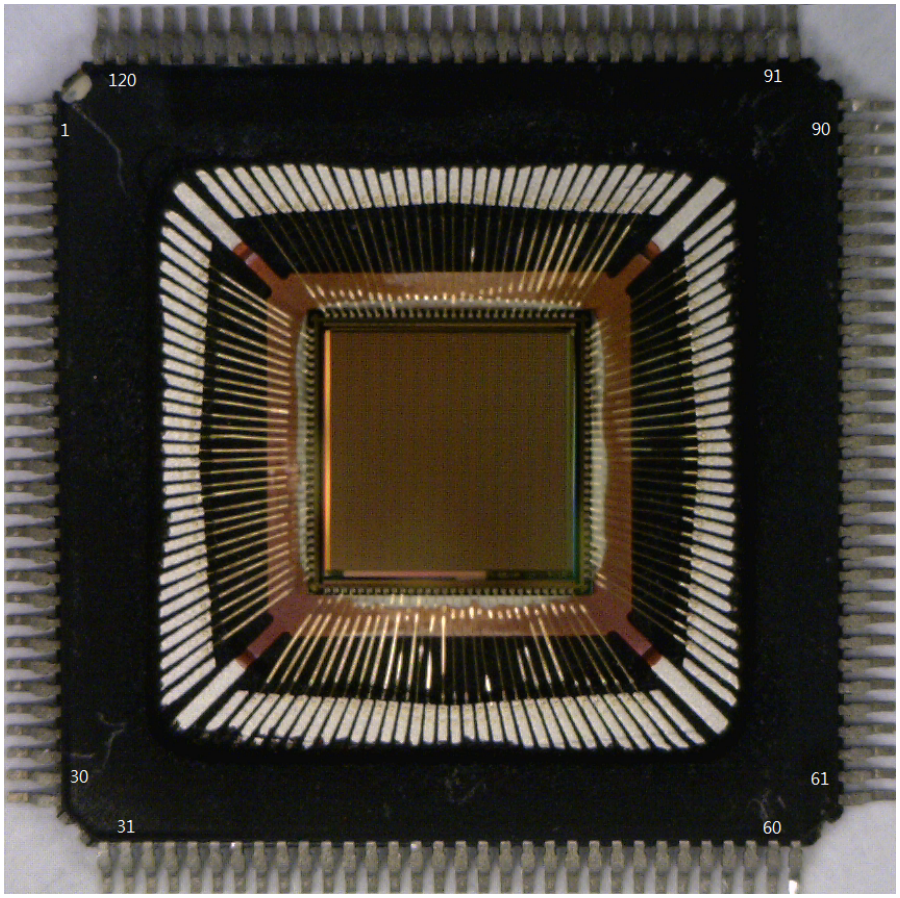


Figure B.1: Imager chip packaging in TQFP 128 leads, 14×14 mm^2 body and 0.4 mm pitch

B.4 Test Board

Fig. B.2 shows test PCB for imager testing.

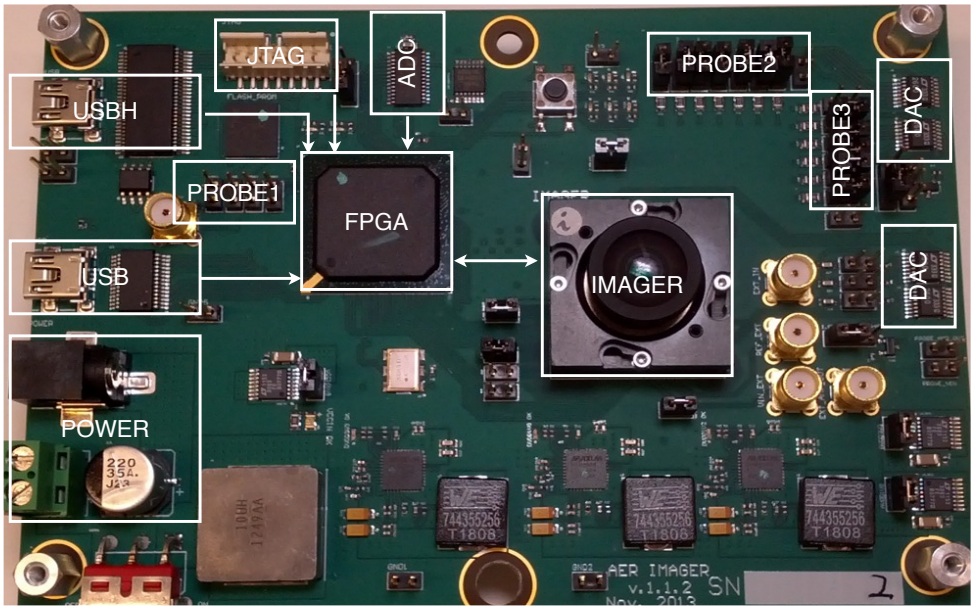


Figure B.2: Imager test board

Bibliography

[1] S. A. Bamford, A. F. Murray, and D. J. Willshaw. Large developing receptive fields using a distributed and locally reprogrammable address-event receiver. *IEEE Transactions on Neural Networks*, 21(2):286–304, 2010.

[2] T. E. Behrens and O. Sporns. Human connectomics. *Current Opinion in Neurobiology*, 22(1):144–153, 2012.

[3] B. Belhadj, A. Joubert, O. Temam, and R. Heliot. Configurable conduction delay circuits for high spiking rates. In *IEEE International Symposium on Circuits and Systems (ISCAS), 2012*, pages 2091–2094, 2012.

[4] B. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. Arthur, P. Merolla, and K. Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[5] H. K. O. Berge and P. Häfliger. High-speed serial AER on FPGA. In *IEEE International Symposium on Circuits and Systems, ISCAS 2007*, pages 857–860, 2007.

[6] R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck. A 240×180 10mW 12μs latency sparse-output vision sensor for mobile applications. In *2013 Symposium on VLSI Circuits*, pages C186–C187, 2013.

[7] K. A. Boahen. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(5):416–434, 2000.

[8] K. A. Boahen and A. G. Andreou. A contrast sensitive silicon retina with reciprocal synapses. In *Advances in Neural Information Processing Systems, NIPS 1991*, volume 4, pages 764–772, 1992.

[9] R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94:3637–3642, 2005.

[10] G. Cauwenberghs. Reverse engineering the cognitive brain. *Proceedings of the National Academy of Sciences*, 110(39):15512–15513, 2013.

[11] V. Chan, C. Jin, and A. van Schaik. An address-event vision sensor for multiple transient object detection. *IEEE Transactions on Biomedical Circuits and Systems*, 1(4):278–288, 2007.

[12] V. Chan, S.-C. Liu, and A. van Schaik. AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(1):48–59, 2007.

[13] Y. M. Chi, R. Etienne-Cummings, and G. Cauwenberghs. Focal-plane change triggered video compression for low-power vision sensor systems. *PLoS ONE*, 4(7):e6384, 2009.

[14] Y. M. Chi, U. Mallik, M. A. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings. CMOS camera with in-pixel temporal change detection and ADC. *IEEE Journal of Solid-State Circuits*, 42(10):2187–2196, 2007.

[15] S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen. *Silicon Neurons That Compute*, volume 7552 of *Lecture Notes in Computer Science*, chapter 16, pages 121–128. Springer Berlin / Heidelberg, 2012.

[16] P. S. Churchland and T. J. Sejnowski. *The Computational Brain*. MIT Press, 1992.

[17] M. DeBole, A. A. Maashri, M. Cotter, C. L. Yu, C. Chakrabarti, and V. Narayanan. A framework for accelerating neuromorphic-vision algorithms on FPGAs. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2011*, pages 810–813, 2011.

[18] S. R. Deiss, R. J. Douglas, and A. M. Whatley. *A pulse-coded communications infrastructure for neuromorphic systems*, chapter 6, pages 157–178. MIT Press, 1999.

[19] C. Eliasmith and C. Adnderson. *Neural engineering: Computatoin, representation, and dynamics in neurobiological systems*. MIT Press, 2004.

[20] D. B. Fasnacht, A. M. Whatley, and G. Indiveri. A serial communication infrastructure for multi-chip address event systems. In *IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pages 648–651, 2008.

[21] J. Fieres, J. Schemmel, and K. Meier. Realizing biological spiking network models in a configurable wafer-scale hardware system. In *IEEE International Joint Conference on Neural Networks, IJCNN 2008*, pages 969–976, 2008.

[22] B. Fischl and A. M. Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences*, 97(20):11050–11055, 2000.

[23] E. Fragnière, A. v. Schaik, and E. Vittoz. Reactive components for pseudo-resistive networks, 1997.

[24] Z. M. Fu, T. Delbrück, P. Lichtsteiner, and E. Culurciello. An address-event fall detector for assisted living applications. *IEEE Transactions on Biomedical Circuits and Systems*, 2(2):88–96, 2008.

[25] S. Furber, D. Lester, L. Plana, J. Garside, E. Painkras, S. Temple, and A. Brown. Overview of the SpiNNaker system architecture. *IEEE Transactions on Computers*, PP(99):1–1, 2012.

[26] S. Fusi and M. Mattia. Collective behavior of networks with linear (VLSI) integrate-and-fire neurons. *Neural Computation*, 11:633–652, 1999.

[27] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou. Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons. *Neural Networks*, 14:781–793, 2001.

[28] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

[29] N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D. S. Modha. A digital neurosynaptic core using event-driven QDI circuits. In *18th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), 2012*, pages 25–32, 2012.

[30] G. Indiveri, B. Linares-Barranco, T. Hamilton, A. v. Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. K. Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5(73), 2011.

[31] G. Indiveri, A. M. Whatley, and J. Kramer. A reconfigurable neuromorphic VLSI multi-chip system applied to visual motion computation. In *Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, 1999.*, pages 37–44, 1999.

[32] E. M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[33] E. M. Izhikevich. Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, 17(10):2443–2452, 2007.

[34] E. M. Izhikevich and F. C. Hoppensteadt. Polychronous wavefront computations. *International Journal of Bifurcation and Chaos*, 19(5):1733–1739, 2009.

[35] T. A. Jarrell, Y. Wang, A. E. Bloniarz, C. A. Brittin, M. Xu, J. N. Thomson, D. G. Albertson, D. H. Hall, and S. W. Emmons. The connectome of a decision-making neural network. *Science*, 337(6093):437–444, 2012.

[36] S. Joshi, S. Deiss, M. Arnold, J. Park, T. Yu, and G. Cauwenberghs. Scalable event routing in hierarchical neural array architecture with global synaptic connectivity. In *12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2010*, pages 1–6, 2010.

[37] U. Kang, H.-J. Chung, S. Heo, D.-H. Park, H. Lee, J.-H. Kim, S.-H. Ahn, S.-H. Cha, J. Ahn, D. Kwon, J.-W. Lee, H.-S. Joo, W.-S. Kim, D. H. Jang, N. S. Kim, J.-H. Choi, T.-G. Chung, J.-H. Yoo, J.-S. Choi, C. Kim, and Y.-H. Jun. 8 Gb 3-D DDR3 DRAM using through-silicon-via technology. *IEEE Journal of Solid-State Circuits*, 45(1):111–119, 2010.

[38] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[39] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras, and S. B. Furber. SpiNNaker: Mapping neural networks onto a massively-parallel chip multiprocessor. In *IEEE International Joint Conference on Neural Networks, IJCNN 2008*, pages 2849–2856, 2008.

[40] D. H. Kim, K. Athikulwongse, M. Healy, M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. H. Woo, X. Zhao, J. Kim, H. Choi, G. Loh, H.-H. Lee, and S.-K. Lim. 3D-MAPS: 3D massively parallel processor with stacked memory. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012*, pages 188–190, 2012.

[41] J. Lazzaro. Temporal adaptation in a silicon auditory nerve. In *Advances in Nueral Inforamtion Processing Systems*, 1992.

[42] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie. Silicon auditory processors as computer peripherals. *IEEE Transactions on Neural Networks*, 4(3):523–528, 1993.

[43] J. A. Lenero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. A 3.6 μ s latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.

[44] P. Lichtsteiner, C. Posch, and T. Delbrück. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[45] J. Lin, P. Merolla, J. Arthur, and K. Boahen. Programmable connections in neuromorphic grids. In *49th IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2006*, volume 1, pages 80–84, 2006.

[46] J. D. Little. A proof for the queuing formula: $l = \lambda w$. *Operations Research*, 9(3):383–387, 1961.

[47] S.-C. Liu and R. Douglas. Temporal coding in a silicon network of integrate-and-fire neurons. *IEEE Transactions on Neural Networks*, 15(5):1305–1314, 2004.

[48] M. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, California Institute of Technology, 1992.

[49] M. Mahowald. *An analog VLSI system for stereoscopic vision*, volume 265. Springer, 1994.

[50] A. Martin and M. Nystrom. Asynchronous techniques for system-on-chip design. *Proceedings of the IEEE*, 94(6):1089–1120, 2006.

[51] A. J. Martin, M. Nystrom, K. Papadantonakis, P. I. Penzes, P. Prakash, C. G. Wong, J. Chang, K. S. Ko, B. Lee, E. Ou, J. Pugh, E. Talvala, J. T. Tong, and A. Tura. The lutonium: a sub-nanojoule asynchronous 8051 microcontroller. In *Proceedings Ninth International Symposium on Asynchronous Circuits and Systems, 2003.*, pages 14–23, 2003.

[52] C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

[53] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha. A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. In *IEEE Custom Integrated Circuits Conference (CICC), 2011*, pages 1–4, 2011.

[54] P. Merolla, J. Arthur, R. Alvarez, J. M. Bussat, and K. Boahen. A multicast tree router for multichip neuromorphic systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(3):820–833, 2014.

[55] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[56] P. A. Merolla, J. V. Arthur, B. E. Shi, and K. A. Boahen. Expandable networks for neuromorphic chips. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(2):301–311, 2007.

[57] S. Mihalas and E. Niebur. A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Computation*, 21:704–718, 2009.

[58] S. Millner, A. Grübl, K. Meier, J. Schemmel, and M.-O. Schwartz. A VLSI implementation of the adaptive exponential integrate-and-fire neuron model. In *Advances in Neural Information Processing Systems, NIPS 2010*, volume 23, pages 1642–1650, 2011.

[59] D. E. Muller and W. S. Bartky. *A theory of asynchronous circuits I.* University of Illinois, Graduate College, Digital Computer Laboratory, 1957.

[60] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. J. Douglas. Synthesizing cognition in neuromorphic electronic systems. *Proceedings of the National Academy of Sciences*, 110(37):E3468–76, 2013.

[61] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs. Event-driven contrastive divergence for spiking neuromorphic systems. *Frontiers Neuroscience*, 7(272), 2014.

[62] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):1943–1953, 2013.

[63] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs. 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver. In *IEEE Biomedical Circuits and Systems Conference (BioCAS), 2014*, page (to appear), 2014.

[64] J. Park, T. Yu, C. Maier, S. Joshi, and G. Cauwenberghs. Live demonstration: Hierarchical address-event routing architecture for reconfigurable large scale neuromorphic systems. In *IEEE International Symposium on Circuits and Systems (ISCAS), 2012*, pages 707–711, 2012.

[65] J. Partzsch and R. Schüffny. Analyzing the scaling of connectivity in neuromorphic hardware and in models of neural networks. *IEEE Transactions on Neural Networks*, 22(6):919–935, 2011.

[66] J. A. Pérez-Carrasco, B. Acha, C. Serrano, L. Camuñas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco. Fast vision through frameless event-based sensing and convolutional processing: Application to texture recognition. *IEEE Transactions on Neural Networks*, 21(4):609–620, 2010.

[67] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier. Stochastic inference with deterministic spiking neurons. *e-print arXiv:1311.3211*, 2013.

[68] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011.

[69] S. Ramakrishnan, R. Wunderlich, and P. Hasler. Neuron array with plastic synapses and programmable dendrites. In *IEEE Biomedical Circuits and Systems Conference (BioCAS), 2012*, pages 400–403, 2012.

[70] J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950, 2010.

[71] S. Scholze, S. Schiefer, J. Partzsch, S. Hartmann, C. G. Mayr, S. Höppner, H. Eisenreich, S. Henker, B. Vogginger, and R. Schüffny. VLSI implementation of a 2.8 Gevent/s packet based AER interface with routing and event sorting functionality. *Frontiers in Neuroscience*, 5, 2011.

[72] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbrück, L. Shih-Chii, R. Douglas, P. Häfliger, G. Jiménez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jiménez, and B. Linares-Barranco. CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory-processing-learning actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20(9):1417–1438, 2009.

[73] T. Serrano-Gotarredona and B. Linares-Barranco. A 128×128 1.5% contrast sensitivity 0.9% FPN 3μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013.

[74] T. Sharp, F. Galluppi, A. Rast, and S. Furber. Power-efficient simulation of detailed cortical microcircuits on SpiNNaker. *Journal of Neuroscience Methods*, 210(1):110–118, 2012.

[75] S. Sheik, E. Chicca, and G. Indiveri. Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2012.

[76] G. Shepherd. *The Synaptic Organization of the Brain*. Oxford University Press, USA, 5th edition, 2003.

[77] M. A. Sivilotti. *Wiring Considerations in Analog VLSI Systems, with Application to Field-Programmable Networks*. PhD thesis, California Institute of Technology, 1991.

[78] D. Sridharan, B. Percival, J. Arthur, and K. A. Boahen. An in-silico neural model of dynamic routing through neuronal coherence. In *Advances in Neural Information Processing Systems, NIPS 2007*, volume 20, pages 1401–1408, 2008.

[79] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber. Power analysis of large-scale, real-time neural networks on spinnaker. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.

[80] R. J. Vogelstein, U. Mallik, J. T. Vogelstein, and G. Cauwenberghs. Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses. *IEEE Transactions on Neural Networks*, 18(1):253–265, 2007.

[81] R. J. Vogelstein, F. Tenore, R. Philipp, M. S. Adlerstein, D. H. Goldberg, and G. Cauwenberghs. Spike timing dependent plasticity in the address domain. In *Advances in Neural Information Processing Systems, NIPS 2002*, volume 15, pages 1171–1178, 2003.

[82] R. Wang, J. Tapson, T. J. Hamilton, and A. van Schaik. An aVLSI programmable axonal delay circuit with spike timing dependent delay adaptation. In *IEEE International Symposium on Circuits and Systems (ISCAS), 2012*, pages 2413–2416, 2012.

[83] T. Yu and G. Cauwenberghs. Log-domain time-multiplexed realization of dynamical conductance-based synapses. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2558–2561, 2010.

[84] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs. 65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing. In *IEEE Biomedical Circuits and Systems Conference (BioCAS), 2012*, pages 21–24, 2012.

[85] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs. Event-driven neural integration and synchronicity in analog VLSI. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2012*, pages 775–778, 2012.

[86] K. A. Zaghloul and K. Boahen. A silicon retina that reproduces signals in the optic nerve. *Journal of Neural Engineering*, 3(4):257–267, 2006.

[87] C. Zamarreño-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco. Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. application to ConvNets. *IEEE Transactions on Biomedical Circuits and Systems*, 7(1):82–102, 2013.

[88] C. Zamarreño-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco. A 0.35 μm sub-ns wake-up time ON-OFF switchable LVDS driver-receiver chip I/O pad pair for rate-dependent power saving in AER bit-serial links. *IEEE Transactions on Biomedical Circuits and Systems*, 6(5):486–497, 2012.