Vetenskapsrådet
Swedish Research Council

Podobas
Research
Group

IzhiRISC-V: a RISC-V Processor with Custom ISA Extension for Spiking Neuron Networks Processing with Izhikevich Neurons

Wiktor Jan Szczerek, Artur Podobas

Division of Software and Computer Science, KTH Royal Institute of Technology, Sweden

Overview

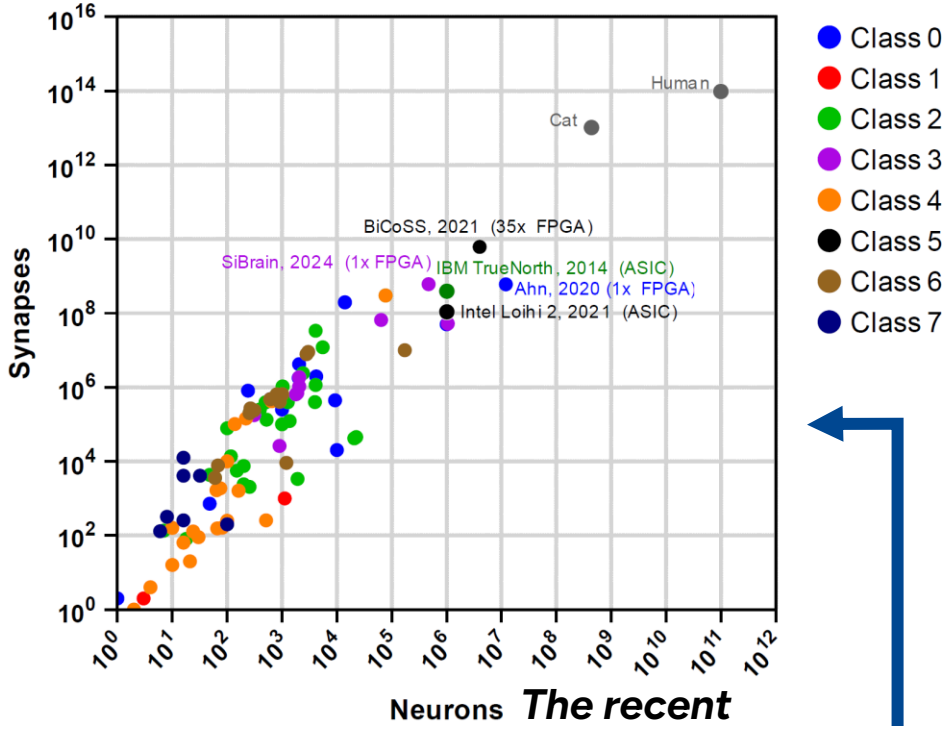
- Motivation
- Background
- Contribution
- Testing & Results
- Future Work & Conclusion

Motivation

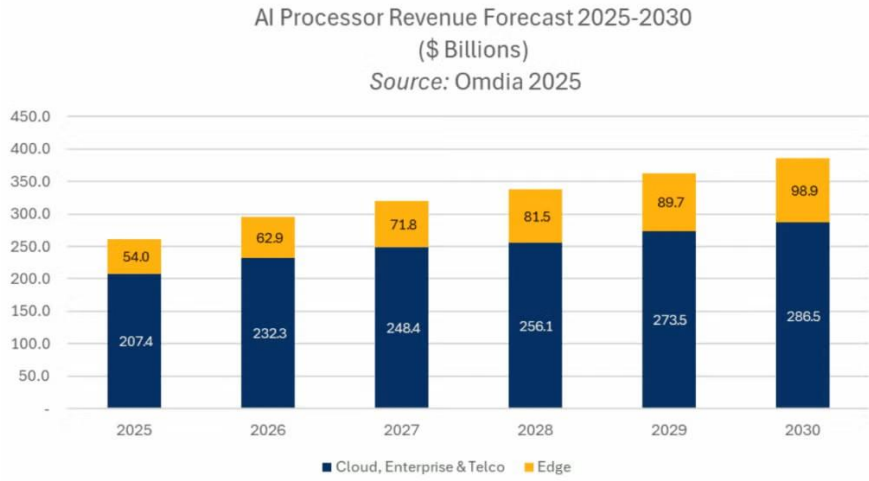
Why do we need an SNN-focused RISC-V?

General idea and motivation

- Neuromorphic systems promise **low power consumption** for both neuroscientific and computer science solutions.
- RISC-V is an open-source, reliable and **highly-expandable ISA** with drastically growing user base – **also in the supercomputing scene**.
- **Creating a custom extension for RISC-V ISA will bring those two worlds closer together!**



The recent advancements in digital neuromorphic architectures bring us closer to realizing the full scale of human brain [1].



AI-capable compute solutions are in high demand – including RISC-V! [2]

Background

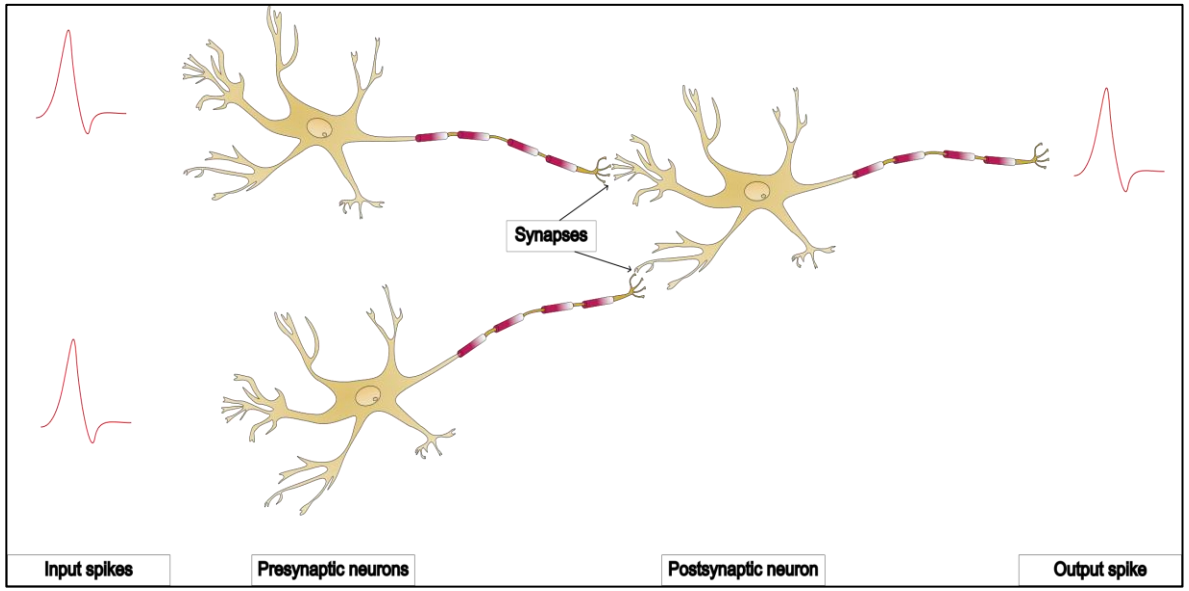
SNNs, neuromorphic systems

SNNs - from biology to digital hardware

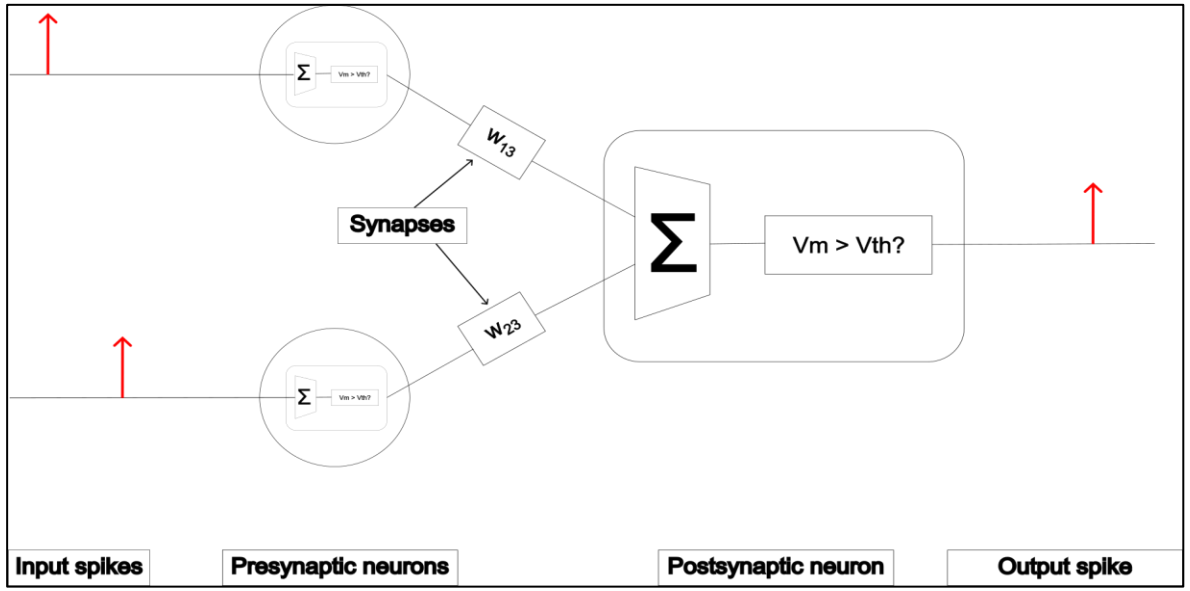
- A **Spiking Neural Network (SNN)** is a third generation of neural networks, bringing them **closer to their biological counterparts.**
- SNNs have been used as tools to **understand the biological brain** [3-5] in the neuroscience community and they are **also used in computer science tasks** [6-8].
- **SNN** are believed **to reduce the energy consumption** of solving the problem and/or solving it faster, primarily by **leveraging sparse spike-based communication.**

SNNs - from biology to digital hardware

Biology



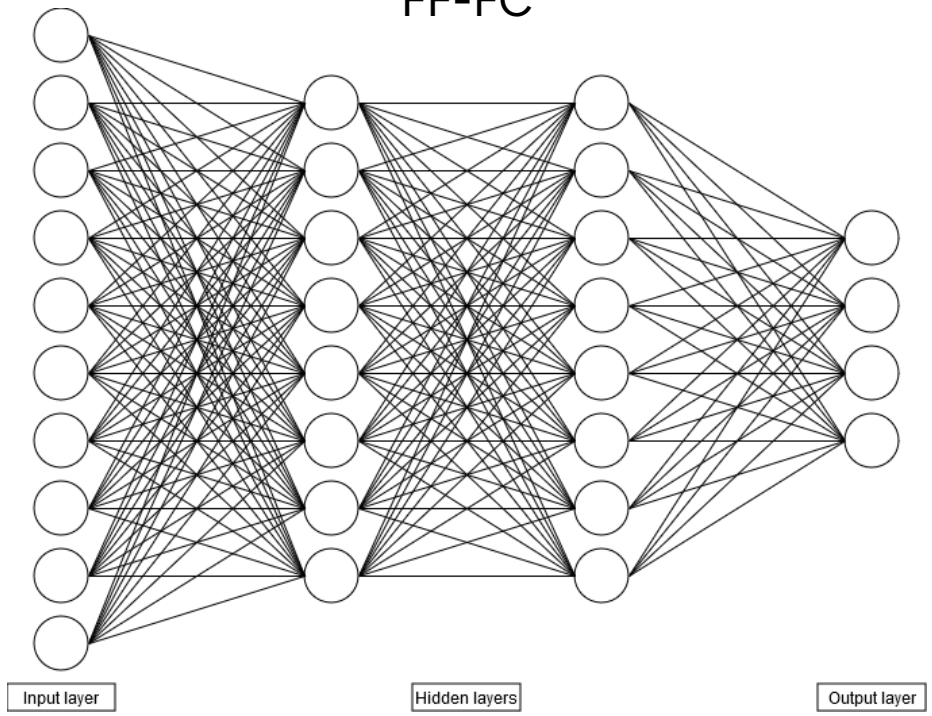
Digital SNN (LIF neuron model)



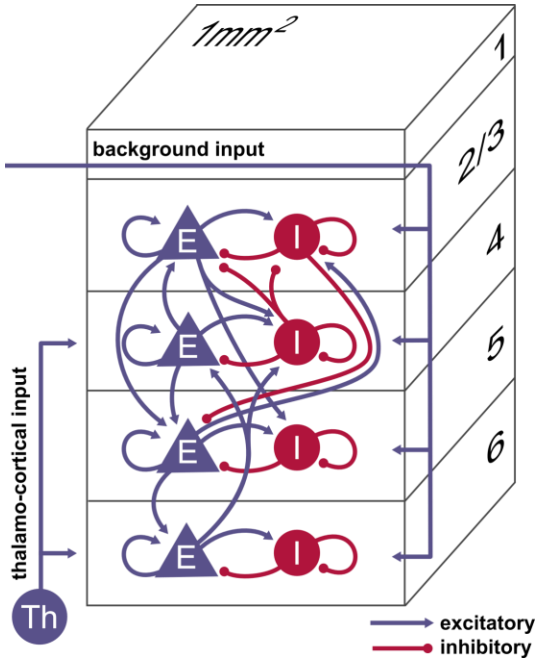
SNNs are way closer to biology than regular ANNs! Here, example with Leaky Integrate and Fire neuron.

Obvious and not-so-obvious SNNs

FF-FC



Cortical microcircuit (image from [9])



Computer Science and Neuroscience have vastly different use cases for them!

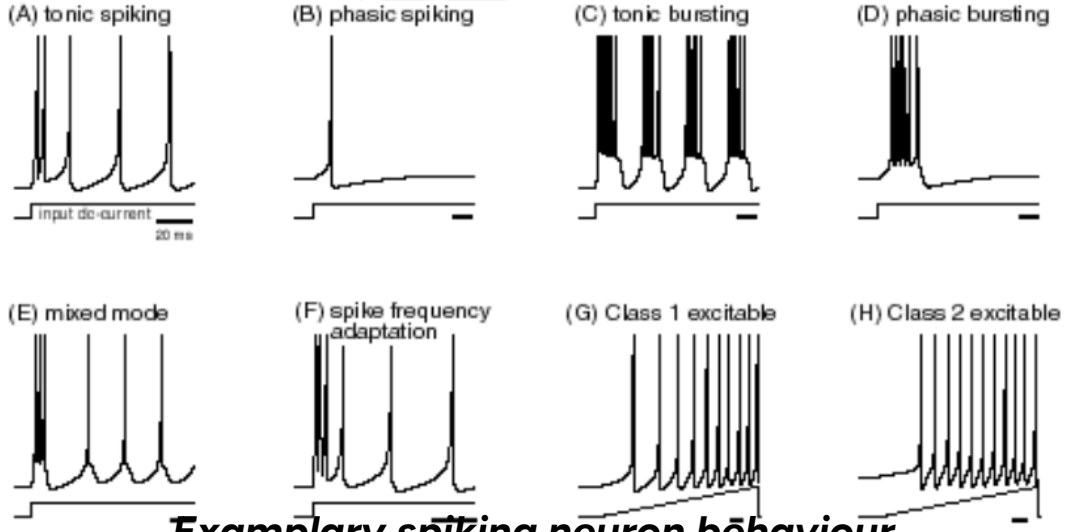
Izhikevich Spiking Neuron

- E. Izhikevich proposed a spiking neural model which was **capable of replicating** many of the dynamics of **Hodgkin-Huxley model** with drastically reduced complexity [9,17].
- The model is often used in both **neuroscience** [10,11] and **computer science-focused** [12,13] applications.
- Its implementation in hardware requires only **19 operations per neuron and synaptic current update** (as opposed to **1200 (!!!)** for Hodgkin-Huxley neuron)

$$\begin{cases} \frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I_{syn} \\ \frac{du}{dt} = a(bv - u) \\ v \geq v_{th} \Rightarrow v \leftarrow c, u \leftarrow u + d \end{cases}$$

$$\frac{I_{syn}}{dt} = -\frac{I_{syn}}{\tau}$$

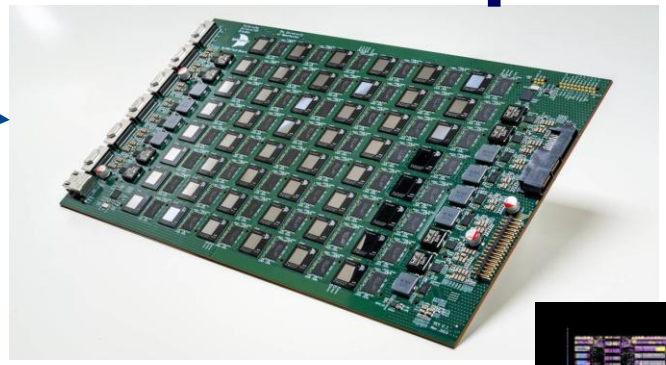
Izhikevich neuron model (4-param.) and AMPA-like current dynamics (exponential decay).

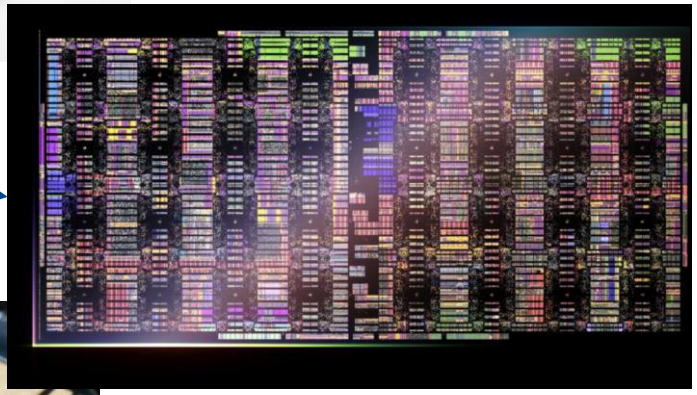


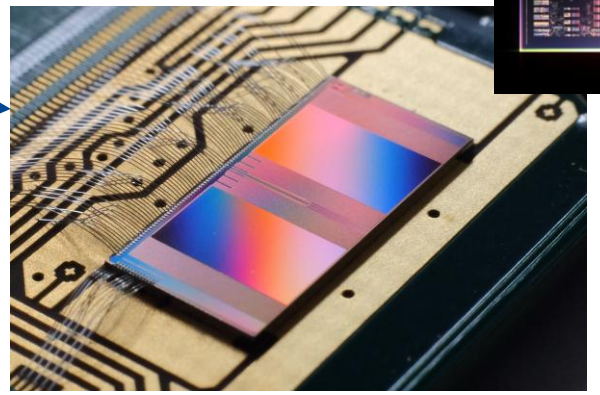
Examplary spiking neuron behaviour representatble by Izhikevich model [13].

Examples of well-known neuromorphic systems

- **SpiNNaker 2 (2021) - digital**
  - **153 ARM cores** with 19MB on-chip SRAM, 2GB DRAM, and dedicated Machine Learning (e.g., MAC) and Neuromorphic (e.g., Exp/Log) accelerators.
- **Intel Loihi 2 (2021) - digital**
  - 6 CPU cores (Lakemont x86) and 128 fully asynchronous neuron cores (NCs) connected by a network-on-chip, single Loihi 2 chip supports up to **1 million** neurons and **120 million synapses**.
- **BrainScaleS 2 (2022) – mixed-signal**
  - single-chip ASIC integrating a custom **analog** core with **512 neuron circuits, 212k plastic synapses**, analog parameter storage, **embedded processors for digital control** and plasticity, and an event routing network



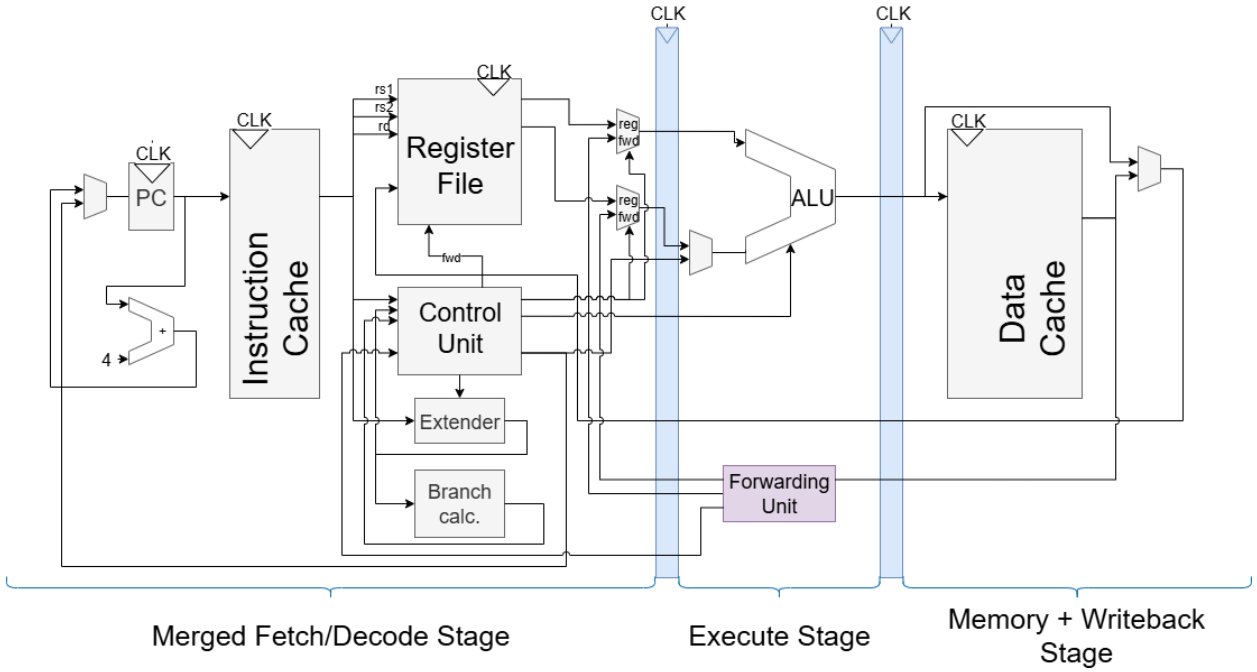




Based on data from [15].

Base RISC-V (DTEK-V) processor

- Our work is based on the **DTEK-V** processor:
  - Compliant with **RV32-IMZICSR** extensions
  - Implemented on Intel MAX10 10M50DAF484C7G
  - Clocked at 30MHz
  - L1 Instruction and Data caches (2kB)
  - 1 IPC (peak)
  - 3-stage pipeline



DTEK-V schematic.

Contribution

ISA extension and programming model for IZH-based SNNs, IzhiRISC-V

Our contributions

- **Extension to the RISC-V** instruction set architecture **(ISA)**, allowing **custom neural instructions** to be included
- Prototype **IzhiRISC-V** architecture with support for the ISA neural extension, allowing single-cycle **Izhikevich ODE integration and AMPA-like current dynamics.**
- We quantify the **resource, frequency, energy-efficiency, and area utilization** of our design on **two** different types of **Field-Programmable Gate Array (FPGAs)** as well as using the **OpenROAD [14]** and mapped against two different standard cell libraries: **FreePDK-45nm** and **ASAP 7nm.**
- We demonstrate **two proof-of-concept applications** running using our IzhiRISC-V system, showing how our extensions can be used to **improve performance** on a **neuroscientific simulation** as well as solving a **constraint satisfaction problem (CSP)** such as Sudoku.

ISA Extension & programming model example

TABLE I: Custom ISA extension - opcode: 0001011

funct3 [14:12]	rs2 [24:20]	rs1 [19:15]	rd [11:7]
$nmldl$ (R)	$31...16 : d$ $(Q4.11)$ $15...0 : c$ $(Q7.8)$	$31...16 : b$ $(Q4.11)$ $15...0 : a$ $(Q4.11)$	$dst : 1 = OK$
$nmldh$ (R)	$reserved$	$31...2 : resv.$ $1 : pin$ $0 : h$	$dst : 1 = OK$
$nmpn$ $("N")$	I_{syn} $(Q15.16)$	$31...16 : v$ $(Q7.8)$ $15...0 : u$ $(Q7.8)$	$src : addr(VU)$ $dst : spike$ $(1 - spike$ $0 - no\ spike)$
$nmdec$ (R)	$\tau\ select$ $(1...9)$	I_{syn} $(Q15.16)$	$dst : dec(I_{syn})$ $(Q15.16)$

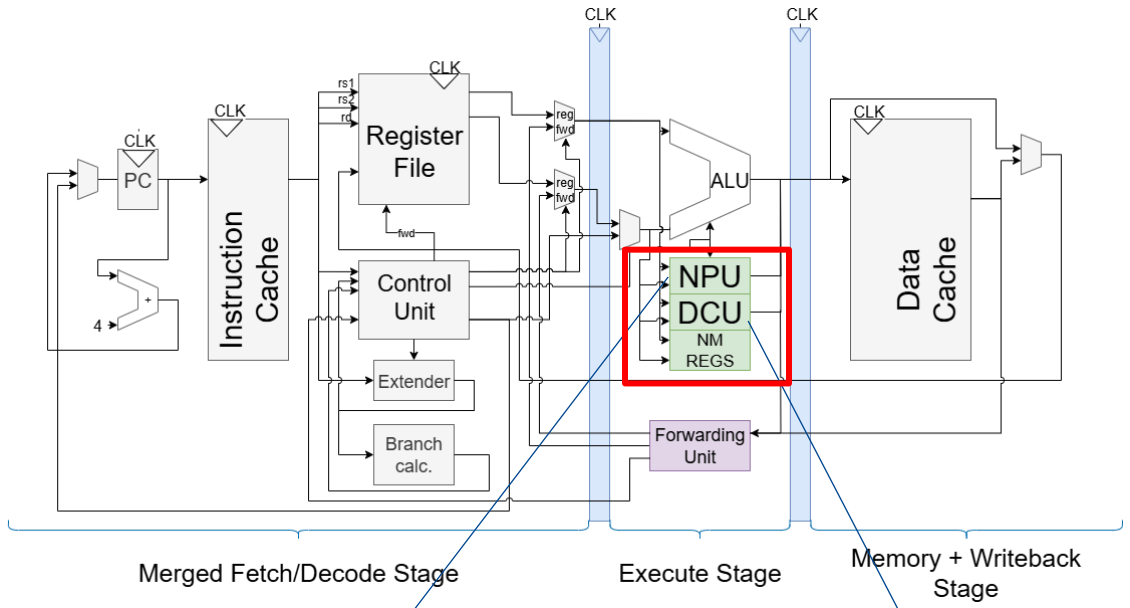
[1]If pin bit is set, NPU caps the neuron voltage at reset potential.
[2]If h bit is set, the NPU operates with timestep of 0.125ms else - 0.5ms.
[3]The $nmpn$ instruction performs Izhikevich neuron update for v and u variables with timestep corresponding to h. The updated neuron state is stored back under the address passed through rd register. The rd register holds the result – information on the neuron spiking in the current timestep.
[4]The $nmdec$ instruction performs exponential decay operation on Q15.16 value with timestep corresponding to h: $y(x) = -(1/2^{\tau})xh$.

```
lw a6, 4(a3)
lw a7, 8(a3)
nmldl x0, a6, a7 # load a,b,c,d
    parameters
lw t5, (a4) # read the thalamic
lw a7, (a0) # read current
lw a6, (a3) # read vu
add a7, a7, t5
add a2, x0, a3
nmpn a2, a6, a7 #process neuron, get
    spike/nospike, store VU word
```

1. Set the neuron parameters (a,b,c,d + decay and timestep)
2. Load neuron state variables
3. (Optionally) Decay the synaptic current
4. Update neuron state and propagate spike

IzhiRISC-V architecture

- Support for custom **ISA extension for Izhikevich spiking neuron** update with **16-bit fixed-point neuron parameters and 32-bit synaptic current representations.**
- Implemented **Neuron Processing Unit (NPU)** and neuron **Decay Unit (DCU)** to realize the IZH ODEs.
- Computation of ODEs for neuron and synaptic current decay in a **single-cycle**, **instead of 19 separate basic operations.**
- Implementation in VHDL, with IEEE's library fixed-point package.



$$\begin{cases} \frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I_{syn} \\ \frac{du}{dt} = a(bv - u) \\ v \geq v_{th} \Rightarrow v \leftarrow c, u \leftarrow u + d \end{cases}$$

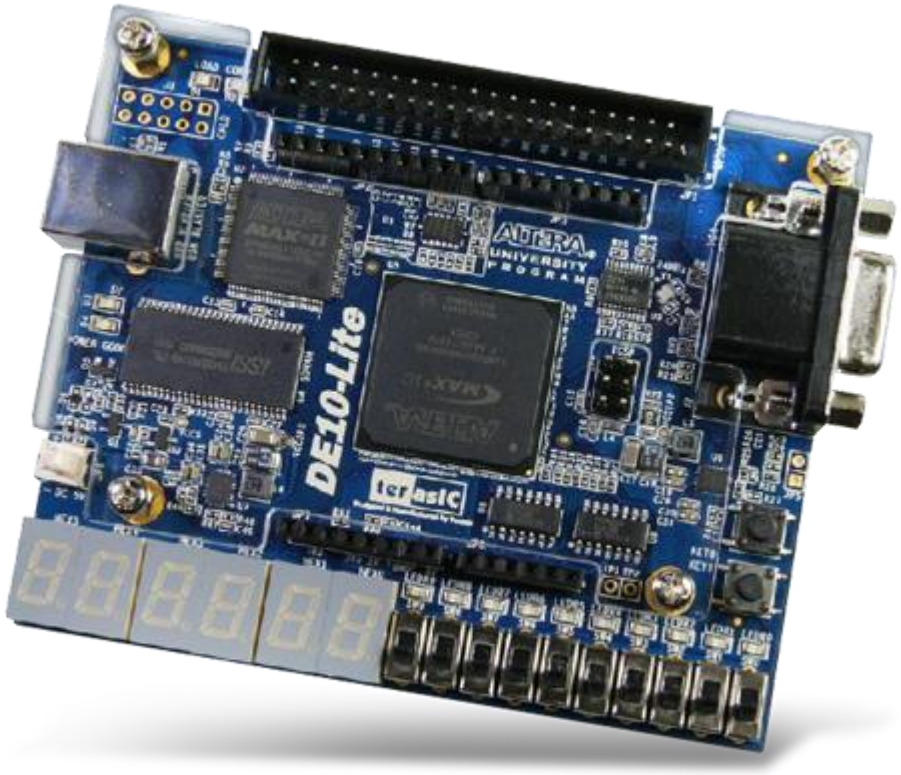
$$\frac{I_{syn}}{dt} = -\frac{I_{syn}}{\tau}$$

Testing & Results

Methodology, performance results, mapping to standard cell library

Testing methodology

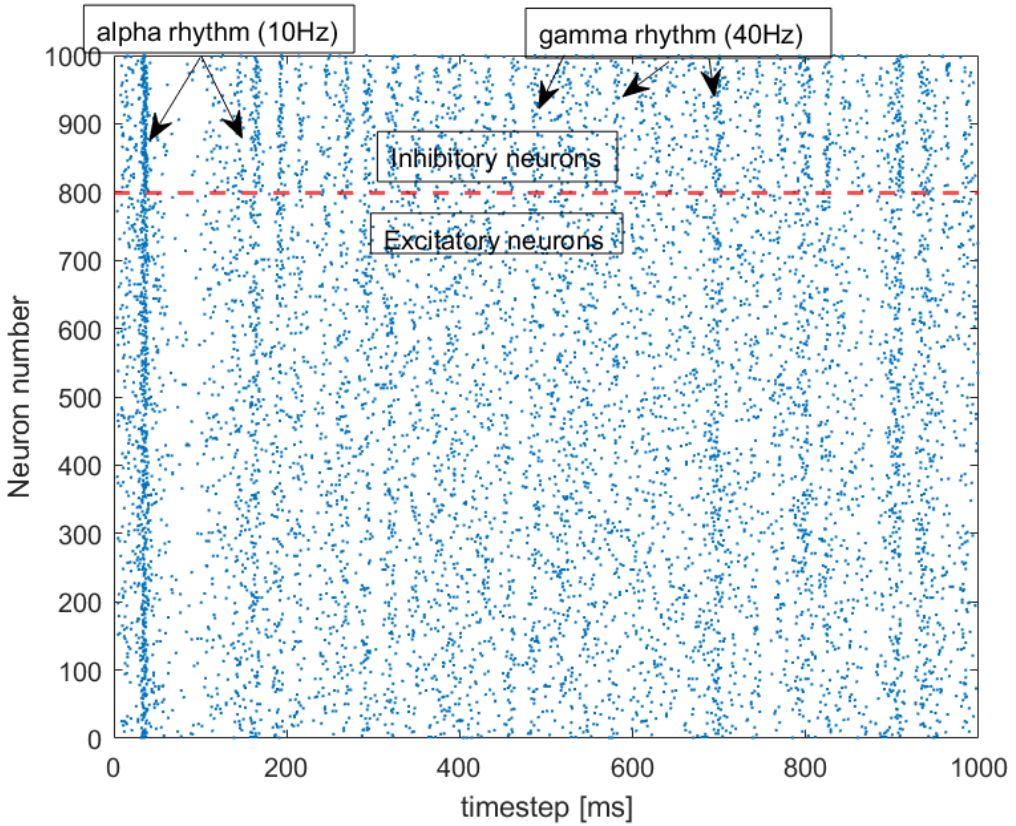
- We define **two main test cases** to verify the IzhiRISC-V's capability to solve neuroscience-focused and CSP problems
  - **Simulating Izhikevich's 80/20 network** showcases expressing biologically-plausible effects in neuron populations
  - **Sudoku solving** showcases ability to solve CSP tasks, which can be translated to other problems in Computer Science
- For each test case we set specific metrics to verify the correctness, as well as measured the achieved performance for **single and dual-core system.**
- We chose **TerasIC DE10-Lite** as the testing platform (Intel MAX10) and performed synthesis for **HPC-grade Agilex-7 system**



TerasIC DE10Lite board. Image courtesy of TerasIC.

Test case #1 – Izhikevich's 80/20 network

- Simulating **1000 IZH** neurons (80% excitatory and 20% inhibitory) with random connections with timestep h = $1ms$. This network was derived from Izhikevich's work [20].
- The correctly simulated network exhibits **alpha/gamma** rhythms in spiking patterns of both excitatory and inhibitory neuron populations – visible in the raster plot to the right.
- Proven by raster plot resemblance to the original work, as well as histograms of Inter-Spike Intervals (ISI).



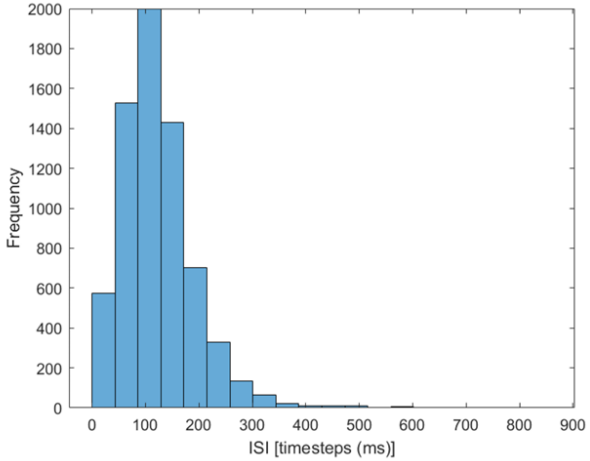
Raster plot of 1000 IZH neurons showcasing the expected spiking behaviour.

Test case #1 - Results

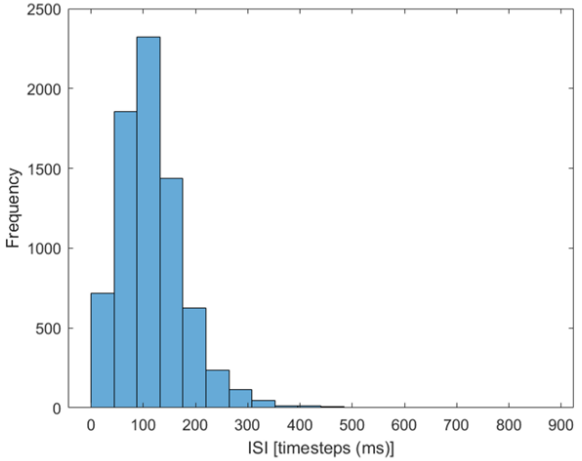
Metric	Single-core	Dual-core	
		Core #1	Core #2
Speedup	1	1.643x	
Execution time [s]	7.870	4.791	4.7906
IPC	0.5735	0.5317	0.51887
IPC_{eff}	0.6516	0.6637	0.6508
Hazard stalls [%]	0.742	5.344	6.259
All cache misses	1306420	639798	675623
I-cache hit rate [%]	99.97	99.97	99.97
D-cache hit rate [%]	96.54	97.18	97.09
Mem intensity	27.15	28.88	30.12

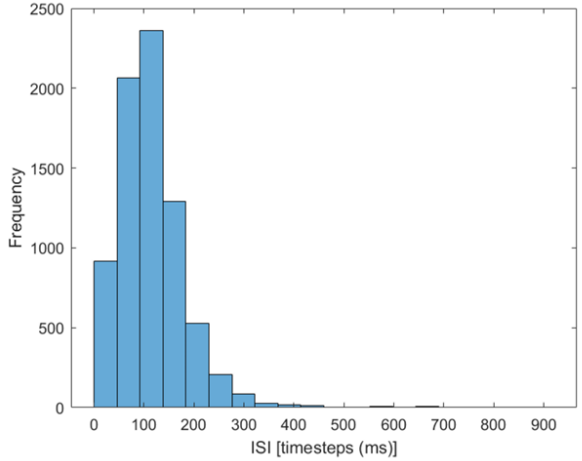
ISI histograms show similar spiking behaviour, and we can see 1.643x speed-up with dual-core system over single-core IzhiRISC-V implementation.

MATLAB (double precision)



MATLAB (fixed-point)



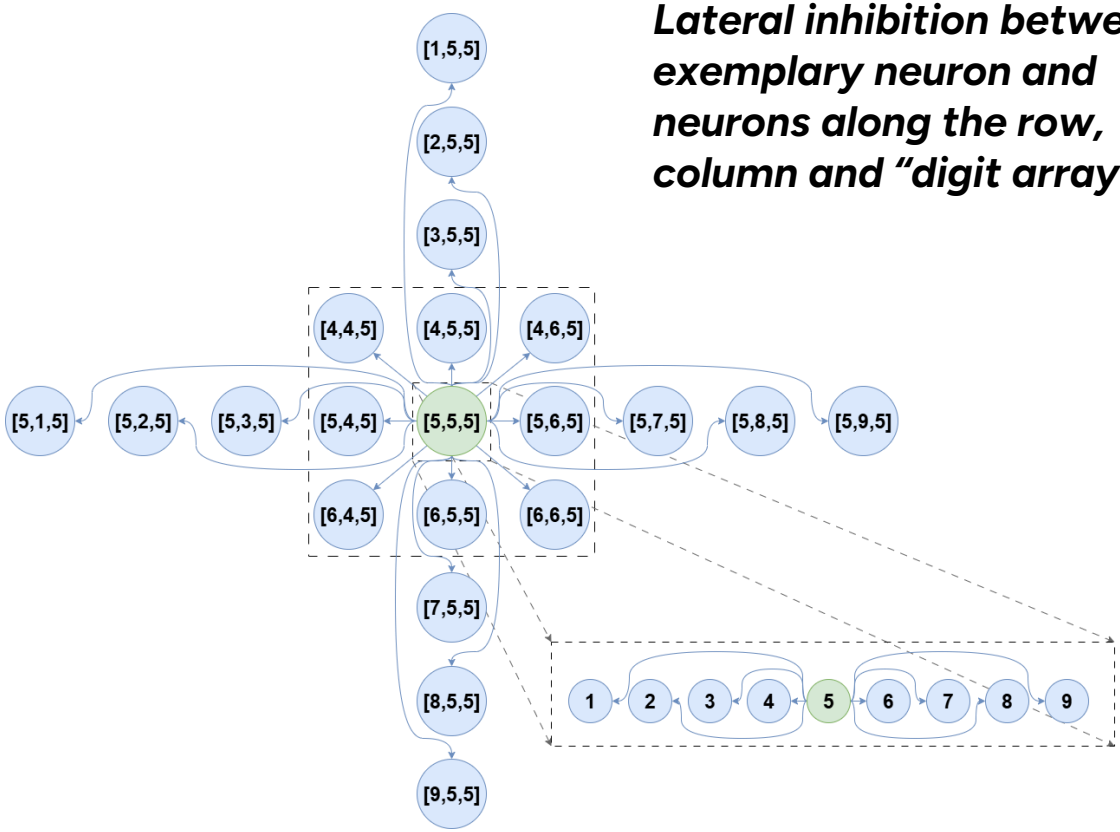


IzhiRISC-V (fixed-point)

Test case #2 – Sudoku solver

- The Sudoku solver use case relied on a network of **729 IZH neurons**, arranged in a particular example of a **Winner-Takes-All (WTA)** network.
- Every cell in a 9x9 Sudoku grid "consists" of an array of nine neurons, every one representing a digit from one to nine.
- Tested with one of many difficult Sudoku puzzle sets [19] – **solving every one of them under 3000 timesteps.**

Metric	Single-core	Dual-core	
		Core #1	Core #2
Speedup	1	1.682x	
Execution time [ms] $(TSTP)$	2.0555	1.2223	1.2223
IPC (AVG)	0.5304	0.4960	0.4194
IPC_{eff} (AVG)	0.7564	0.8635	0.7865
Hazard stalls [%] (AVG)	5.136	6.4793	9.1493
I-cache hit rate [%] (AVG)	98.7230	98.6848	98.8331
D-cache hit rate [%] (AVG)	99.9999	100.0	99.9999
Mem intensity (AVG)	21.3853	22.3176	23.9244



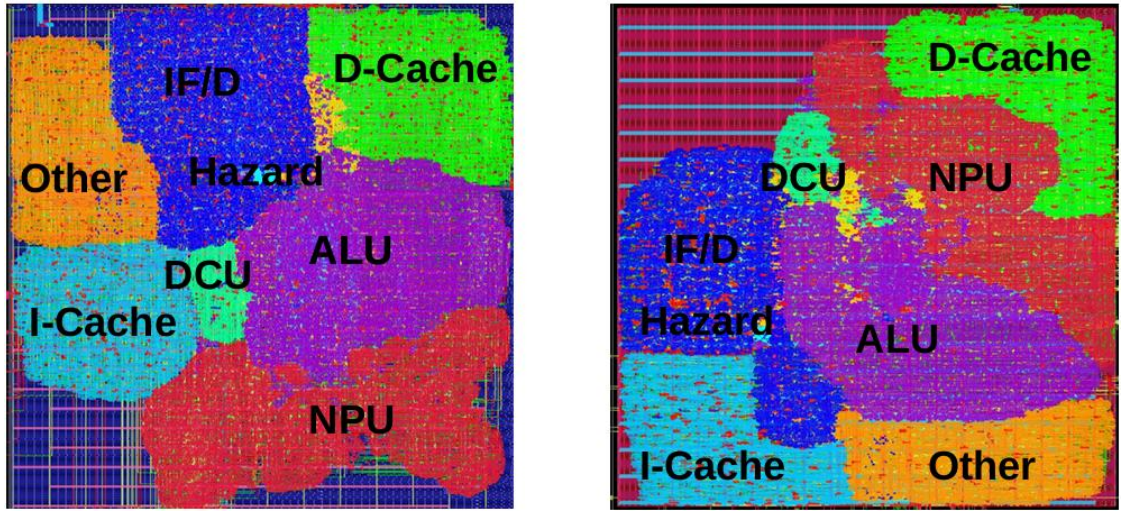
Lateral inhibition between an exemplary neuron and neurons along the row, column and "digit array".

Speed-up of 1.682x for dual-core system over single-core IzhiRISC-V.

Mapping to standard cell library

- We performed mapping to standard cell library to explore hypothetical performance of our design if taped out in both **45nm** and **7nm** processes through OpenROAD framework.

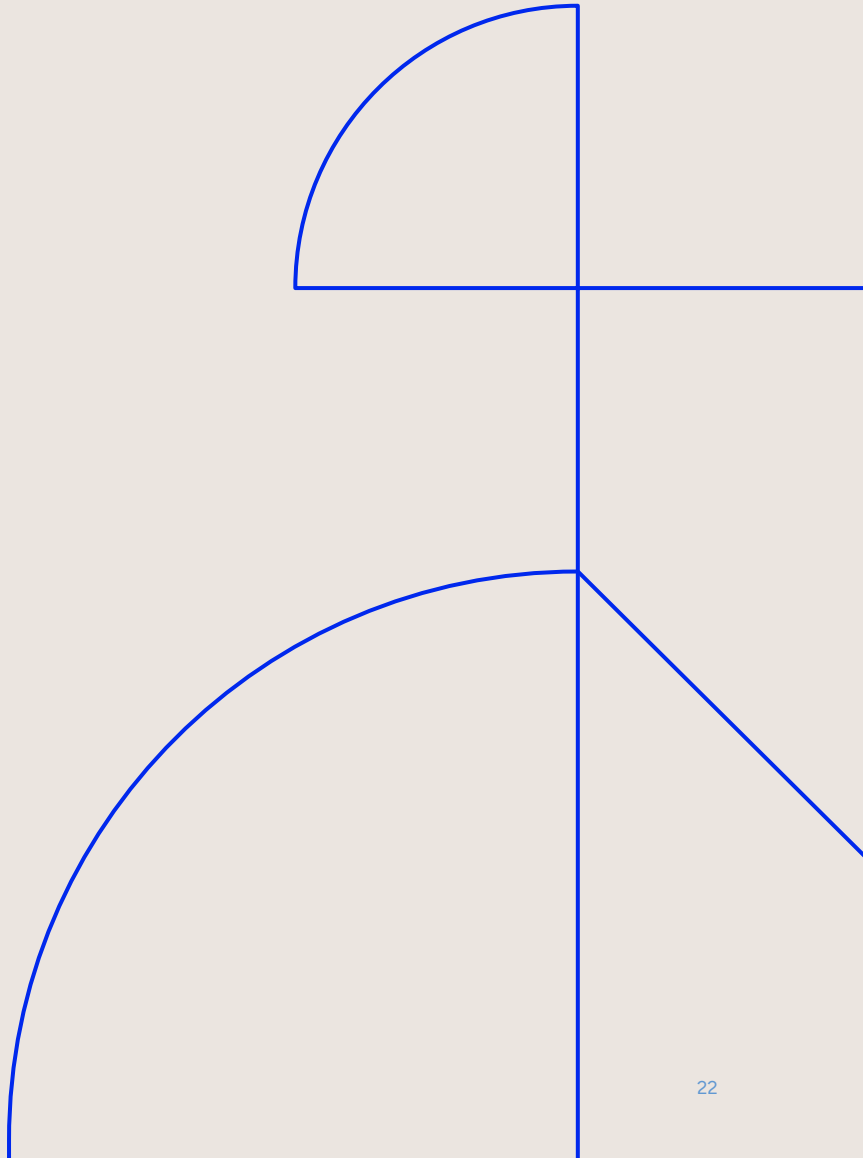
	Metric	FreePDK7	ASAP7	unit
Area	Total area	95654.664	6599.375	μm^2
	Fetch/Decode	16924.250	1116.522	μm^2
	Instruction Cache	10588.662	723.941	μm^2
	Data Cache	12097.414	799.830	μm^2
	Hazard Unit	146.300	7.480	μm^2
	ALU	19873.924	1441.364	μm^2
	NPU	19516.154	1292.196	μm^2
	DCU	2005.640	141.411	μm^2
	Other	11449.172	809.584	μm^2
Power	Total Power	49.5	10.9	mW
	Internal	25.7(51.9%)	6.05(55.5%)	mW
	Switching	21.5(43.5%)	4.85(44.5%)	mW
	Leakage	2.31(4.7%)	6.45(0.1%)	μW
Speed	Clock freq.	201.5	316.3	MHz
	Throughput	67.6	105.4	$MUpd./s$
	Power efficiency:	1.371	9.67	$GUpd/s/W$
	Peak neural IPS[6]:	3.022	4.74	$GInstr./s$



Final floorplan of an IzhiRISC-V core on both the FreePDK 45nm (left) and ASAP 7nm (right) process, showing the NPU and DCU in comparison to other blocks in the pipeline.

***4.74 giga-neuron updates per second and watt**, with **105M neural updates per second** @316.3MHz for 7nm process*

Future plans & Conclusion

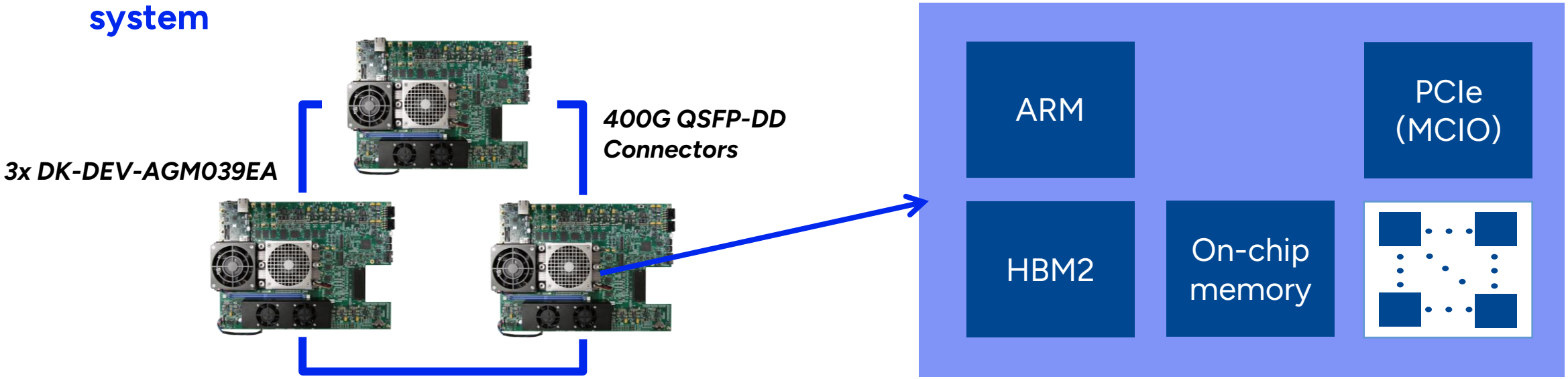


Future plans

- Performance improvements to the core
  - Increasing the pipeline stages to 6-7
  - **Interrupt-based spike issuing**
  - Increasing the operating frequency
- **Many-core (500+) system on 3x Agilex-7 system**

Metric	16 cores	32 cores	64 cores
Frequency	100 MHz		
ALM	107144 (8%)	216448 (17%)	420977 (32%)
FF	95624 (2%)	186760 (4%)	372741 (7%)
RAM blocks	390 (2%)	646 (3%)	1158 (6%)
DSP	152 (1%)	304 (2%)	608 (5%)

Synthesis results for many-core IzhiRISC-V system on a single Agilex-7 device.



Conclusion

- We proposed a **neuromorphic ISA extension** for RISC-V processor with focus on Izhikevich neuron computations.
- We proposed **IzhiRISC-V** – a prototype architecture supporting the aforementioned ISA extension and proved the validity of the implementation in both neuroscientific and CSP tasks.
- We built a foundation for our future work in **bringing SNNs to the RISC-V** and **IzhiRISC-V to HPC space** through HPC-grade-FPGA-based systems.

Vetenskapsrådet
Swedish Research Council

Podobas
Research
Group

Thank you greatly for you attention!

Wiktor Jan Szczerek, Artur Podobas

Division of Software and Computer Science, KTH Royal Institute of Technology, Sweden

References

1. **Szczerek, Wiktor J., and Artur Podobas. "A Quarter of a Century of Neuromorphic Architectures on FPGAs--an Overview." *arXiv preprint arXiv:2502.20415* (2025).**
2. Omdia, AI Processors for Cloud and Data Center Forecast Report – 2025 Analysis, Oct 2025.
3. S. Y. Bonabi, H. Asgharian, R. Bakhtiari, S. Safari, and M. N. Ahmadabadi, "FPGA implementation of a cortical network based on the Hodgkin-Huxley neuron model," vol. 7663 LNCS. pp. 243–250, 2012. doi: 10.1007/978-3-642-34475-6_30.
4. A. Heittmann *et al.*, "Simulating the Cortical Microcircuit Significantly Faster Than Real Time on the IBM INC-3000 Neural Supercomputer," *Frontiers in Neuroscience*, vol. 15. 2022. doi: 10.3389/fnins.2021.728460.
5. T. C. Potjans and M. Diesmann, "The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model," *Cerebral cortex*, vol. 24, no. 3, pp. 785–806, 2014.
6. E. Z. Farsa, A. Ahmadi, M. A. Maleki, M. Gholami, and H. N. Rad, "A Low-Cost High-Speed Neuromorphic Hardware Based on Spiking Neural Network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 9. pp. 1582–1586, 2019. doi: 10.1109/TCSII.2019.2890846.
7. A. Carpegna, A. Savino, and S. Di Carlo, "Spiker+: a framework for the generation of efficient Spiking Neural Networks FPGA accelerators for inference at the edge," *arXiv preprint arXiv:2401.01141*, 2024.
8. S. Yang *et al.*, "BiCoSS: Toward Large-Scale Cognition Brain With Multigranular Neuromorphic Architecture," *IEEE Transactions on Neural Networks and Learning Systems*. 2021. doi: 10.1109/TNNLS.2020.3045492.
9. Izhikevich, Eugene M. "Which model to use for cortical spiking neurons?." IEEE transactions on neural networks 15.5 (2004): 1063-1070.
10. M. Ambroise, T. Levi, Y. Bornat, and S. Saighi, "Biorealistic spiking neural network on FPGA," in 2013 47th Annual Conference on Information Sciences and Systems (CISS), 2013, pp. 1–6. doi: 10.1109/CISS.2013.6616689.
11. K. Kauth, T. Stadtmann, V. Sobhani, and T. Gemmeke, "neuroAIx-Framework: design of future neuroscience simulation systems exhibiting execution of the cortical microcircuit model 20× faster than biological real-time," Frontiers in Computational Neuroscience, vol. 17, 2023, doi: 10.3389/fncom.2023.1144143.
12. K. Wang, J. Wang, X. Hao, B. Deng, Z. Zhang, and G. Yi, "Large-Scale Bio-Inspired FPGA Models for Path Planning," IEEE Transactions on Biomedical Circuits and Systems, 2023.
13. Y. Liu, Y. Chen, W. Ye, and Y. Gui, "FPGA-NHAP: A General FPGA-Based Neuromorphic Hardware Acceleration Platform With High Speed and Low Power," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 6, pp. 2553–2566, Jun. 2022, doi: 10.1109/TCSI.2022.3160693.
14. Ajayi, Tutu, and David Blaauw. "OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain." Proceedings of Government Microcircuit Applications and Critical Technology Conference. 2019.
15. https://open-neuromorphic.org/
16. http://magictour.free.fr/top100
17. Izhikevich, Eugene M. "Simple model of spiking neurons." IEEE Transactions on neural networks 14.6 (2003): 1569-1572.

Additional slides

Spiking neuron models

Neuron Models			
Name	Formula	Bio. plausibility	Complexity
Integrate-and-Fire (IF)	$\frac{dv}{dt} = RI$	V. Low	5 FLOPs
Leaky IF (LIF)	$\frac{dv}{dt} = -(v - V_{reset}) + RI$	Low	10 FLOPs
Izhikevich (IZH)	$\begin{cases} \frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \\ \frac{du}{dt} = a(bv - u) \end{cases}$	Low	60 FLOPs
FitzHugh-Nagumo (FHN)	$\begin{cases} \frac{dv}{dt} = v - \frac{v^3}{3} - w + I \\ \frac{dw}{dt} = 0.08(v + 0.7 - 0.8w) \end{cases}$	Medium	100 FLOPs
Adaptive IF (AdEX)	$\begin{cases} \frac{dv}{dt} = \frac{-g_L(v - E_L) + g_L\Delta_T exp(\frac{v - v_T}{\Delta_T}) - w + I}{C} \\ \frac{dw}{dt} = a(v - E_L) - w \end{cases}$	Medium	200 FLOPs
Hodgkin-Huxley (HH)	$\begin{cases} \frac{du}{dt} = \frac{-g_K n^4 (U - V_K) - g_{Na} m^3 h (U - V_{Na}) - g_{leak}(U - u_{leak}) + I}{C_m} \\ \frac{dn}{dt} = \alpha_n(U)(1 - n) - \beta_n(U)n \\ \frac{dm}{dt} = \alpha_m(U)(1 - m) - \beta_m(U)m \\ \frac{dh}{dt} = \alpha_h(U)(1 - h) - \beta_h(U)h \end{cases}$	High	1200 FLOPs