# A System-on-Chip Based Hybrid Neuromorphic Compute (HNC) Node Architecture for Reproducible Hyper-Real-Time Simulations of Spiking Neural Networks

**Guido Trensch[1,3] and Abigail Morrison[1,2,3]**

[1]Simulation and Data Laboratory Neuroscience, [2]Institute of Neuroscience and Medicine (INM-6), JARA-Institute Brain Structure-Function Relationship (JBI-1 / INM-10), Research Centre Jülich, [3]Department of Computer Science 3 - Software Engineering, RWTH Aachen University

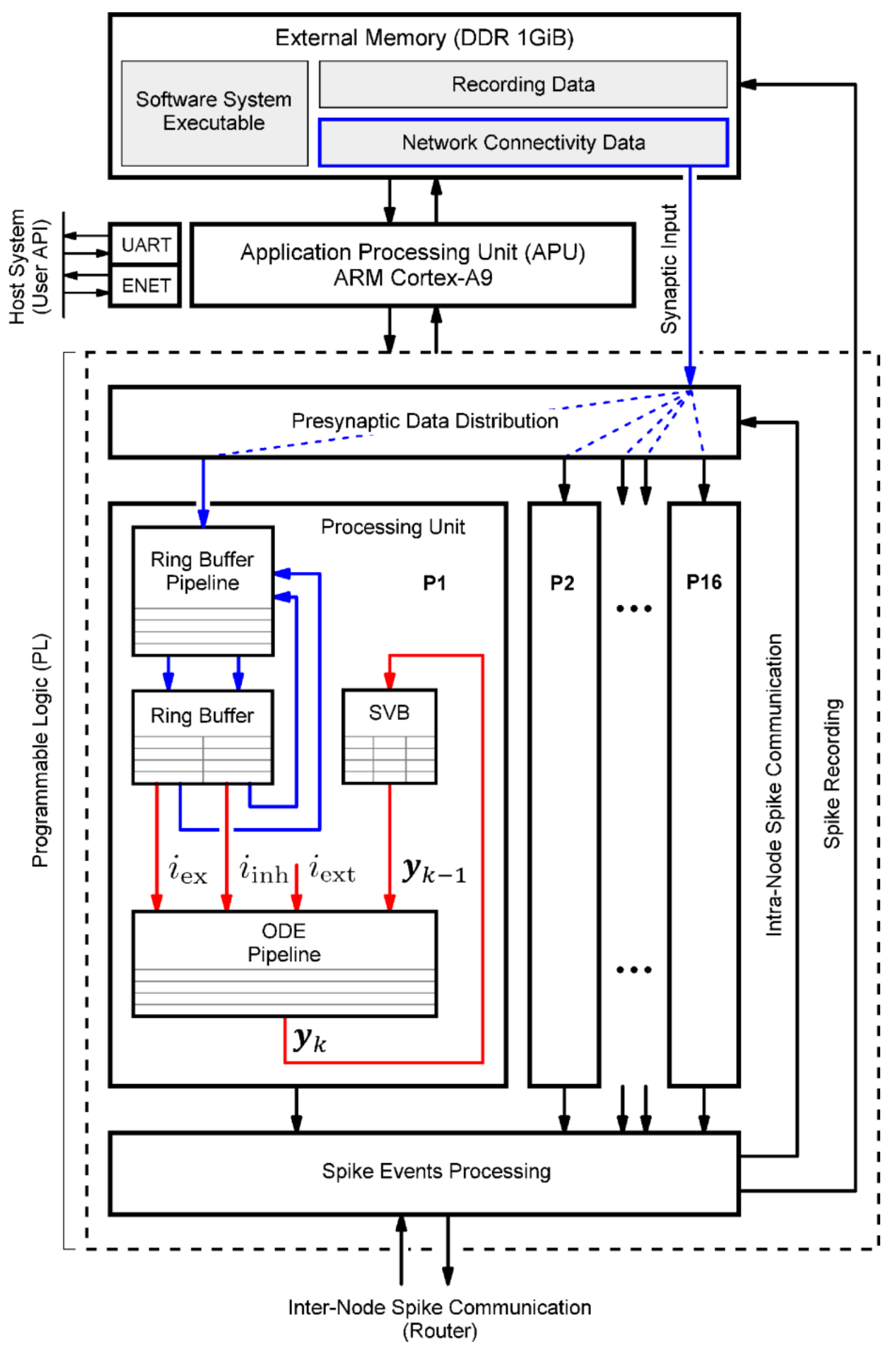
https://doi.org/10.3389/fninf.2022.884033

## Goal

Benefiting from the continued advances in semiconductor technology, in recent years, programmable device technology and tools have greatly increased.

**Proof of Concept:** Prototypical implementation of an AMD Xilinx System-on-Chip (SoC) based hybrid software-hardware architecture approach for a neuromorphic compute node capable of meeting the high demands for modeling and simulation in neuroscience.

## HNC Node High-Level Architecture



### Simulation paradigm

- Hybrid strategy for time-discrete neural network simulations of point neuron models:
  - time-driven neuron state update (blue data paths) at fixed intervals, $\Delta t = 0.1ms$
  - event-driven synapse update (red data paths)

### Performance

- Exploiting the tight coupling of an Application Processing Unit (APU) with a Field Programmable Gate Array (FPGA) located on the same chip.
- Off-loading of performance critical algorithms to programmable logic.
- Parallelization by distributing the computational load over 16 processing units (P1, P2, …, P16).
- Data locality of state variables by storing them in fast on-chip block RAM memories (BRAMs).
- Latency hiding by exploiting the true-dual port capability of BRAMs.

### Flexibility

- ODE pipeline module can be replaced to implement different neuron and synapse models.
- Flexibility in the choice of data types.
- Connectivity data is stored in external memory, thus synaptic weights are adjustable and accessible to the APU.
- Non-critical tasks are executed in software by the APU.

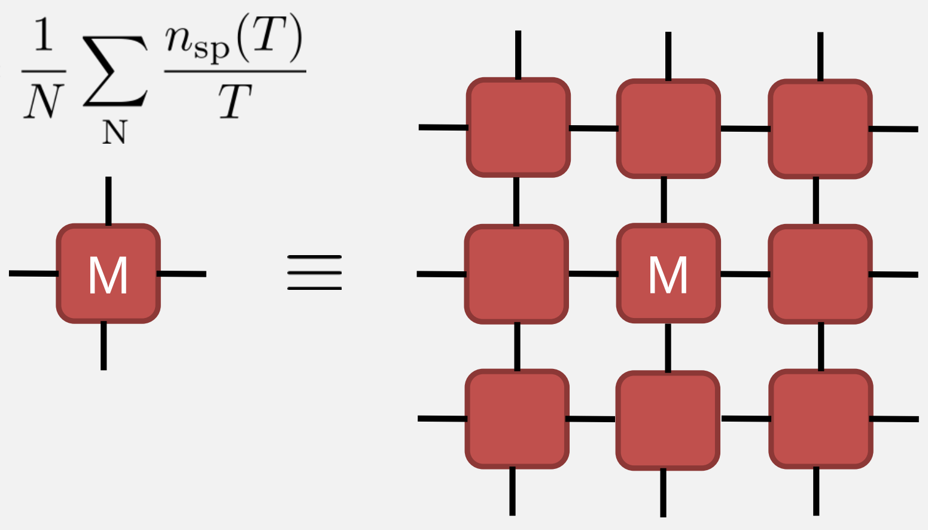
## Workload Model

**From the workload perspective, it can be considered equivalent.**

**small** network<br>
**high** average firing rate

**large** network<br>
**low** average firing rate

$$\bar{\nu} = \frac{1}{N}\sum_{\mathrm{N}} \frac{n_{\mathrm{sp}}(T)}{T}$$



**Indicator for the computational workload: average number of spikes per time step *k***

$$\bar{\nu}_{\mathrm{k}} = \frac{h}{T}\sum_{\mathrm{N}} n_{\mathrm{sp}}(T) = N\bar{\nu}h$$

| Parameter | | HNC Node |
|---|---|---|
| number of compute nodes | $M$ | N/A |
| total number of neurons | $N$ | N/A |
| number of neurons per compute node | $N^{\mathrm{M}}$ | max. 1024 |
| connection probability | $\epsilon$ | < 0.1 |
| presynaptic neuron’s max. number of connections per node | $C^{\mathrm{M}} = \epsilon N^{\mathrm{M}} = \dfrac{\epsilon N}{M}$ | max.128 |
| spike count of neuron $n$ in interval $T$ | $n_{\mathrm{sp}}(T)$ | N/A |
| temporal resolution of the simulation | $h$ | 0.1 ms |

## Performance Model

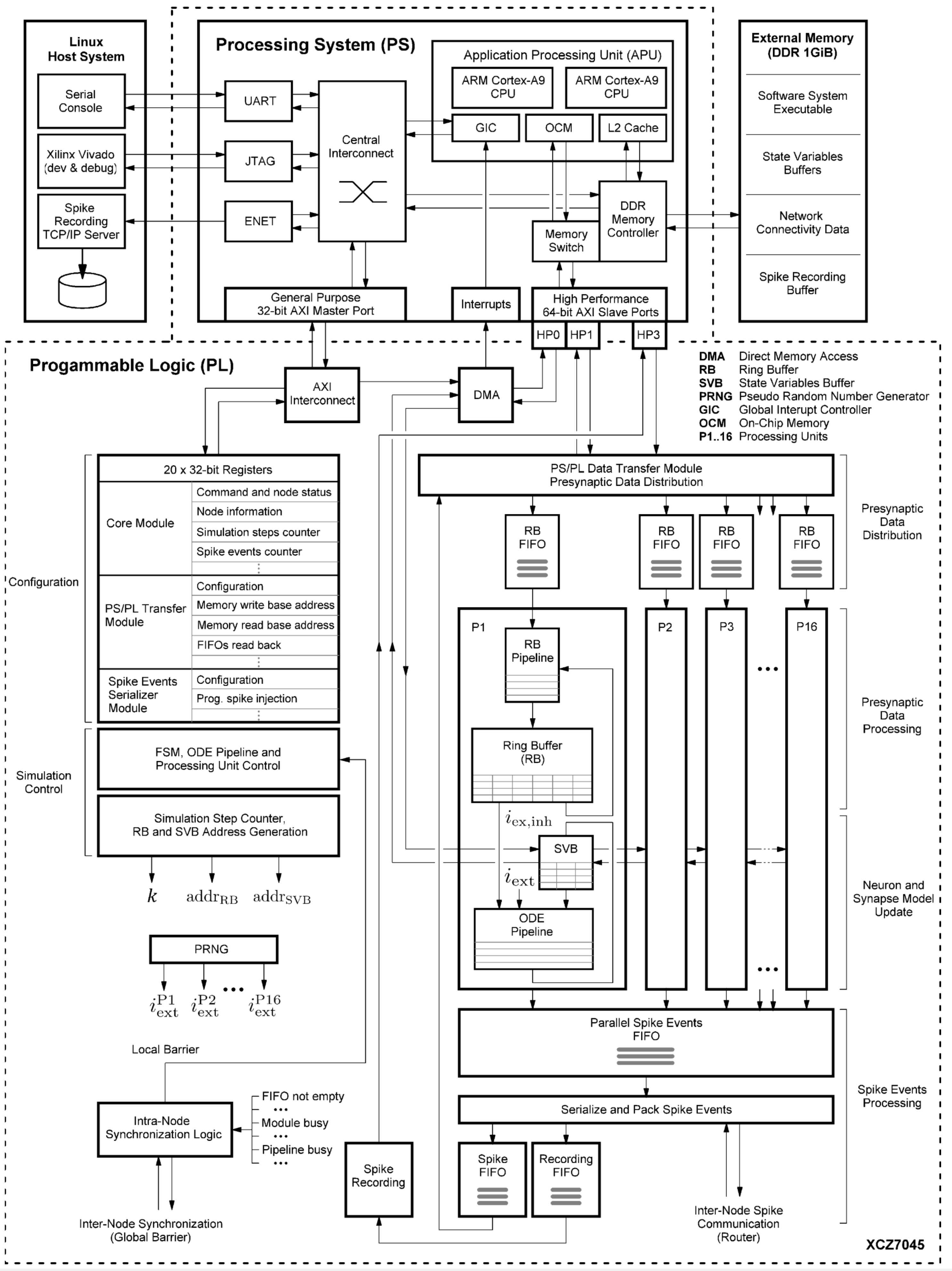
**Acceleration factors derived from operating latencies.**

Max. acceleration factor (single node):
$$F_{\mathrm{S}}^{\mathrm{MAX}} = \frac{hf_{\mathrm{clk}}}{L_{\Sigma}}$$

Without communication (single node):
$$F_{\mathrm{S}}(\bar{\nu}_{\mathrm{k}}) = \begin{cases} \dfrac{hf_{\mathrm{clk}}}{\bar{\nu}_{\mathrm{k}}(L_{\Sigma}^{\mathrm{SE}} + L_{\mathrm{DS}}) + (1-\bar{\nu}_{\mathrm{k}})L_{\Sigma}} & \text{if } \bar{\nu}_{\mathrm{k}} < 1 \\ \dfrac{hf_{\mathrm{clk}}}{L_{\Sigma}^{\mathrm{SE}} + \bar{\nu}_{\mathrm{k}} L_{\mathrm{DS}}} & \text{otherwise} \end{cases}$$

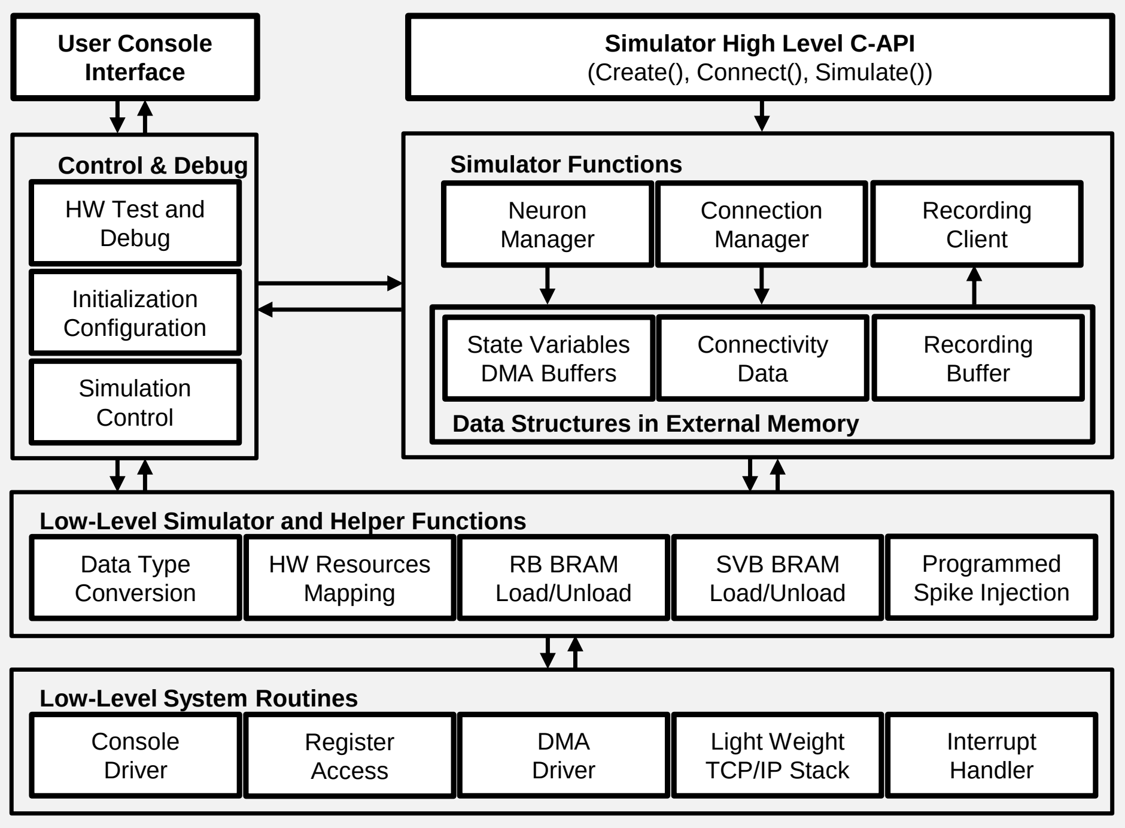
With communication (cluster):
$$F_{\mathrm{C}}(\bar{\nu}_{\mathrm{k}}) = \begin{cases} \dfrac{hf_{clk}}{\bar{\nu}_{\mathrm{k}}(L_{\Sigma}^{\mathrm{SE}} + L_{\mathrm{DS}} + \alpha L_{\mathrm{COM}}) + (1-\bar{\nu}_{\mathrm{k}})L_{\Sigma} + L_{\mathrm{COM}}} & \text{if } \bar{\nu}_{\mathrm{k}} < 1 \\ \dfrac{hf_{clk}}{L_{\Sigma}^{\mathrm{SE}} + \bar{\nu}_{\mathrm{k}}(L_{\mathrm{DS}} + \alpha L_{\mathrm{COM}}) + L_{\mathrm{COM}}} & \text{otherwise} \end{cases}$$

## System-Level Architecture


Linux Host System
Serial Console
Xilinx Vivado (dev & debug)
Spike Recording TCP/IP Server
Processing System (PS)
UART
JTAG
ENET
Central Interconnect
Application Processing Unit (APU)
ARM Cortex-A9 CPU
ARM Cortex-A9 CPU
GIC
OCM
L2 Cache
DDR Memory Controller
Memory Switch
General Purpose 32-bit AXI Master Port
Interrupts
High Performance 64-bit AXI Slave Ports
HP0
HP1
HP3
External Memory (DDR 1GiB)
Software System Executable
State Variables Buffers
Network Connectivity Data
Spike Recording Buffer
Progammable Logic (PL)
AXI Interconnect
DMA
DMA Direct Memory Access
RB Ring Buffer
SVB State Variables Buffer
PRNG Pseudo Random Number Generator
GIC Global Interupt Controller
OCM On-Chip Memory
P1..16 Processing Units
20 x 32-bit Registers
Core Module
Command and node status
Node information
Simulation steps counter
Spike events counter
PS/PL Transfer Module
Configuration
Memory write base address
Memory read base address
FIFOs read back
Spike Events Serializer Module
Configuration
Prog. spike injection
Configuration
FSM, ODE Pipeline and Processing Unit Control
Simulation Step Counter, RB and SVB Address Generation
Simulation Control
$k$
$\mathrm{addr}_{\mathrm{RB}}$
$\mathrm{addr}_{\mathrm{SVB}}$
PRNG
$i_{\mathrm{ext}}^{\mathrm{P1}}$
$i_{\mathrm{ext}}^{\mathrm{P2}}$
$i_{\mathrm{ext}}^{\mathrm{P16}}$
Local Barrier
Intra-Node Synchronization Logic
FIFO not empty
Module busy
Pipeline busy
Inter-Node Synchronization (Global Barrier)
Spike Recording
PS/PL Data Transfer Module Presynaptic Data Distribution
RB FIFO
RB FIFO
RB FIFO
RB FIFO
Presynaptic Data Distribution
P1
P2
P3
P16
RB Pipeline
Ring Buffer (RB)
$i_{\mathrm{ex,inh}}$
SVB
$i_{\mathrm{ext}}$
ODE Pipeline
Presynaptic Data Processing
Neuron and Synapse Model Update
Parallel Spike Events FIFO
Serialize and Pack Spike Events
Spike FIFO
Recording FIFO
Spike Events Processing
Inter-Node Spike Communication (Router)
XCZ7045

**Software system executed on the APU**

- Orchestrates the overall node operation.
- A minimal C-API provides `Create()`, `Connect()`, and `Simulate()` function calls.
- Implemented as bare-metal application in C.



**User Console Interface**

**Simulator High Level C-API**
(Create(), Connect(), Simulate())

**Control & Debug**
HW Test and Debug
Initialization Configuration
Simulation Control

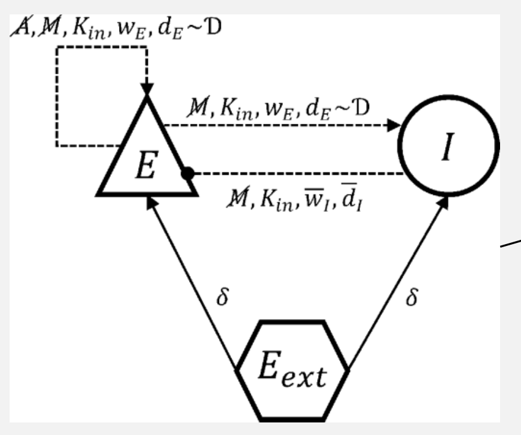
**Simulator Functions**
Neuron Manager
Connection Manager
Recording Client
State Variables DMA Buffers
Connectivity Data
Recording Buffer
**Data Structures in External Memory**

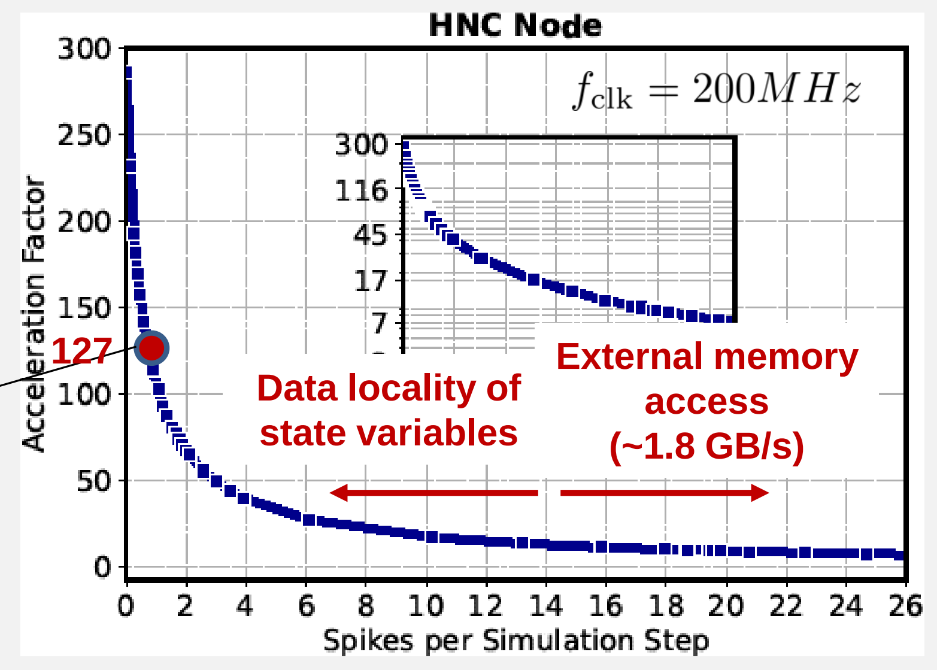
**Low-Level Simulator and Helper Functions**
Data Type Conversion
HW Resources Mapping
RB BRAM Load/Unload
SVB BRAM Load/Unload
Programmed Spike Injection

**Low-Level System Routines**
Console Driver
Register Access
DMA Driver
Light Weight TCP/IP Stack
Interrupt Handler

## Single Node Performance

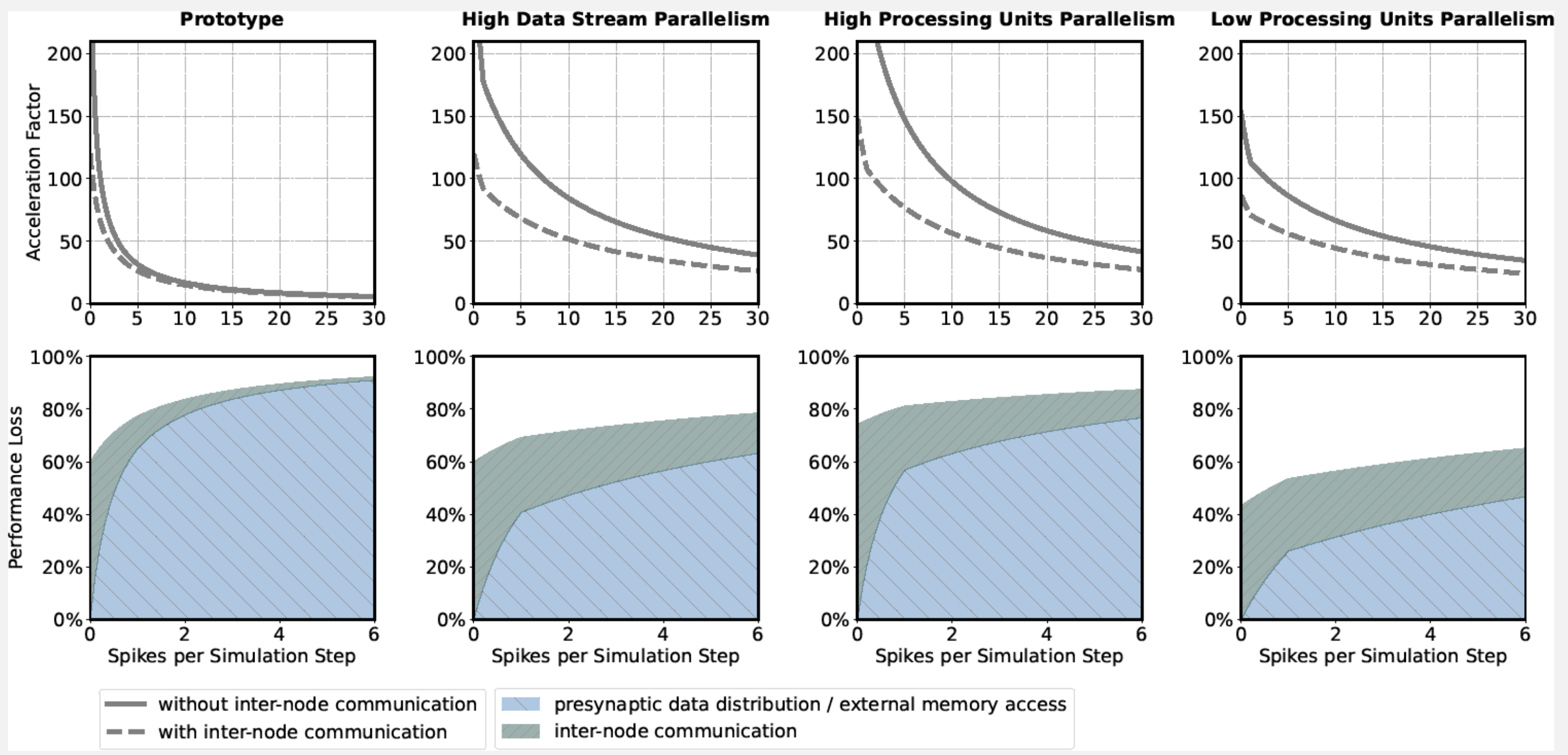
**1000 neurons two-population Izhikevich network.**


$\mathcal{A}, \mathcal{M}, K_{in}, w_E, d_E \sim \mathcal{D}$
$\mathcal{M}, K_{in}, w_E, d_E \sim \mathcal{D}$
$E$
$I$
$\mathcal{M}, K_{in}, \overline{w}_I, \overline{d}_I$
$\delta$
$\delta$
$E_{ext}$



Simulate workload: consecutive simulation runs of 5 min simulated biological time with an increasing external offset current. $i_{\mathrm{ext}} = \{-3.0pA, .., +100pA\}$

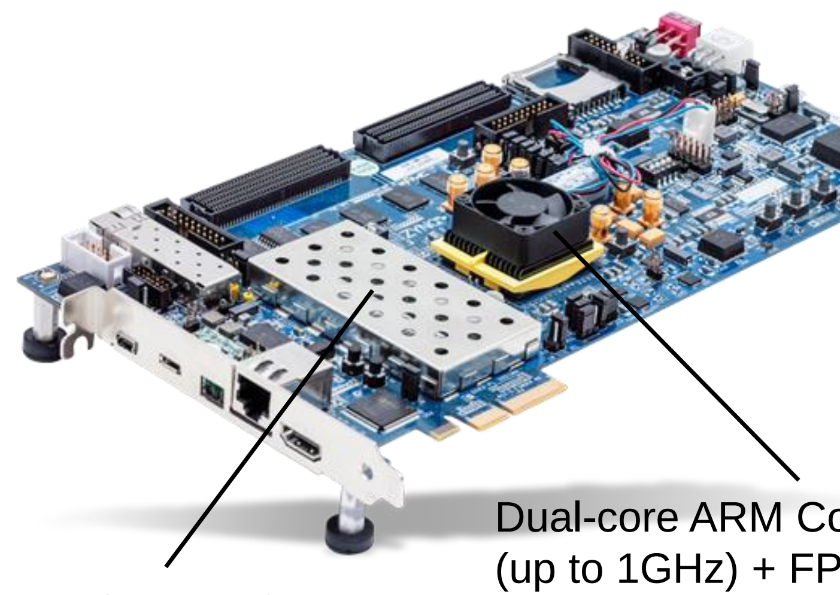
## Performance Characteristics



| Parameters | Prototype | High Data Stream Parallelism | High Proc. Units Parallelism | Low Proc. Units Parallelism |
|---|---|---|---|---|
| number of data streams, $DS$ | 2 | 16 | 16 | 16 |
| data stream latency, $L_{\mathrm{DS}}$ | 110 | 14 | 14 | 14 |
| number processing units, $P$ | 16 | 16 | 32 | 8 |
| number of neurons per processing unit, $N^{\mathrm{P}}$ | 64 | 64 | 32 | 128 |
| ODE pipeline iteration latency, $IL_{\mathrm{N}}$ | 64 | 64 | 32 | 128 |
| **Acceleration factors w/o communication** | | | | |
| maximum, $F_{\mathrm{S}}^{\mathrm{MAX}} = F_{\mathrm{S}}(\bar{\nu}_{\mathrm{k}} = 0)$ | 298.5 | 298.5 | 571.4 | 152.7 |
| low workload, $F_{\mathrm{S}}(1.0)$ | 104.7 | 177.0 | 246.9 | 113.0 |
| medium workload, $F_{\mathrm{S}}(10.0)$ | 16.9 | 84.5 | 97.7 | 66.5 |
| high workload, $F_{\mathrm{S}}(20.0)$ | 8.8 | 52.4 | 58.4 | 45.6 |
| **Acceleration factors with communication** | | | | |
| maximum, $F_{\mathrm{C}}^{\mathrm{MAX}} = F_{\mathrm{C}}(0)$ | 119.8 | 119.8 | 148.1 | 86.6 |
| low workload, $F_{\mathrm{C}}(1.0)$ | 67.6 | 91.7 | 107.5 | 70.9 |
| medium workload, $F_{\mathrm{C}}(10.0)$ | 15.0 | 51.7 | 56.4 | 44.4 |
| high workload, $F_{\mathrm{C}}(20.0)$ | 8.1 | 34.8 | 36.9 | 31.3 |

## Prototyping Platform

AMD Xilinx Zynq®-7000 SoC ZC706 Development Board



1GB DDR3 external memory

Dual-core ARM Cortex-A9 CPU (up to 1GHz) + FPGA (250K logic cells, 19.1Mb Block RAM, 900 DSP slices)

Contact: Guido Trensch, g.trensch@fz-juelich.de