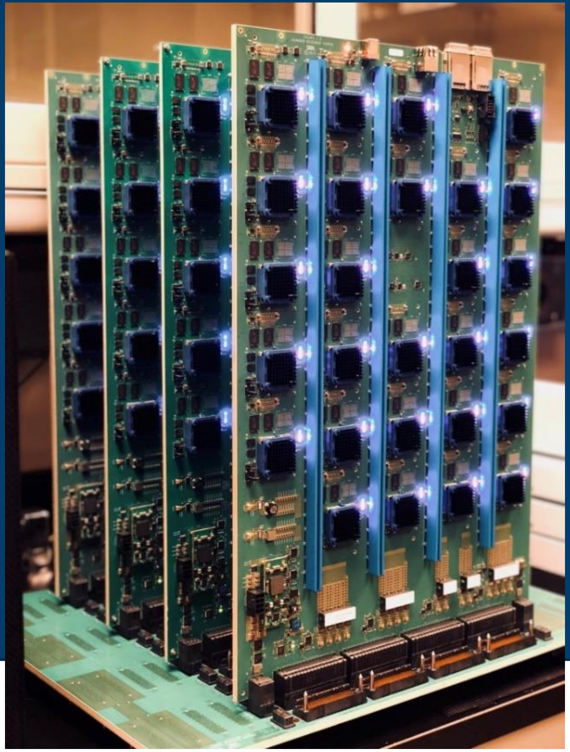
RECORD SIMULATION OF THE FULL-DENSITY SPIKING POTJANS-DIESMANN-MICROCIRCUIT MODEL ON THE IBM NEURAL SUPERCOMPUTER INC 3000

Arne Heittmann[1], Georgia Psychou[1], Guido Trensch[2], Charles E. Cox[3], Winfried W. Wilcke[3], Markus Diesmann[4] and Tobias G. Noll[1]



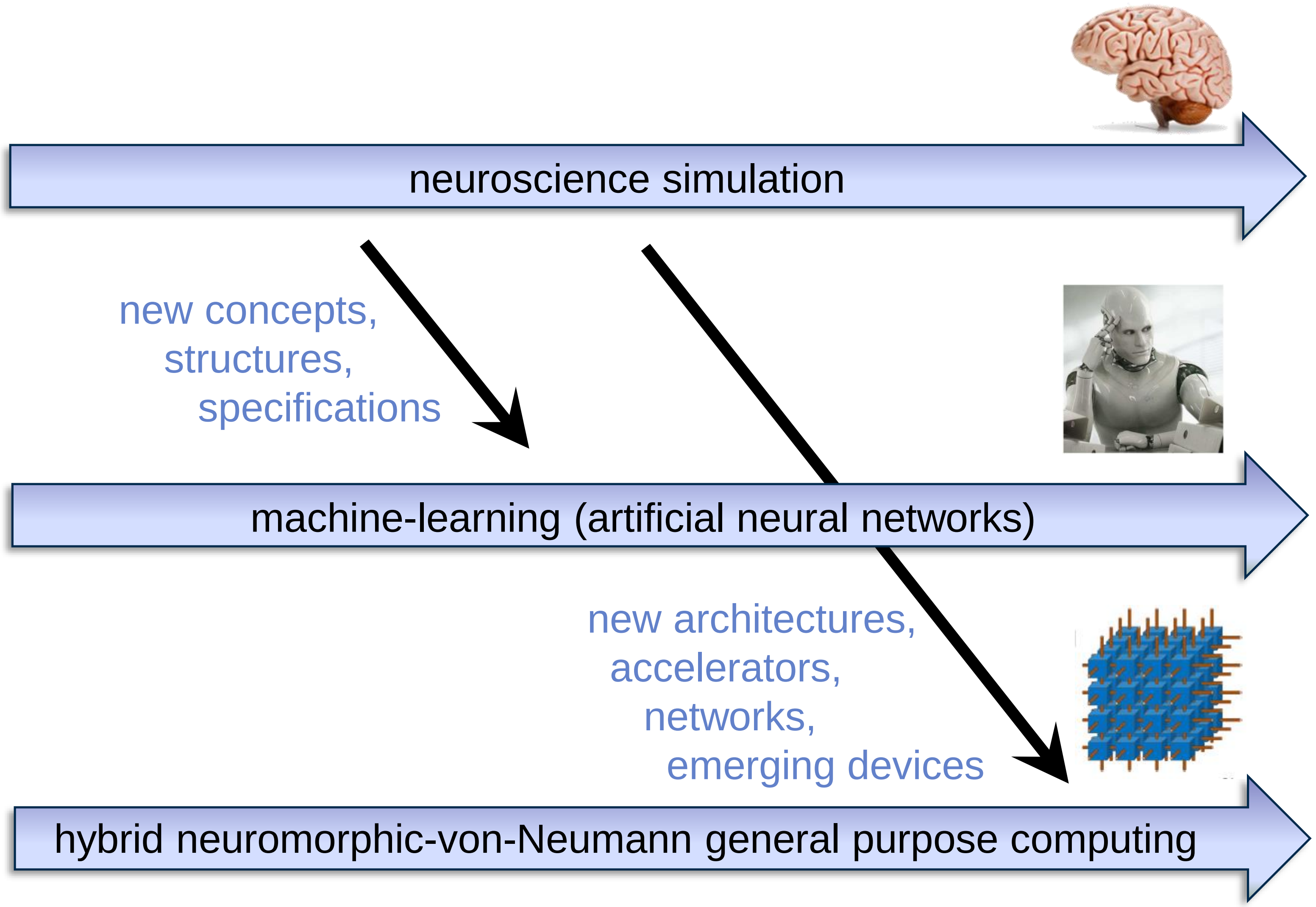
March 30 2022

ACA- towards multi-scale natural-density Neuromorphic Computing

[1]JARA-Institute Green IT (PGI-10), Jülich Research Centre, D-52425 Jülich, Germany
[2]Simulation & Data Lab Neuroscience, Jülich Supercomputing Centre Jülich Research Centre, D-52425 Jülich, Germany
[3]IBM Research Division, Almaden Research Center, San Jose, CA 9512
[4]Institute of Neuroscience and Medicine (INM-6)

NEUROMORPHIC COMPUTING

- towards advanced general purpose computing architectures



neuroscience simulation

new concepts,
structures,
specifications

machine-learning (artificial neural networks)

new architectures,
accelerators,
networks,
emerging devices

hybrid neuromorphic-von-Neumann general purpose computing

Neuroscience

- dynamics of natural brains
- function of natural brains
- learning, plasticity, development

Artificial General Intelligence (AGI)

- deep learning using few samples
- ability for contextual adaptation
- ability to explain descisions in natural language

ACA – ADVANCED COMPUTING ARCHITECTURES

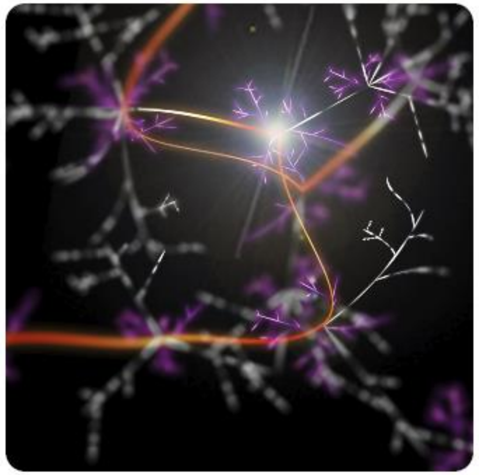
www.fz-juelich.de/aca

towards multi-scale natural-density neuromorphic computing

- Pilot project preparing a long-term research initiative in the application area of *Neuroscience Simulation*
- Specification of a neuromorphic computing architecture for accelerating simulation experiments

Requirements, Validation & Benchmarking

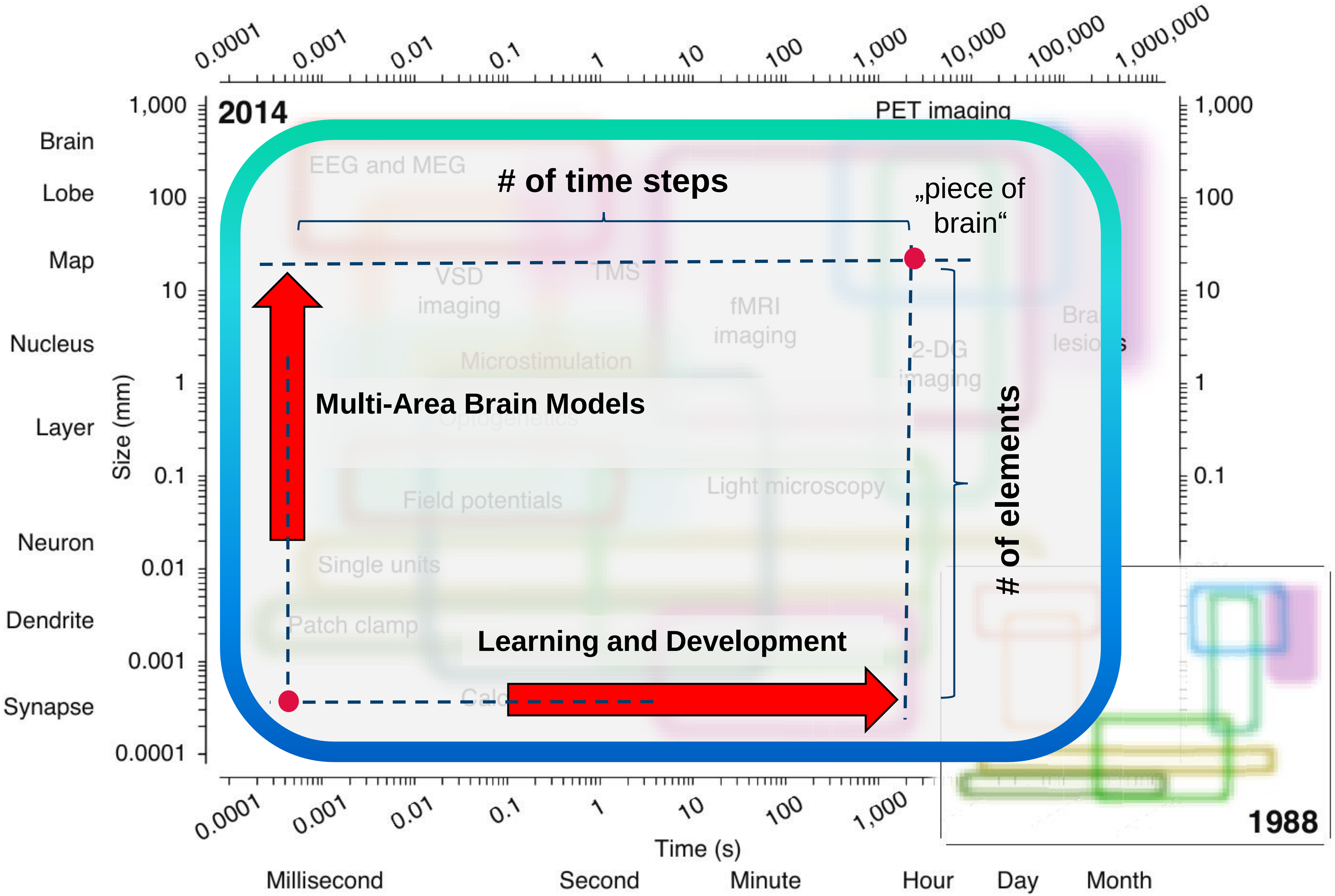
System Definition, Integration & Operation



Network Connectivity & Communication

Accelerated Numerics

SCALES OF BRAIN ORGANIZATION



Sejnowski et al. (2014) Nature Neuroscience

Trends in Neuroscience Simulation Experiments

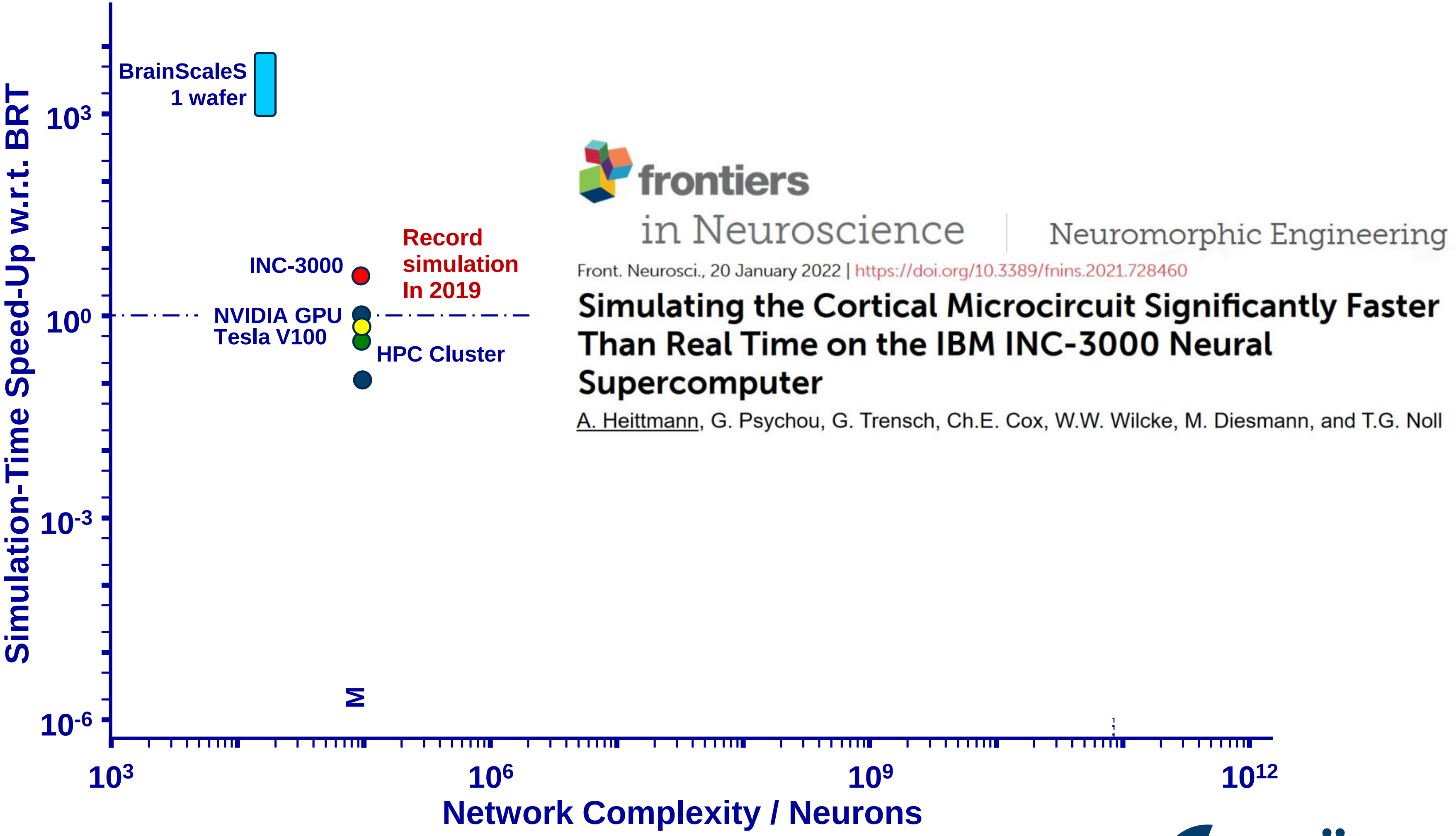
BrainScaleS,
Univ. Heidelberg

IBM INC 3000
FZ-Jülich

JUQUEEN
FZ-Jülich

K-Computer Riken

State-of-the-Art Simulation Time (0.1-ms time grid)



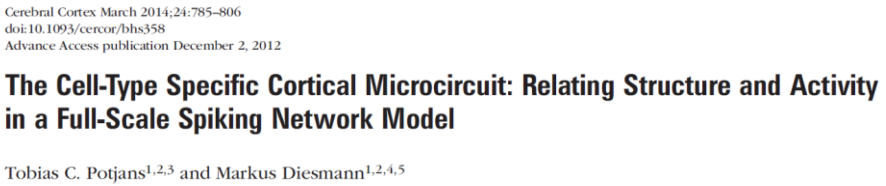
frontiers in Neuroscience | Neuromorphic Engineering

Front. Neurosci., 20 January 2022 | https://doi.org/10.3389/fnins.2021.728460

Simulating the Cortical Microcircuit Significantly Faster Than Real Time on the IBM INC-3000 Neural Supercomputer

A. Heittmann, G. Psychou, G. Trensch, Ch.E. Cox, W.W. Wilcke, M. Diesmann, and T.G. Noll

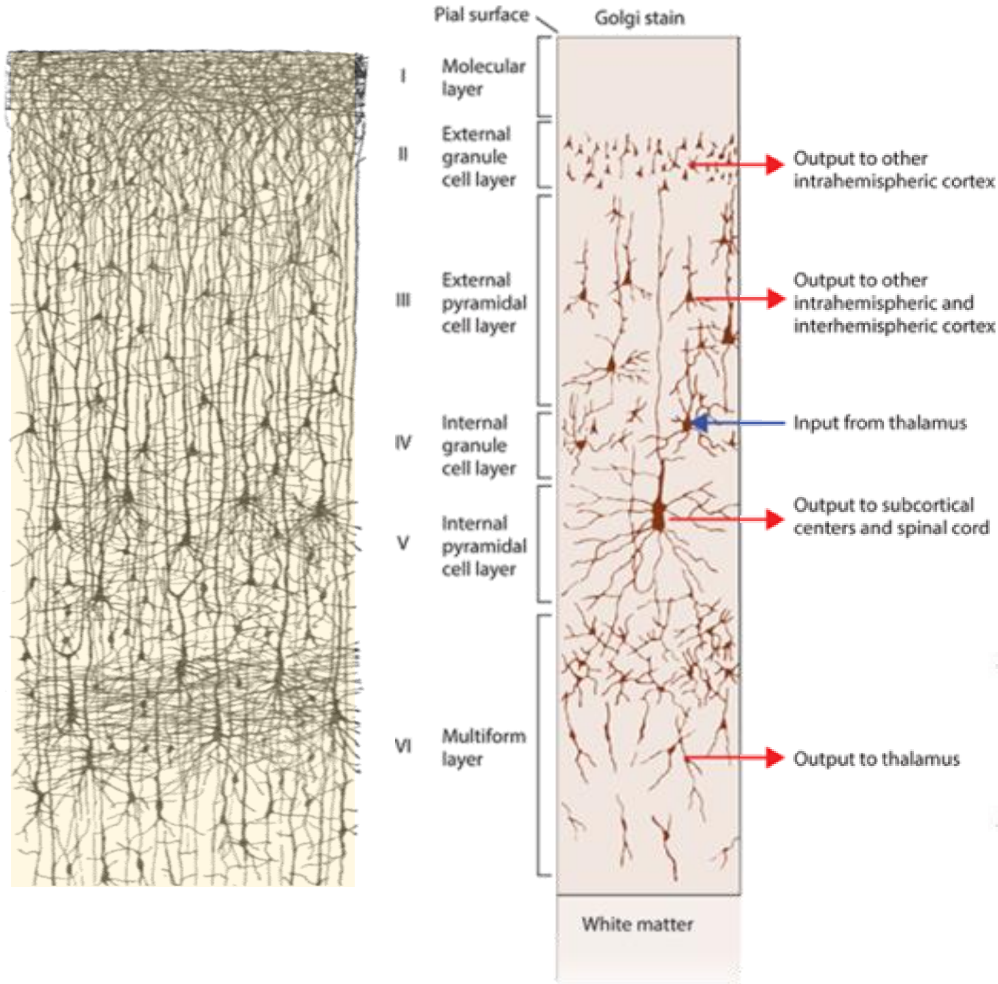
THE CORTICAL MICROCIRCUIT



Cerebral Cortex March 2014;24:785–806
doi:10.1093/cercor/bhs358
Advance Access publication December 2, 2012

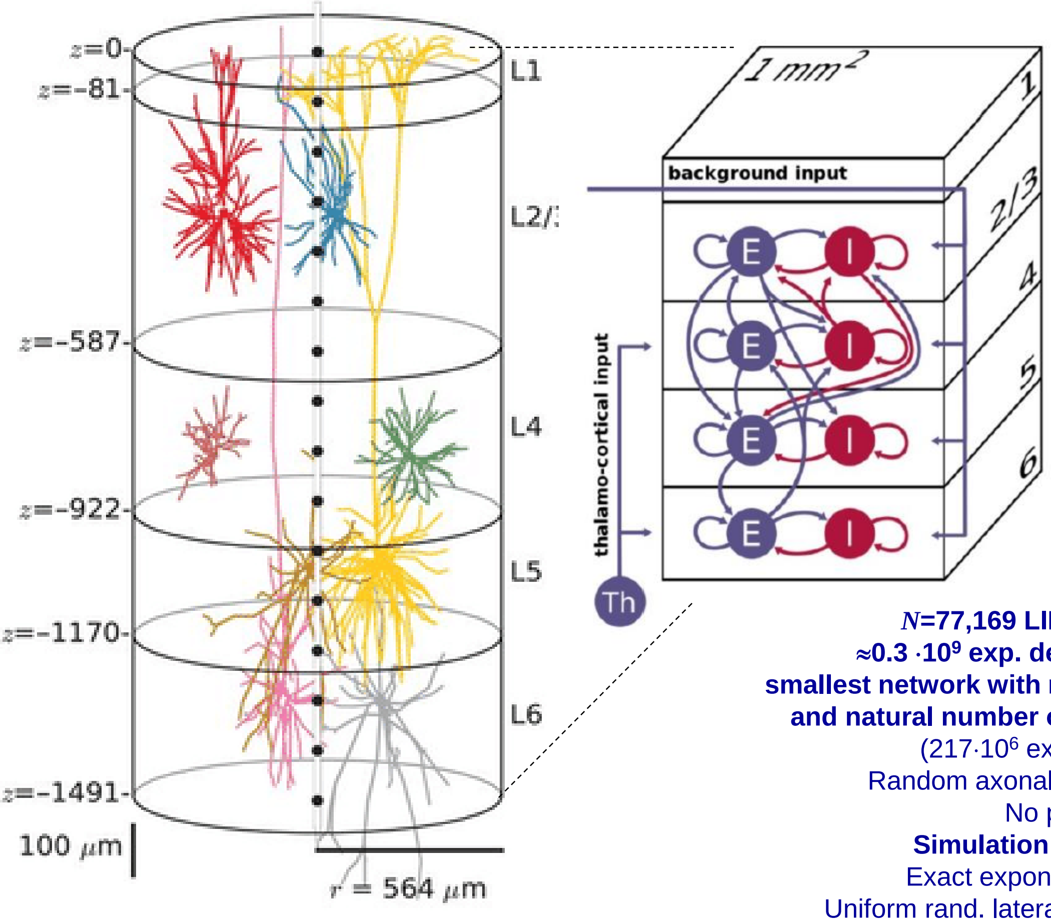
The Cell-Type Specific Cortical Microcircuit: Relating Structure and Activity in a Full-Scale Spiking Network Model

Tobias C. Potjans[1,2,3] and Markus Diesmann[1,2,4,5]



Pial surface
Golgi stain
I
Molecular layer
II
External granule cell layer
III
External pyramidal cell layer
IV
Internal granule cell layer
V
Internal pyramidal cell layer
VI
Multiform layer
White matter
Output to other intrahemispheric cortex
Output to other intrahemispheric and interhemispheric cortex
Input from thalamus
Output to subcortical centers and spinal cord
Output to thalamus

Source: Tony Mosconi, Victoria Graham: *Neuroscience for Rehabilitation*
Copyright © McGraw-Hill Education. All ri



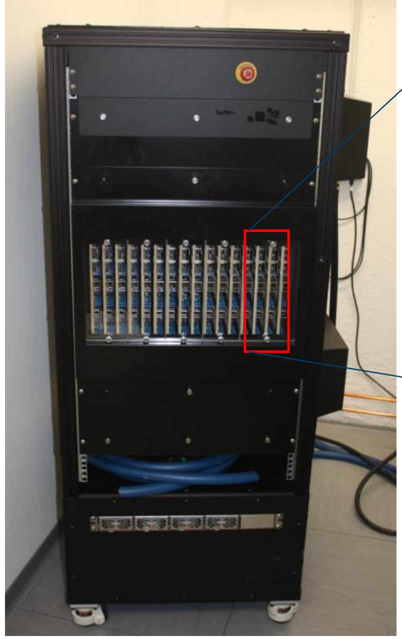
N=77,169 LIF point-neurons
≈0.3 ·10^9 exp. dec. CUBA synapses
smallest network with natural connection density
and natural number of synapses per dendrite
($217 \cdot 10^6$ exc., $82 \cdot 10^6$ inh.)
Random axonal propagation delays
No plasticity
Simulation: 0.1ms grid
Exact exponential integration
Uniform rand. lateral connection probability

THE IBM INC 3000 NEURAL SUPERCOMPUTER

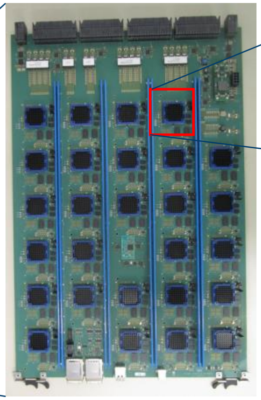
- Originally developed for the IBM AGI (Artificial General Intelligence) Project (IBM Research, Almaden, CA)

- Platform for development and evaluation of prototypical circuit and architecture concepts for ACA

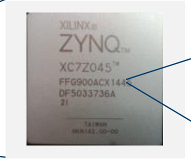
INC 3000



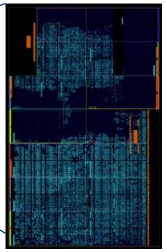
- 16 x INC board
- 432 SoC Nodes



- INC board
- 27 x Xilinx XC 7Z045 SOC

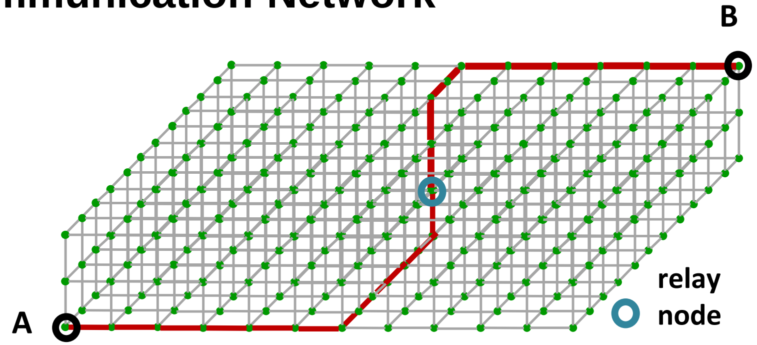


- XC 7Z045 SOC
- 200 k LUTs
- 400 k FFs
- 900 DSPs
- 19.2 Mb BRAM
- 12.5 Gb/s GTX tranceivers
- 2 x ARM A9



- FPGA logic for Simulator, Router, and Configuration

Communication Network



Communication Network Topology:

12 x 12 x 3   3D-node mesh

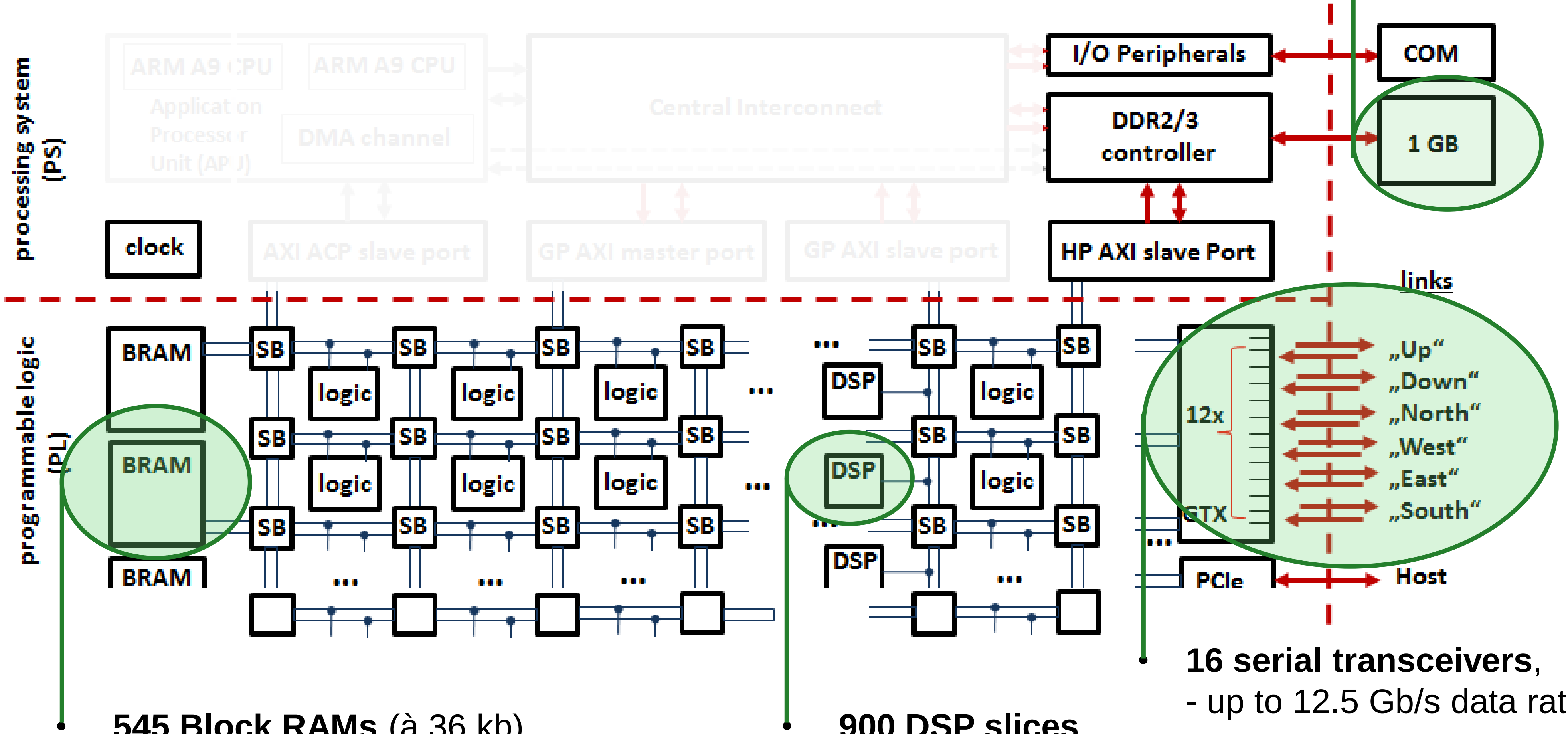
Cross-sectional bandwidth:

B = 450 Gb/s

Worst case packet-path (betw. A and B)

Latency T=24 µs

XILINX 7Z045 SOC: THE COMPUTE NODE



1 GB SRRAM external

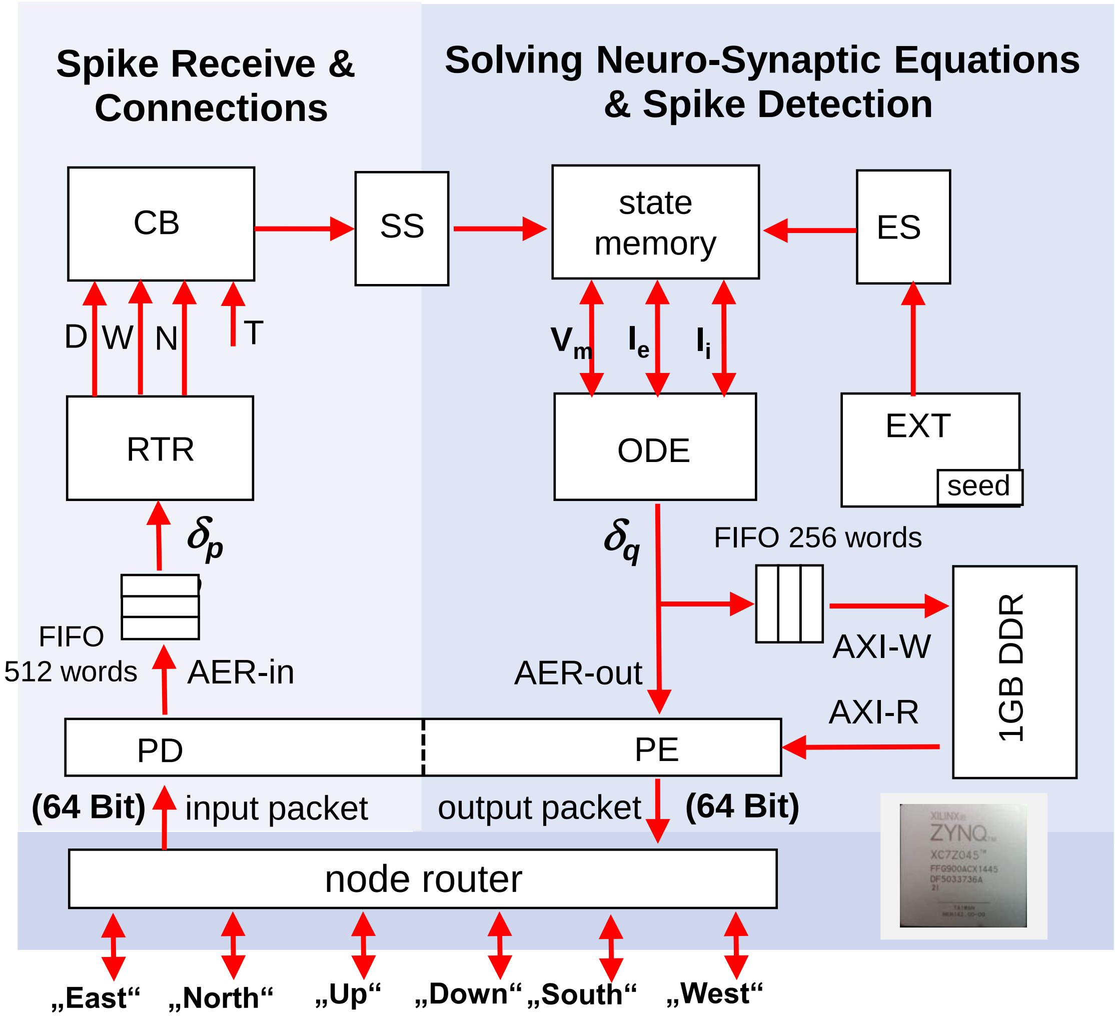
16 serial transceivers,
- up to 12.5 Gb/s data rate (GTX)

545 Block RAMs (à 36 kb) ,
- total 19.2 Mbit
- configurable as single/dual port SRAM
- up to 72 bit data width

900 DSP slices
- 18x25 bit signed multiply
- 48-bit adder/accumulator

~ 180 single float Adder/Multiplier

NODE SCHEMATICS, MODELS, AND DESIGN FLOW



Simulation Node Schematics

- **256-Neuron-Fascicle per Node**
- **Various Point-Neuron Models**

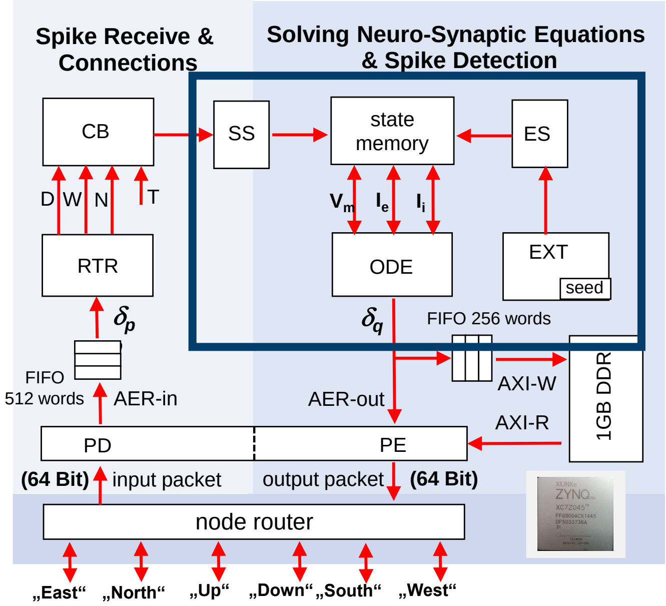
  - LIF, MAT2, Izhikevich, AdEx
- **Several Synapse Models**

  - CUBA- and COBA-based,
  - Exponential Decay-, Alpha-, Beta-Shaped
- **Several ODE System Solvers**

  - Exact Exponential
  - Runge-Kutta
  - Parker-Sochacki
- **Network Generation „on the fly“ during simulation run time based on highly efficient PRNGs („Procedural Connectivity“)**
- **Node-Synchronization by Barrier-Messages**

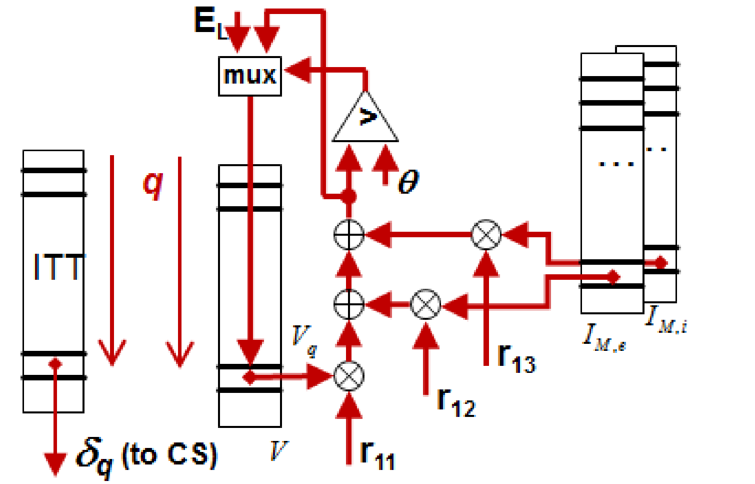
  - avoid spike-loss
- **Single-Float Precision Arithmetics**
- **High-Level-Synthesis Design Flow**

ODE SOLVER



- membrane of a point-neuron (LIF)

$$\tau_m \cdot \frac{dV_q}{dt} = -(V_q - E_L) + R_M \cdot I_{M,q}$$

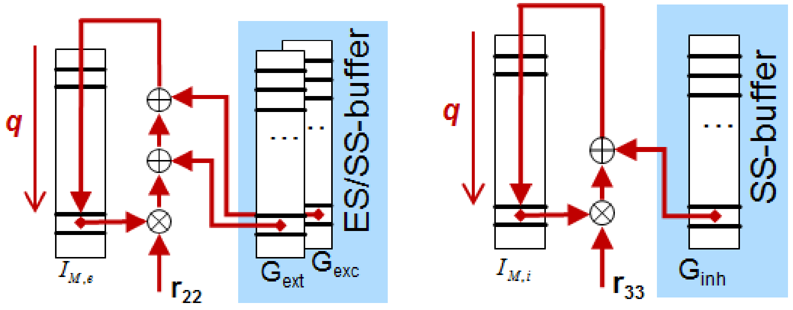


synthesized datapaths, pipelined

- lumped CUBA-synapses

- exponential decay

$$\tau_{s,x} \cdot \frac{dI_{M,q,x}}{dt} = -I_{M,q,x} + I_{S,q,x} \quad , \; x \in \{e, i\}$$



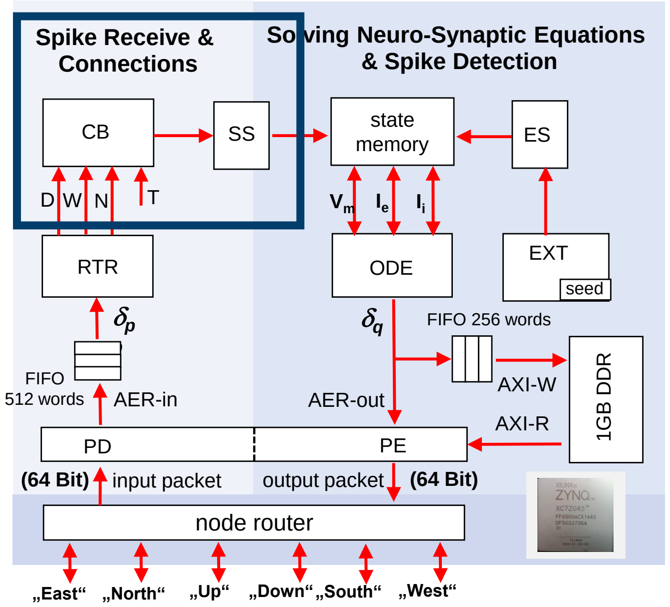
- State-Vector update

- exact exponential for linear ODEs

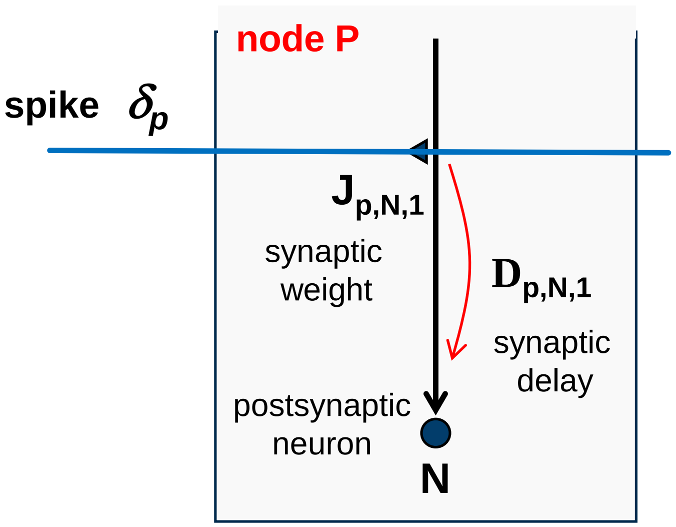
$$\begin{bmatrix} V_q \\ I_e \\ I_i \end{bmatrix}(t+h) \leftarrow \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ 0 & r_{22} & 0 \\ 0 & 0 & r_{33} \end{bmatrix} \cdot \begin{bmatrix} V_q \\ I_e \\ I_i \end{bmatrix}(t)$$

update equation, 3 state variables per neuron

LUMPED SYNAPSES



- network schematic



- synaptic equation

synaptic multiplicity (multapses)

pre-synaptic spike train

$$I_{S,q,x}(T) = \sum_{p \in B_x} \sum_{m=1}^{M_{q,p}} J_{q,p,m} \cdot S_p(T - D_{q,p,m})$$

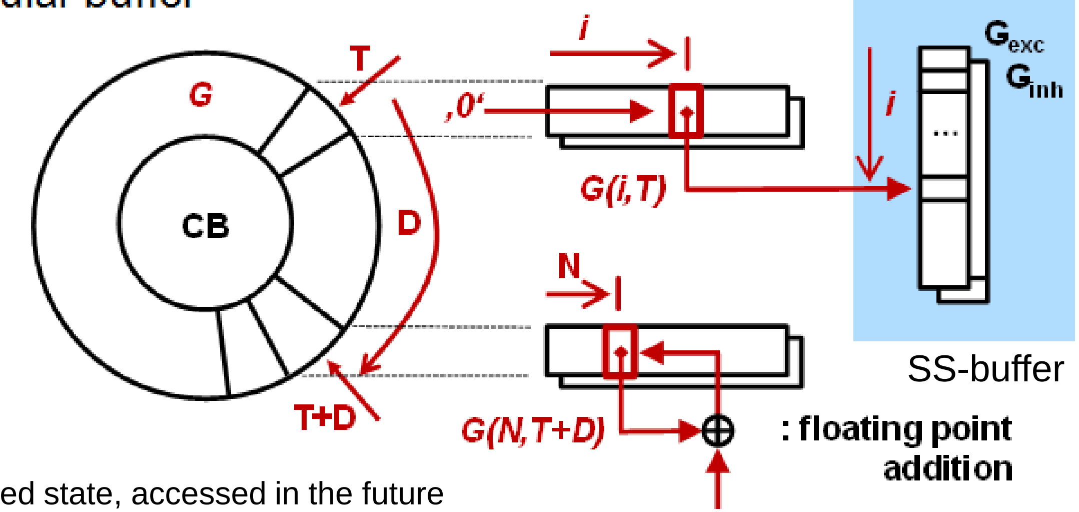
synaptic delay

synaptic weight

$$S_p(t) = \sum_i \delta(t - t_{p.i})$$

- circular buffer

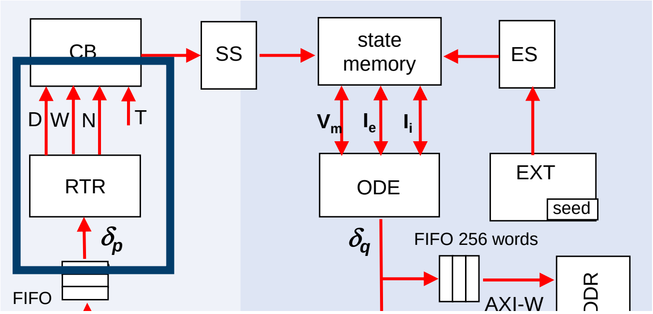
state, valid for time step T



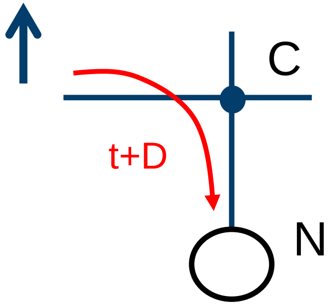
delayed state, accessed in the future

+ addressing by ‚time‘ **T** and ‚neuron index‘ **N**

PROCEDURAL CONNECTIVITY

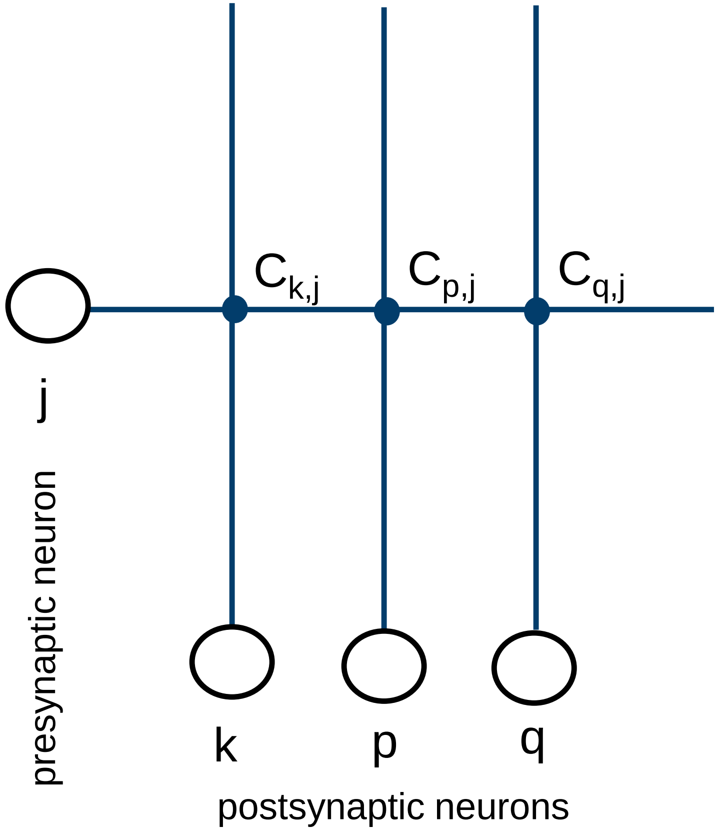


- **Parameters of a synapse C**

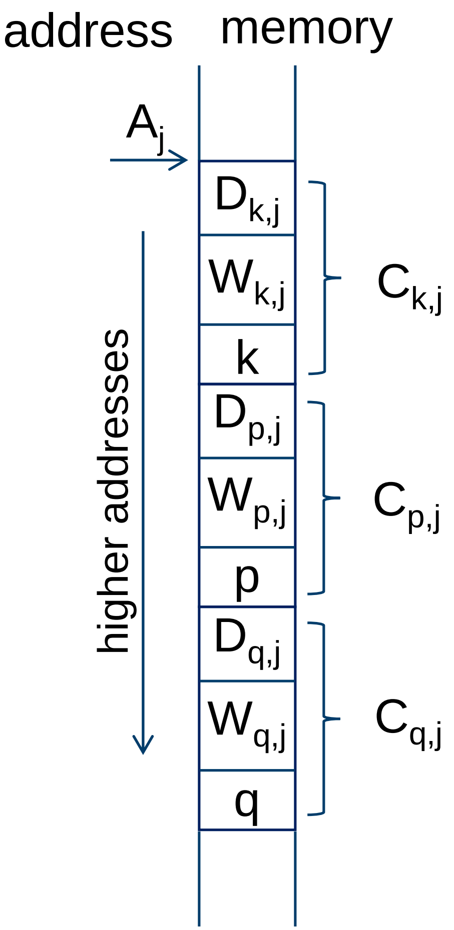


C

D	ax. + dend. delay
W	weight
N	postsynaptic neuron

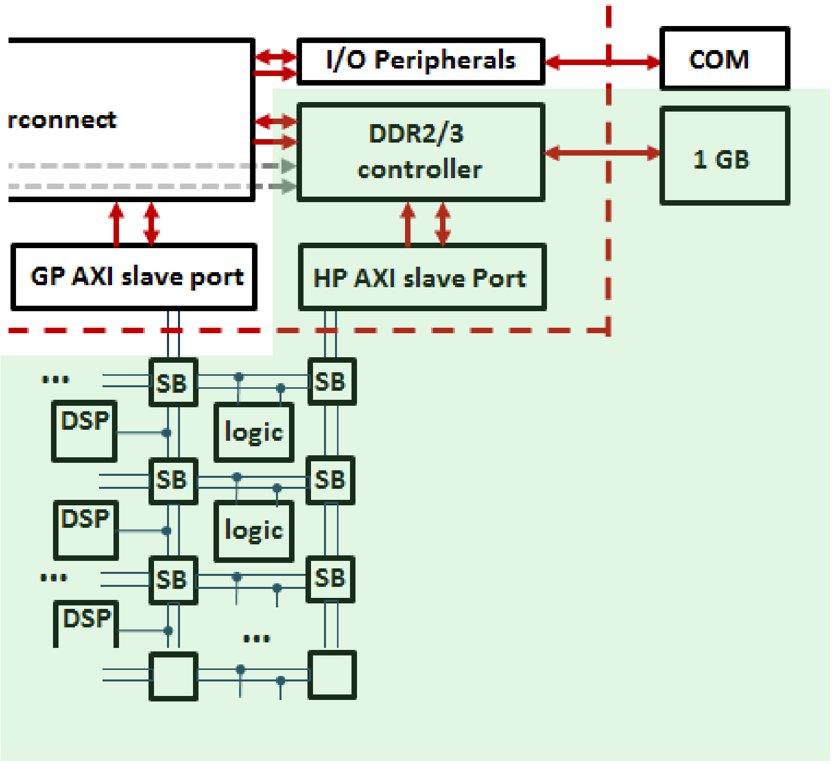


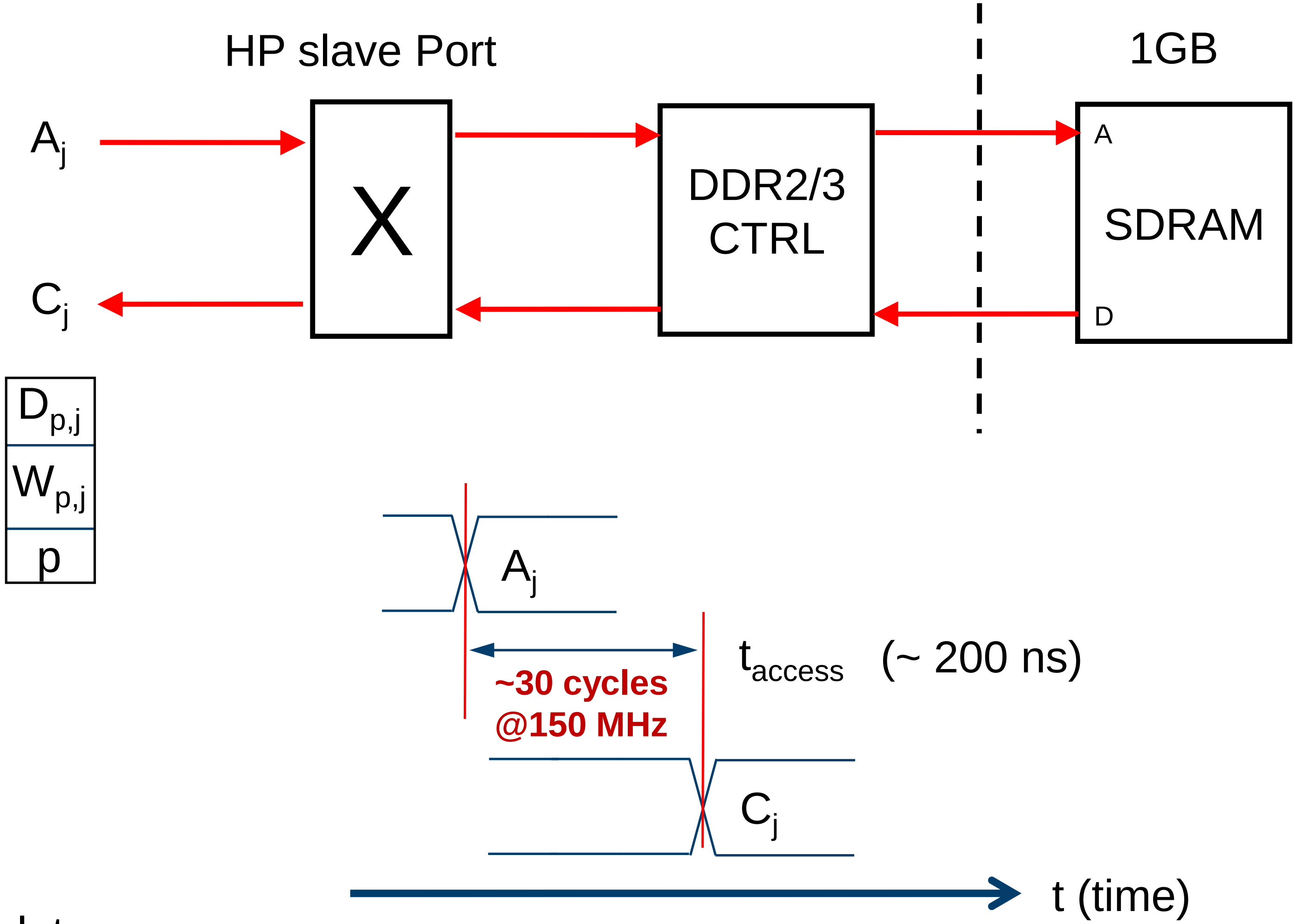
- **example network**



- **(naive) memory layout on the receiving node**

PROCEDURAL CONNECTIVITY



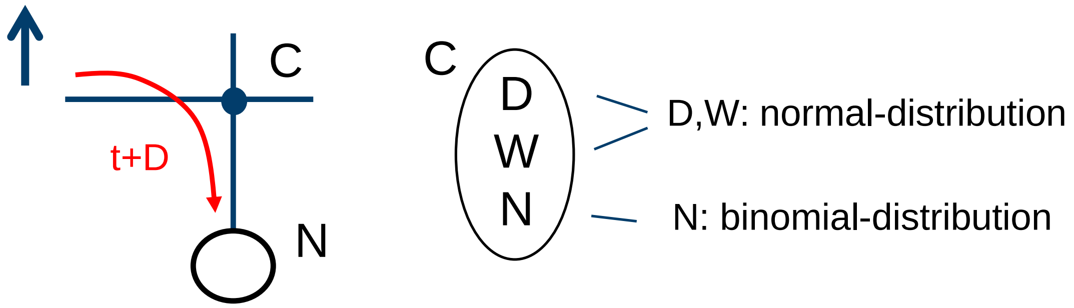


- access latency

- minimal time from ‚address valid“ to „first data out“

NETWORK GENERATION

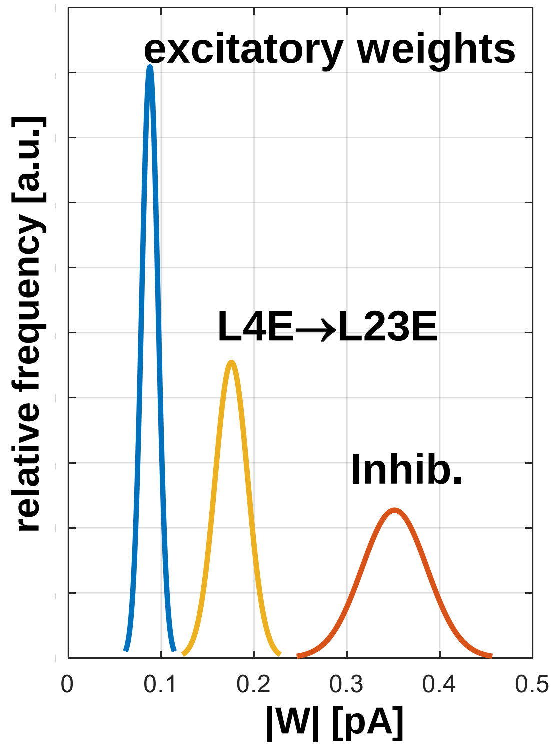
Microcircuit: connections *C* are defined by pseudo random numbers



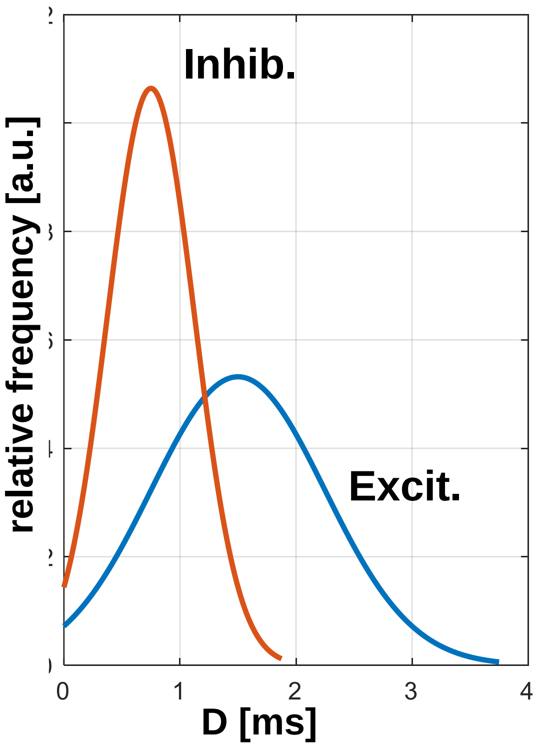
D,W: normal-distribution

N: binomial-distribution

weight distribution



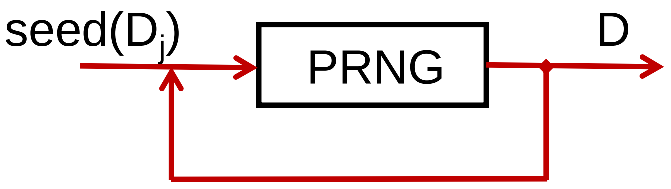
delay distribution



- In NEST: all synapse parameters are drawn offline using pseudo-random-number generators (PRNGs)
- **Alternative approach**: implement these random number generators on-the-chip

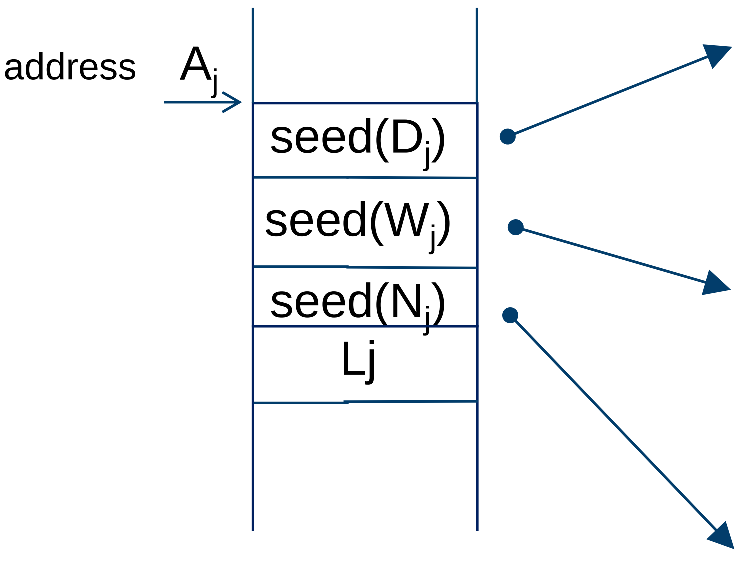
  - re-generate the synaptic parameters „on-the-fly“, when needed

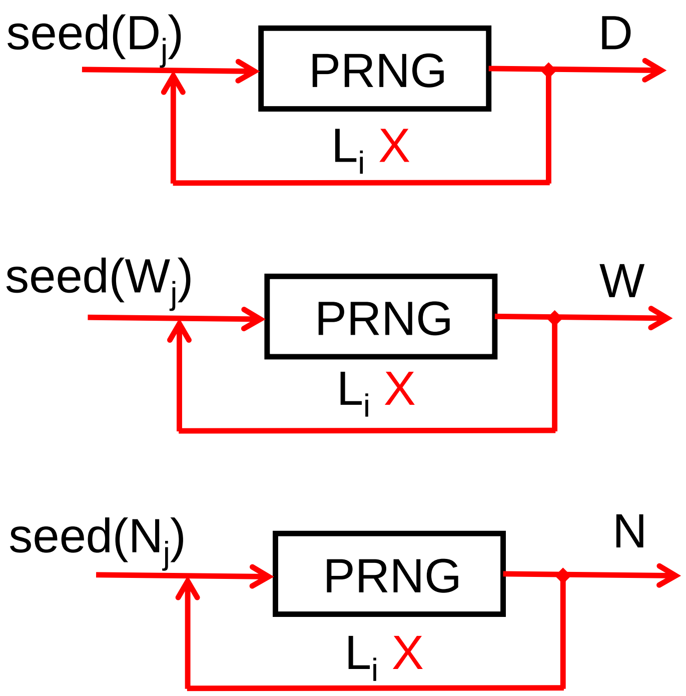
  - initial seeds define the deterministic sequence of random numbers



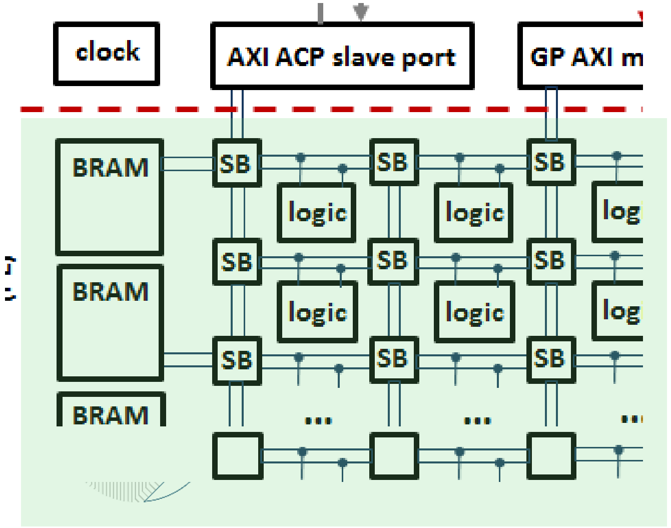
PROCEDURAL CONNECTIVITY

seed-memory (on-chip BRAM)

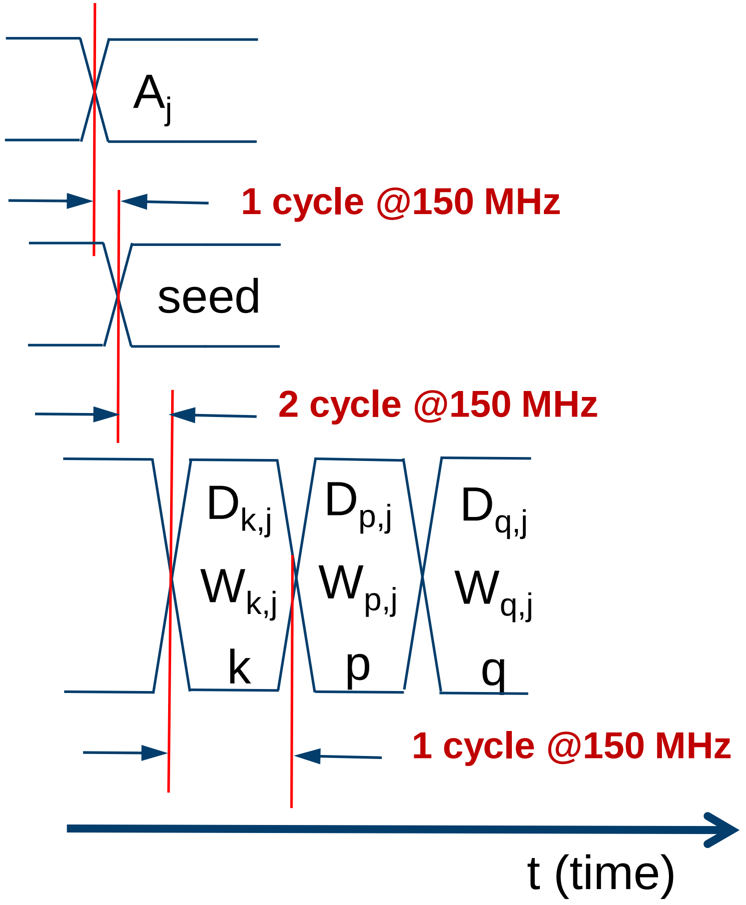




Pseudo-Random-Number Generators: create sequences in parallel



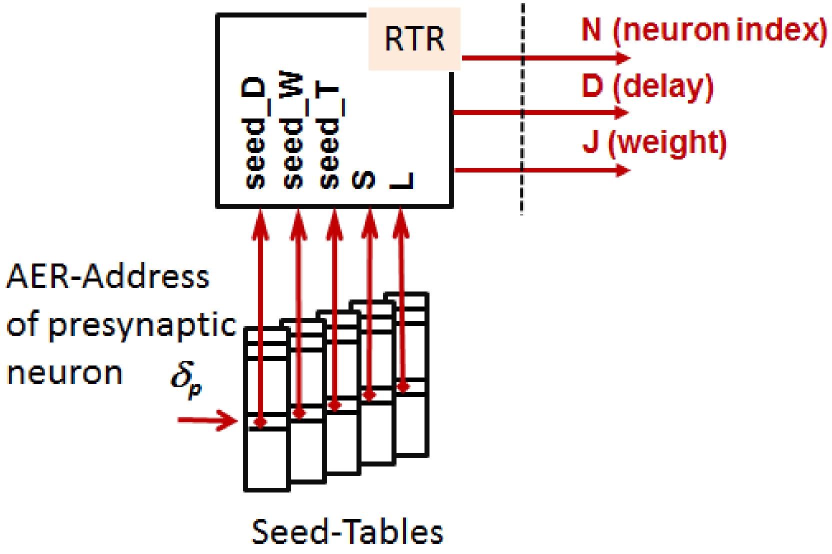
Timing



JÜLICH
Forschungszentrum

PROCEDURAL CONNECTIVITY

- seed-lookup tables

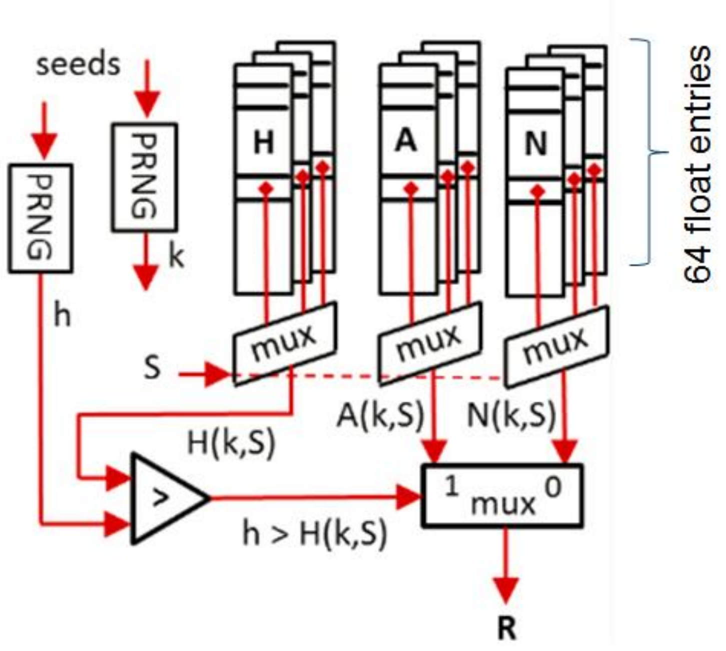


- 96 seed-bits per presynaptic neuron

- compression factor ~ 6.5

- full connectome can be stored in the local BRAM

- RTR data path

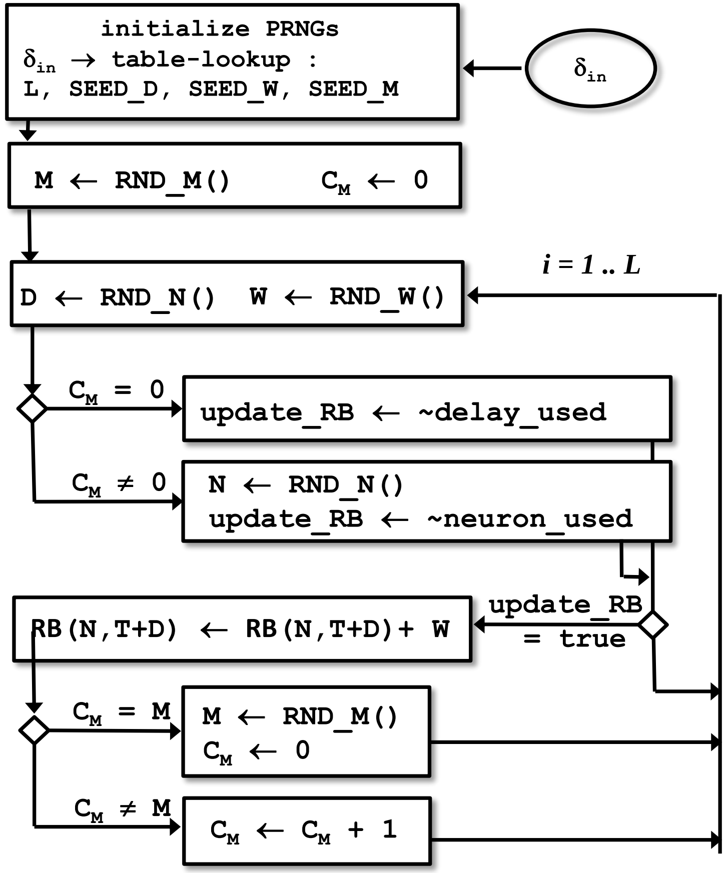


- Walker‘s table-based PRNG

- 64 sampling points per CDF/PDF

Walker, A. J., "An Efficient Method for Generating Discrete Random Variables with General Distributions". ACM Transactions on Mathematical Software. **3** (3): 253–256, 1977, doi:10.1145/355744.355749

- RTR control loop



initialize PRNGs
δ_{in} → table-lookup :
L, SEED_D, SEED_W, SEED_M

δ_{in}

M ← RND_M() C_M ← 0

i = 1 .. L

D ← RND_N() W ← RND_W()

C_M = 0

update_RB ← ~delay_used

C_M ≠ 0

N ← RND_N()
update_RB ← ~neuron_used

update_RB = true

RB(N,T+D) ← RB(N,T+D)+ W

C_M = M

M ← RND_M()
C_M ← 0

C_M ≠ M

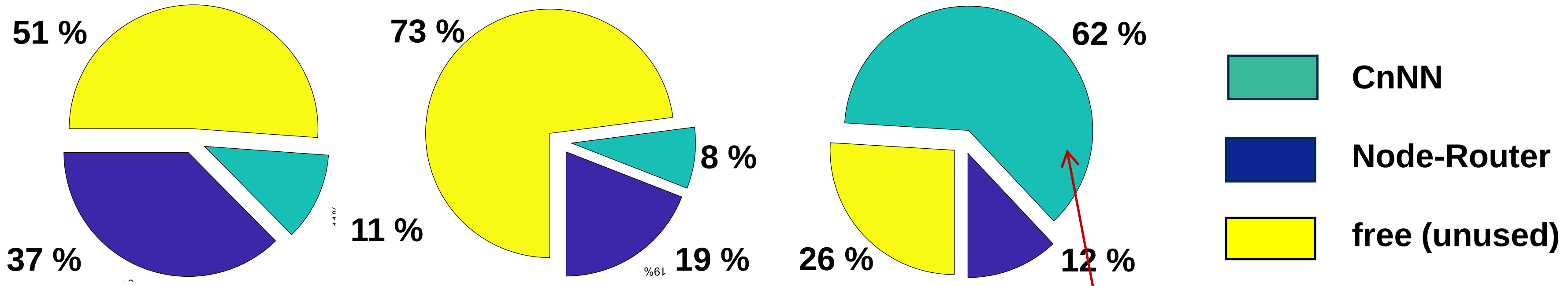
C_M ← C_M + 1

- latency: 3 cycles @ 150 MHz

- fully pipelined

Gate and Memory Breakdown (LIF, CUBA)

- Overall FPGA resources

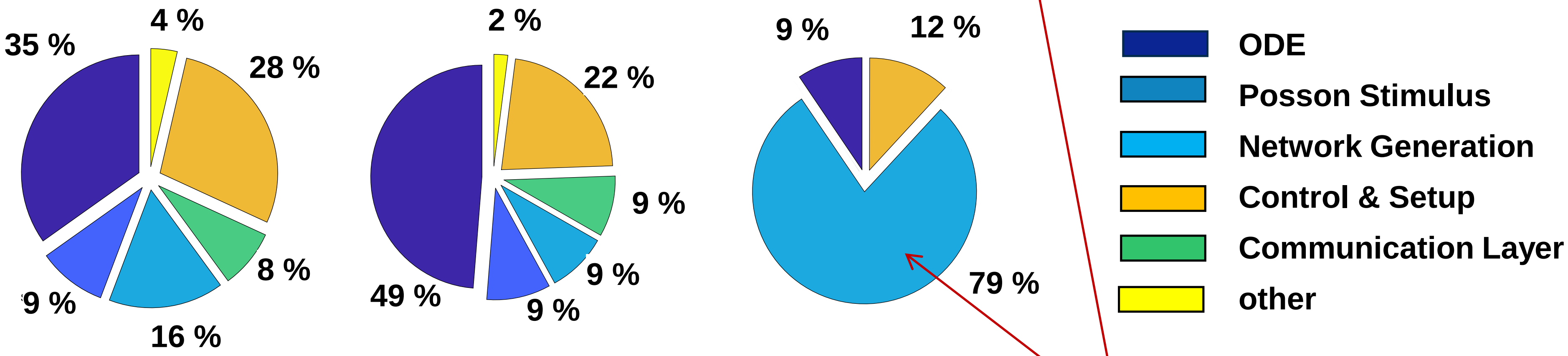


Look-up tables
N_{LUT} = 218.600

Flip-Flops
N_{FF}= 437.200

Block-RAM
N_{BRAM}= 19.2 Mbit

- Circuit resources for CnNN



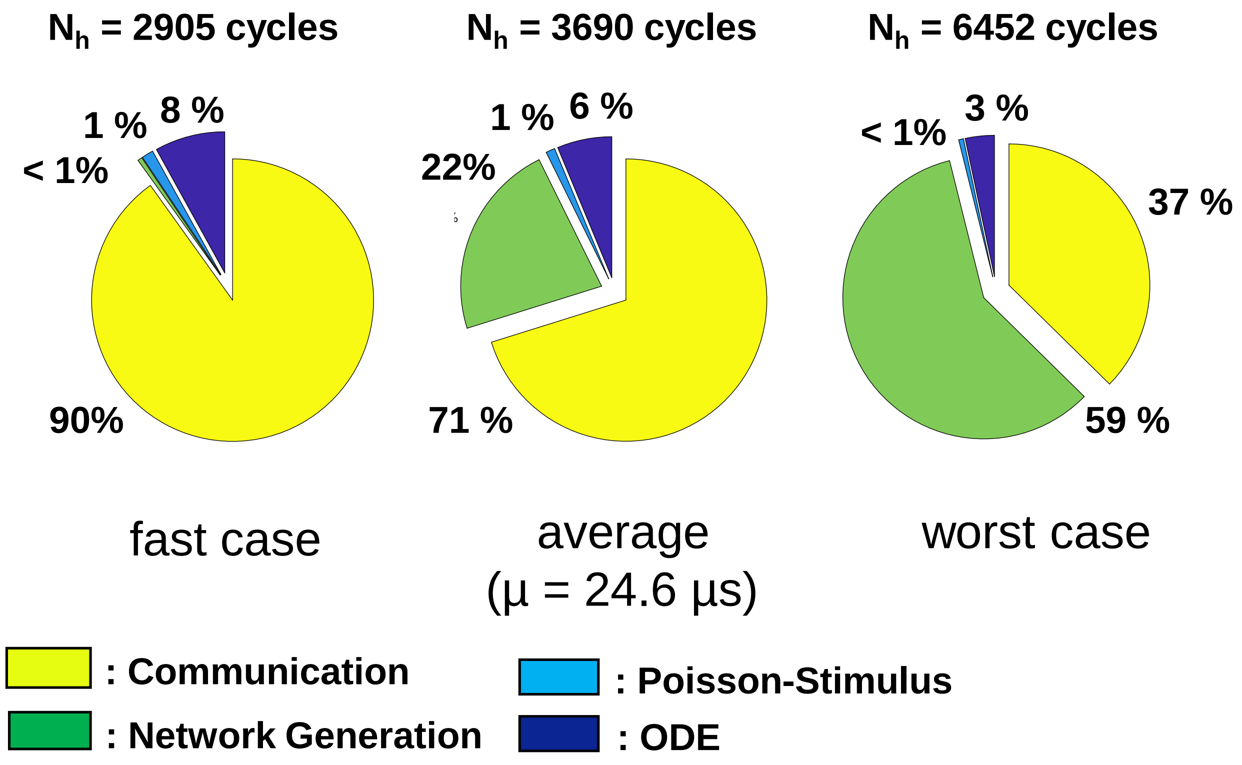
Look-up tables
N_{LUT}= 23.860

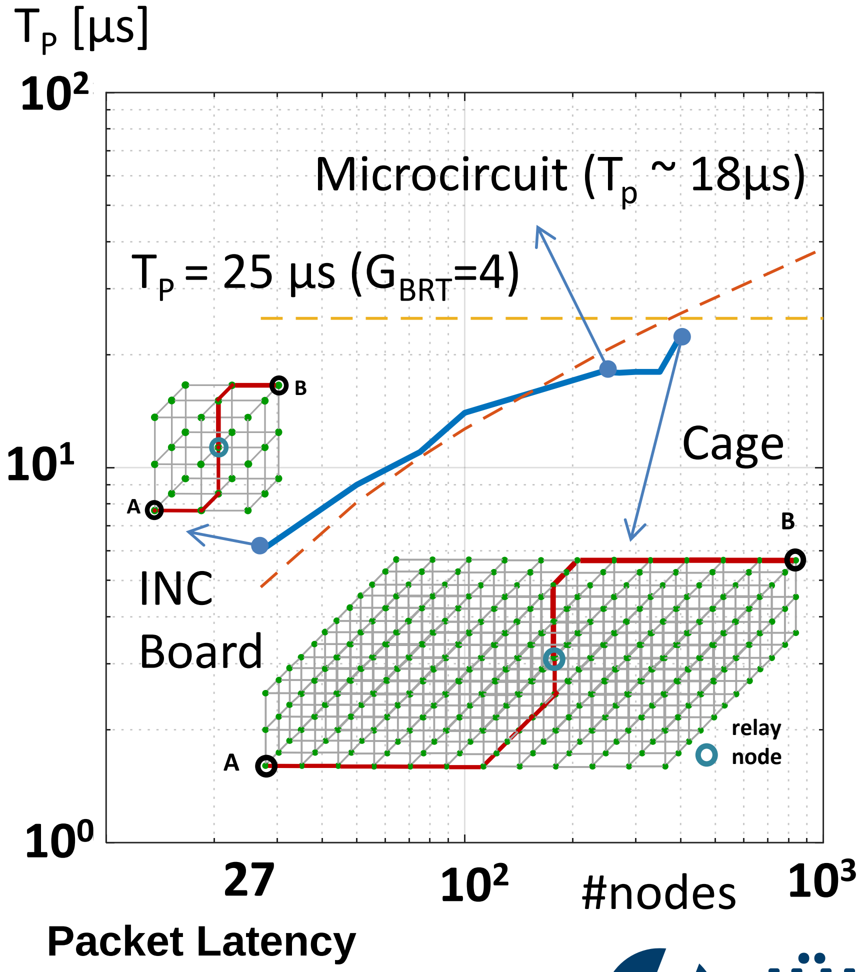
Flip-Flops
N_{FF}= 32.662

Block-RAM
N_{BRAM}= 12 Mbit

- SNNs are memory-dominated

LATENCY-BREAKDOWN : THE MICRO-CIRCUIT

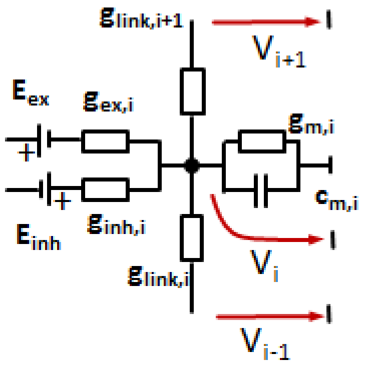




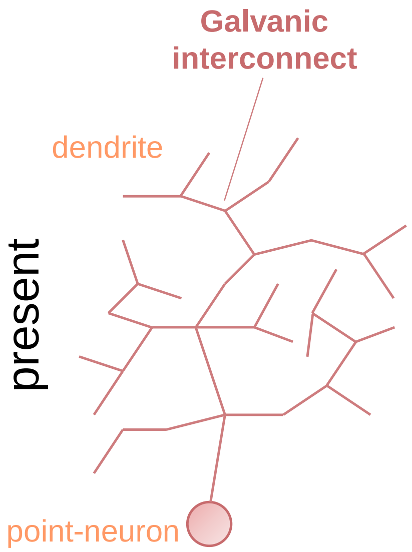
- procedural connectivity: 20 % speedup over ext. DRAM

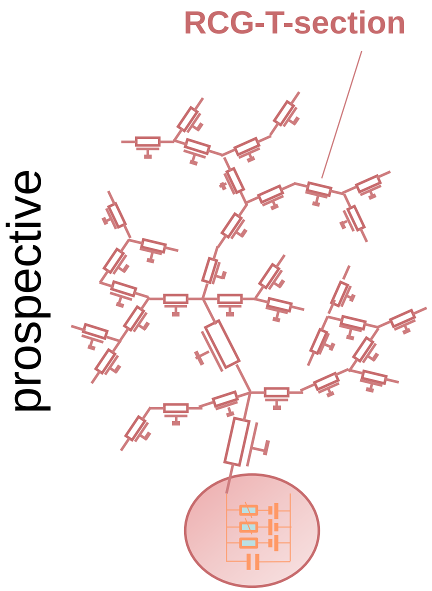
- Speedup X 4.06 over BRT

TOWARDS REALISTIC NEURON MODELS



COBA compartment

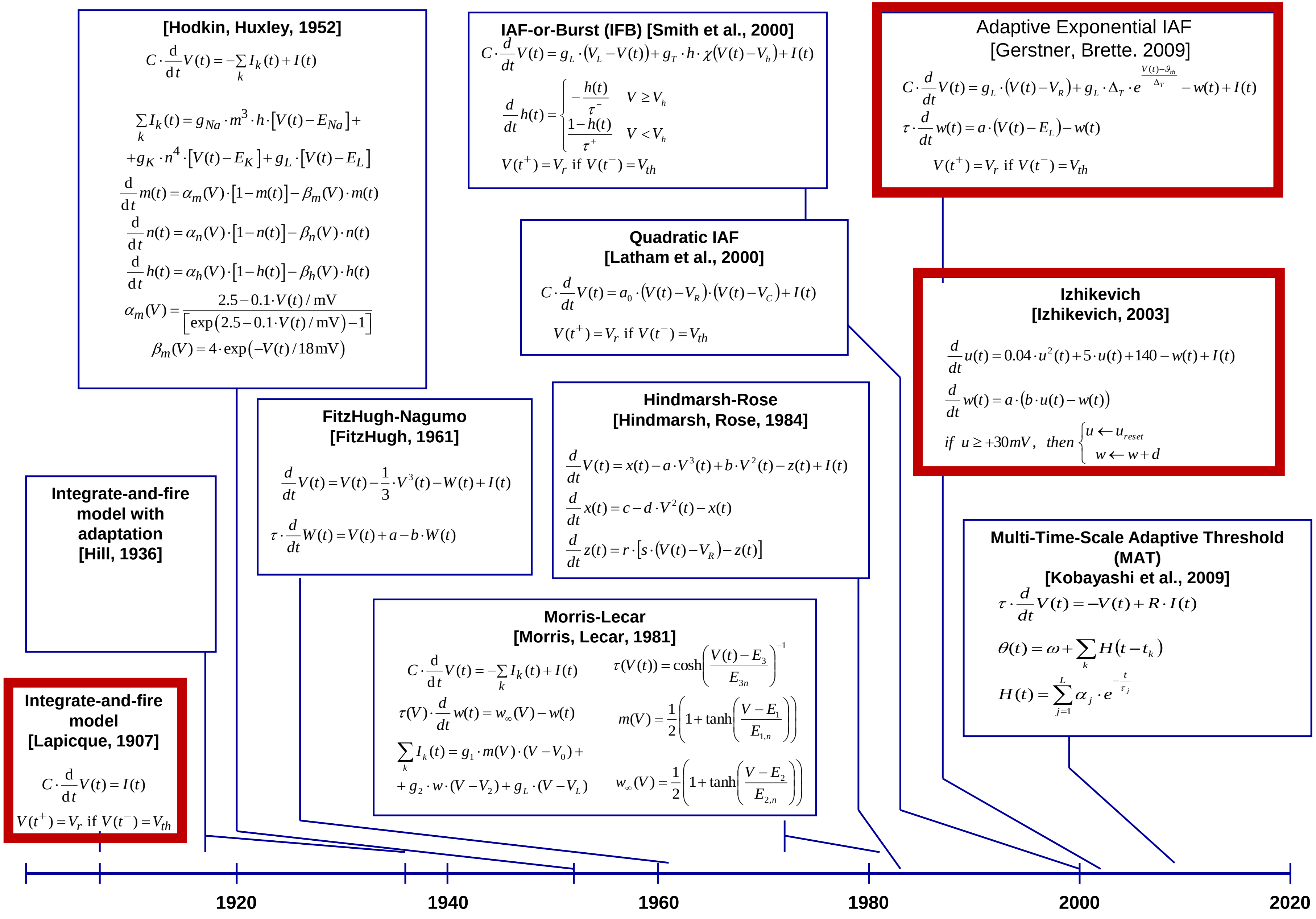




The ‘common case‘	Computational Complexity
• **Linear** point-neuron dynamics • **No dendritic tree** (galvanic interconnect)	• ODE integration by **exponential Euler‘s method** • **High degree of parallelism**
• **Nonlinear** neuron dynamics (e.g. Izhikevich, AdEx, HH) • **compartmental dendritic tree**, model of ‘the dendritic computational toolkit‘ *)	• ODE integration by **advanced numeric methods** • **Limited degree of parallelism** and significant critical paths in arithmetics • particular strategies for **stiff problems**

*) M.London, M.Häusser, Annu.Rev.Neurosci. 2005.28:503-532

TOWARDS REALISTIC NEURON MODELS



Integrate-and-fire model [Lapicque, 1907]

$$C \cdot \frac{\mathrm{d}}{\mathrm{d}t} V(t) = I(t)$$

$$V(t^+) = V_r \text{ if } V(t^-) = V_{th}$$

Integrate-and-fire model with adaptation [Hill, 1936]

[Hodkin, Huxley, 1952]

$$C \cdot \frac{\mathrm{d}}{\mathrm{d}t} V(t) = -\sum_k I_k(t) + I(t)$$

$$\sum_k I_k(t) = g_{Na} \cdot m^3 \cdot h \cdot \left[V(t) - E_{Na}\right] +$$

$$+ g_K \cdot n^4 \cdot \left[V(t) - E_K\right] + g_L \cdot \left[V(t) - E_L\right]$$

$$\frac{\mathrm{d}}{\mathrm{d}t} m(t) = \alpha_m(V) \cdot \left[1 - m(t)\right] - \beta_m(V) \cdot m(t)$$

$$\frac{\mathrm{d}}{\mathrm{d}t} n(t) = \alpha_n(V) \cdot \left[1 - n(t)\right] - \beta_n(V) \cdot n(t)$$

$$\frac{\mathrm{d}}{\mathrm{d}t} h(t) = \alpha_h(V) \cdot \left[1 - h(t)\right] - \beta_h(V) \cdot h(t)$$

$$\alpha_m(V) = \frac{2.5 - 0.1 \cdot V(t)/\mathrm{mV}}{\left[\exp\left(2.5 - 0.1 \cdot V(t)/\mathrm{mV}\right) - 1\right]}$$

$$\beta_m(V) = 4 \cdot \exp\left(-V(t)/18\,\mathrm{mV}\right)$$

FitzHugh-Nagumo [FitzHugh, 1961]

$$\frac{d}{dt} V(t) = V(t) - \frac{1}{3} \cdot V^3(t) - W(t) + I(t)$$

$$\tau \cdot \frac{d}{dt} W(t) = V(t) + a - b \cdot W(t)$$

Morris-Lecar [Morris, Lecar, 1981]

$$C \cdot \frac{\mathrm{d}}{\mathrm{d}t} V(t) = -\sum_k I_k(t) + I(t) \qquad \tau(V(t)) = \cosh\left(\frac{V(t) - E_3}{E_{3n}}\right)^{-1}$$

$$\tau(V) \cdot \frac{d}{dt} w(t) = w_\infty(V) - w(t) \qquad m(V) = \frac{1}{2}\left(1 + \tanh\left(\frac{V - E_1}{E_{1,n}}\right)\right)$$

$$\sum_k I_k(t) = g_1 \cdot m(V) \cdot (V - V_0) +$$

$$+ g_2 \cdot w \cdot (V - V_2) + g_L \cdot (V - V_L) \qquad w_\infty(V) = \frac{1}{2}\left(1 + \tanh\left(\frac{V - E_2}{E_{2,n}}\right)\right)$$

Hindmarsh-Rose [Hindmarsh, Rose, 1984]

$$\frac{d}{dt} V(t) = x(t) - a \cdot V^3(t) + b \cdot V^2(t) - z(t) + I(t)$$

$$\frac{d}{dt} x(t) = c - d \cdot V^2(t) - x(t)$$

$$\frac{d}{dt} z(t) = r \cdot \left[s \cdot \left(V(t) - V_R\right) - z(t)\right]$$

IAF-or-Burst (IFB) [Smith et al., 2000]

$$C \cdot \frac{d}{dt} V(t) = g_L \cdot \left(V_L - V(t)\right) + g_T \cdot h \cdot \chi\left(V(t) - V_h\right) + I(t)$$

$$\frac{d}{dt} h(t) = \begin{cases} -\dfrac{h(t)}{\tau^-} & V \geq V_h \\ \dfrac{1 - h(t)}{\tau^+} & V < V_h \end{cases}$$

$$V(t^+) = V_r \text{ if } V(t^-) = V_{th}$$

Quadratic IAF [Latham et al., 2000]

$$C \cdot \frac{d}{dt} V(t) = a_0 \cdot \left(V(t) - V_R\right) \cdot \left(V(t) - V_C\right) + I(t)$$

$$V(t^+) = V_r \text{ if } V(t^-) = V_{th}$$

Izhikevich [Izhikevich, 2003]

$$\frac{d}{dt} u(t) = 0.04 \cdot u^2(t) + 5 \cdot u(t) + 140 - w(t) + I(t)$$

$$\frac{d}{dt} w(t) = a \cdot \left(b \cdot u(t) - w(t)\right)$$

$$\text{if } \; u \geq +30mV, \;\; \text{then} \begin{cases} u \leftarrow u_{reset} \\ w \leftarrow w + d \end{cases}$$

Adaptive Exponential IAF [Gerstner, Brette. 2009]

$$C \cdot \frac{d}{dt} V(t) = g_L \cdot \left(V(t) - V_R\right) + g_L \cdot \Delta_T \cdot e^{\frac{V(t) - \vartheta_{rh}}{\Delta_T}} - w(t) + I(t)$$

$$\tau \cdot \frac{d}{dt} w(t) = a \cdot \left(V(t) - E_L\right) - w(t)$$

$$V(t^+) = V_r \text{ if } V(t^-) = V_{th}$$

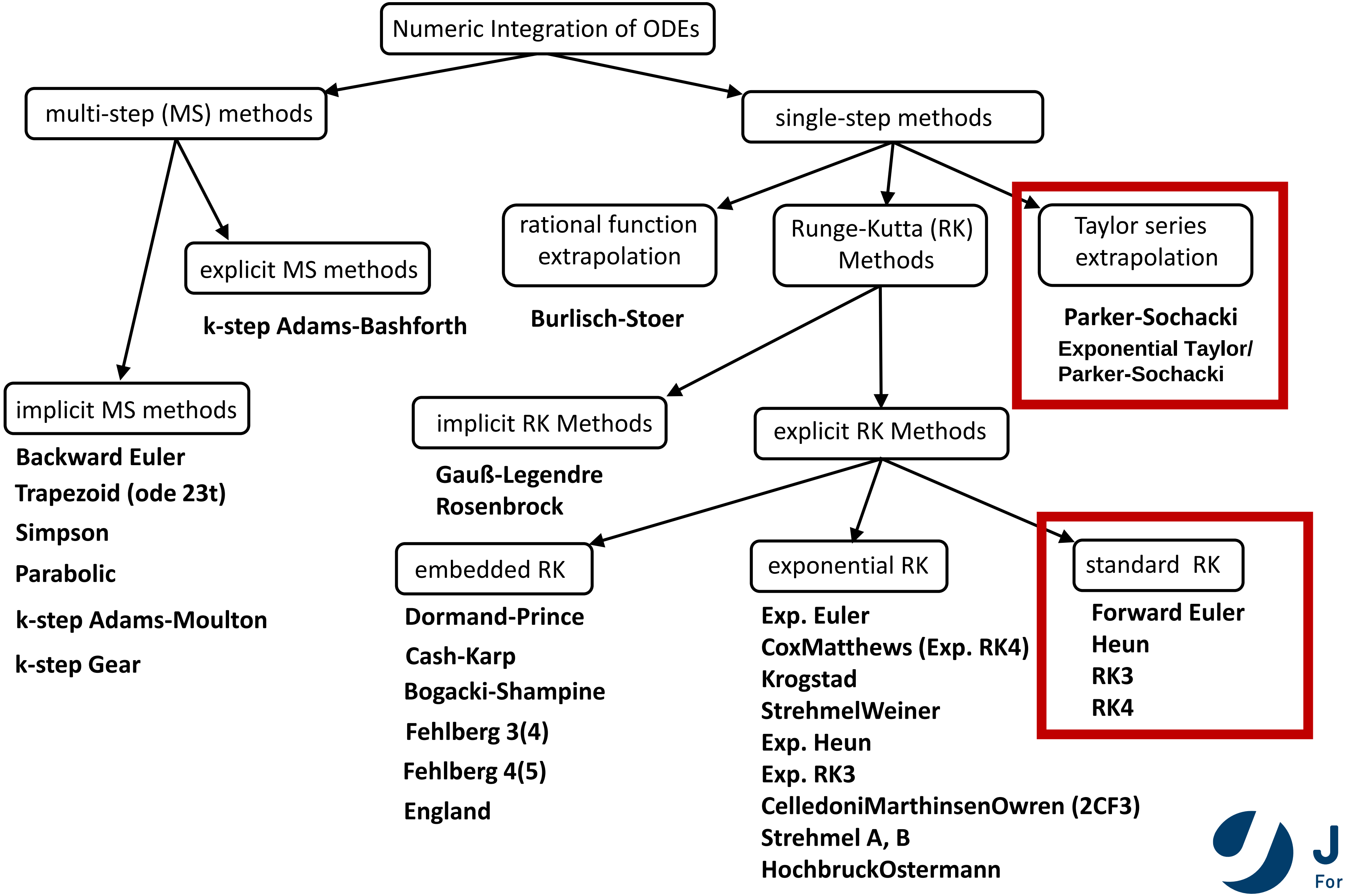
Multi-Time-Scale Adaptive Threshold (MAT) [Kobayashi et al., 2009]

$$\tau \cdot \frac{d}{dt} V(t) = -V(t) + R \cdot I(t)$$

$$\theta(t) = \omega + \sum_k H\left(t - t_k\right)$$

$$H(t) = \sum_{j=1}^{L} \alpha_j \cdot e^{-\frac{t}{\tau_j}}$$

ODE SOLVERS



Numeric Integration of ODEs

multi-step (MS) methods

explicit MS methods

k-step Adams-Bashforth

implicit MS methods

Backward Euler
Trapezoid (ode 23t)
Simpson
Parabolic
k-step Adams-Moulton
k-step Gear

single-step methods

rational function extrapolation

Burlisch-Stoer

Runge-Kutta (RK) Methods

implicit RK Methods

Gauß-Legendre
Rosenbrock

explicit RK Methods

embedded RK

Dormand-Prince
Cash-Karp
Bogacki-Shampine
Fehlberg 3(4)
Fehlberg 4(5)
England

exponential RK

Exp. Euler
CoxMatthews (Exp. RK4)
Krogstad
StrehmelWeiner
Exp. Heun
Exp. RK3
CelledoniMarthinsenOwren (2CF3)
Strehmel A, B
HochbruckOstermann

standard RK

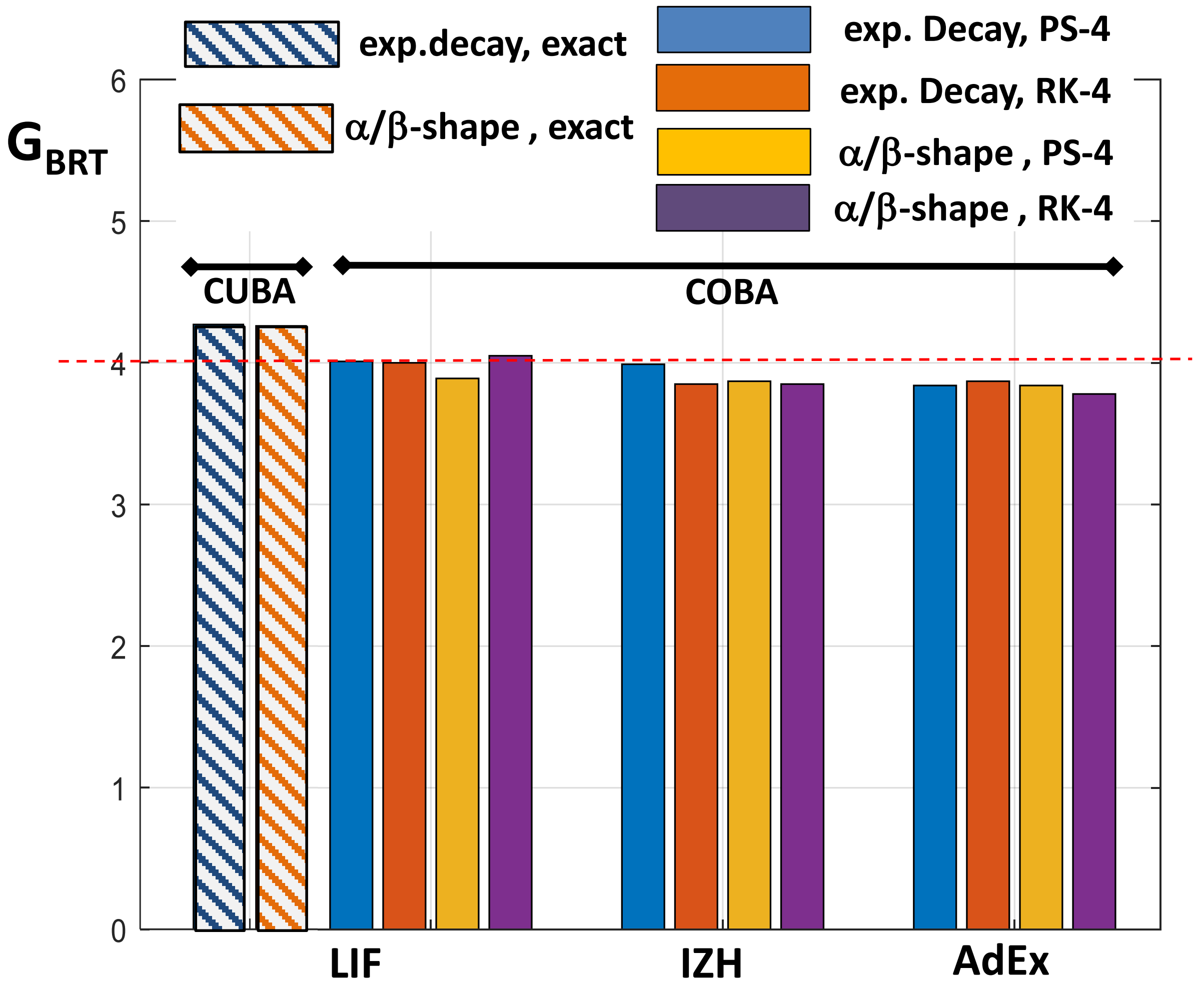
Forward Euler
Heun
RK3
RK4

Taylor series extrapolation

Parker-Sochacki
Exponential Taylor/
Parker-Sochacki

JÜLICH
Forschungszentrum

SPEED-UP FACTOR G_{BRT} FOR VARIOUS MODELS



⇒ Speed-up not limited by arithmetics

Exact Integration (applies for linear ODEs only)

$$V(t+h) = e^{Ah} \cdot V(t)$$

S-step Runge-Kutta Method (RK-S)

$$k_j = f\left(t_n + h \cdot c_j \ ; \ V(t) + h \cdot \sum_{l=1}^{s} a_{jl} \cdot k_l \right) \ , j = 1,\dots,s$$

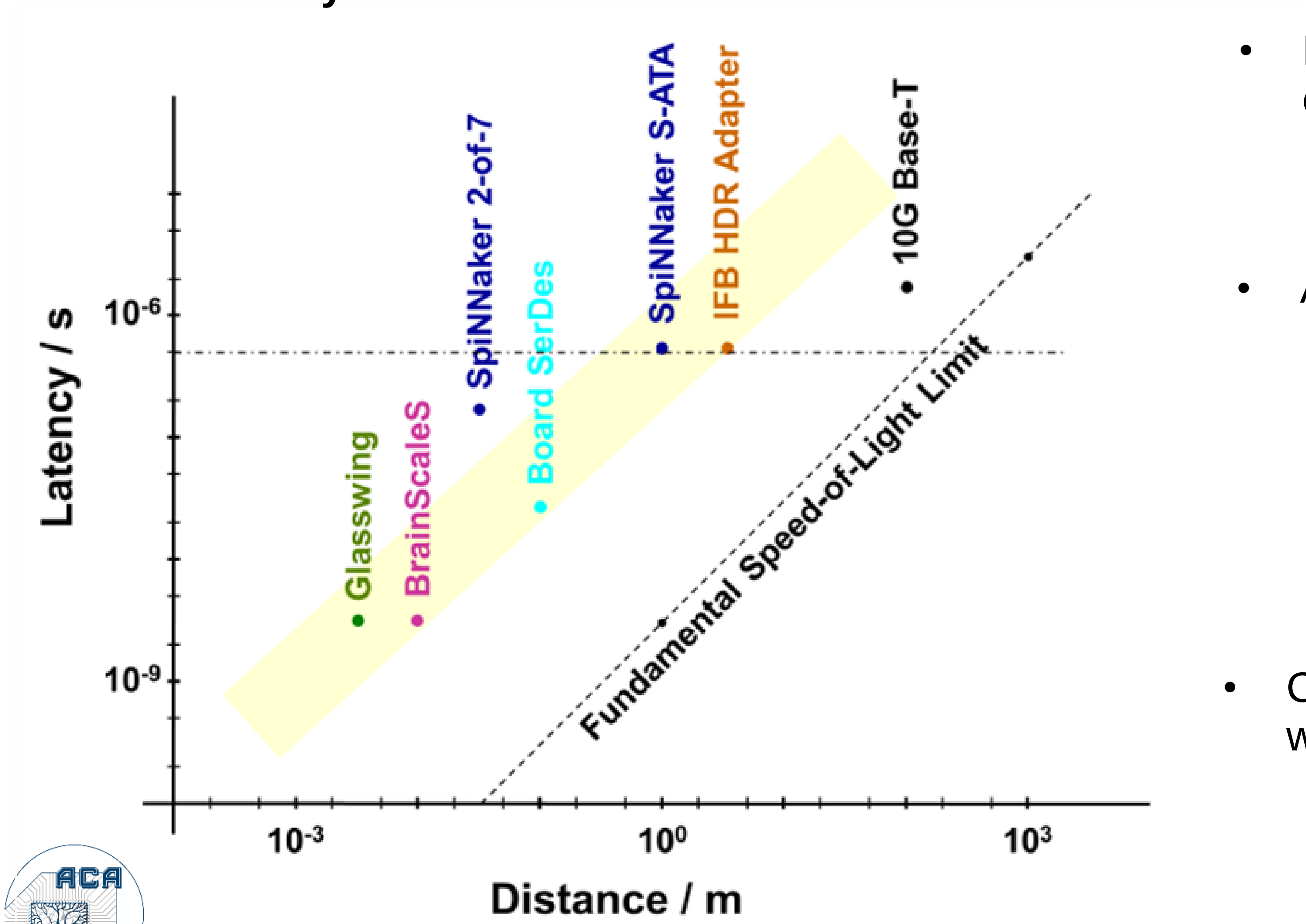
$$V(t+h) = V(t) + h \cdot \sum_{j=1}^{s} b_j \cdot k_j$$

S-step Parker-Sochacki Method (PS-S)

$$V_i(t+h) = V_i(t) + \frac{V_i^{'}(t)}{1!} \cdot h + \frac{V_i^{''}(t)}{2!} \cdot h^2 + \frac{V_i^{'''}(t)}{3!} \cdot h^3 + \dots + \frac{V_i^{(s)}(t)}{s!} \cdot h^s$$

Trends in State-of-the-Art Communication Standards

1-Hub latency



- Latency grows with the distance of compute-nodes

- Avoid large latencies by

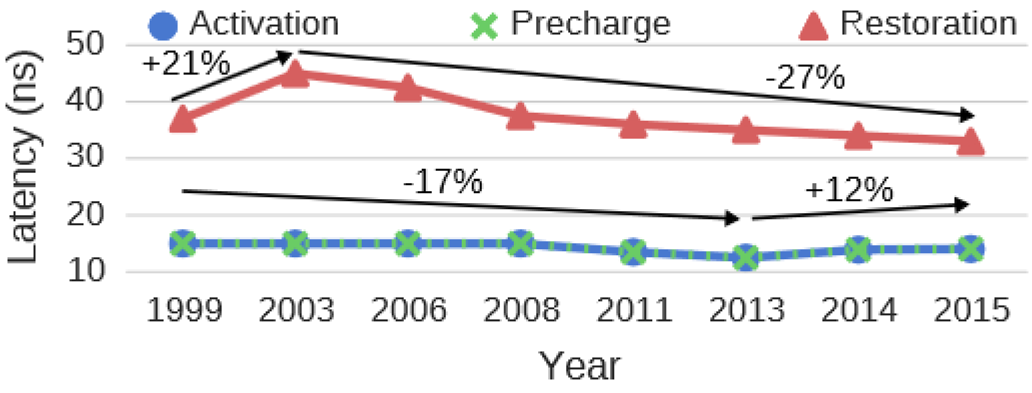
  - few hubs between nodes

  - dense (3D-stacked) circuit integration

- Conceptually, highly integrated components with short physical distance could be required

Trends in State-of-the-Art DRAM Performance

- Latency Trends



K.K.Change et al., „Understanding Latency Variation in modern DRAM Chips: Experimental Characterization, Analysis, and Optimization“, doi:10.1145/2896377.2901453 (2016)

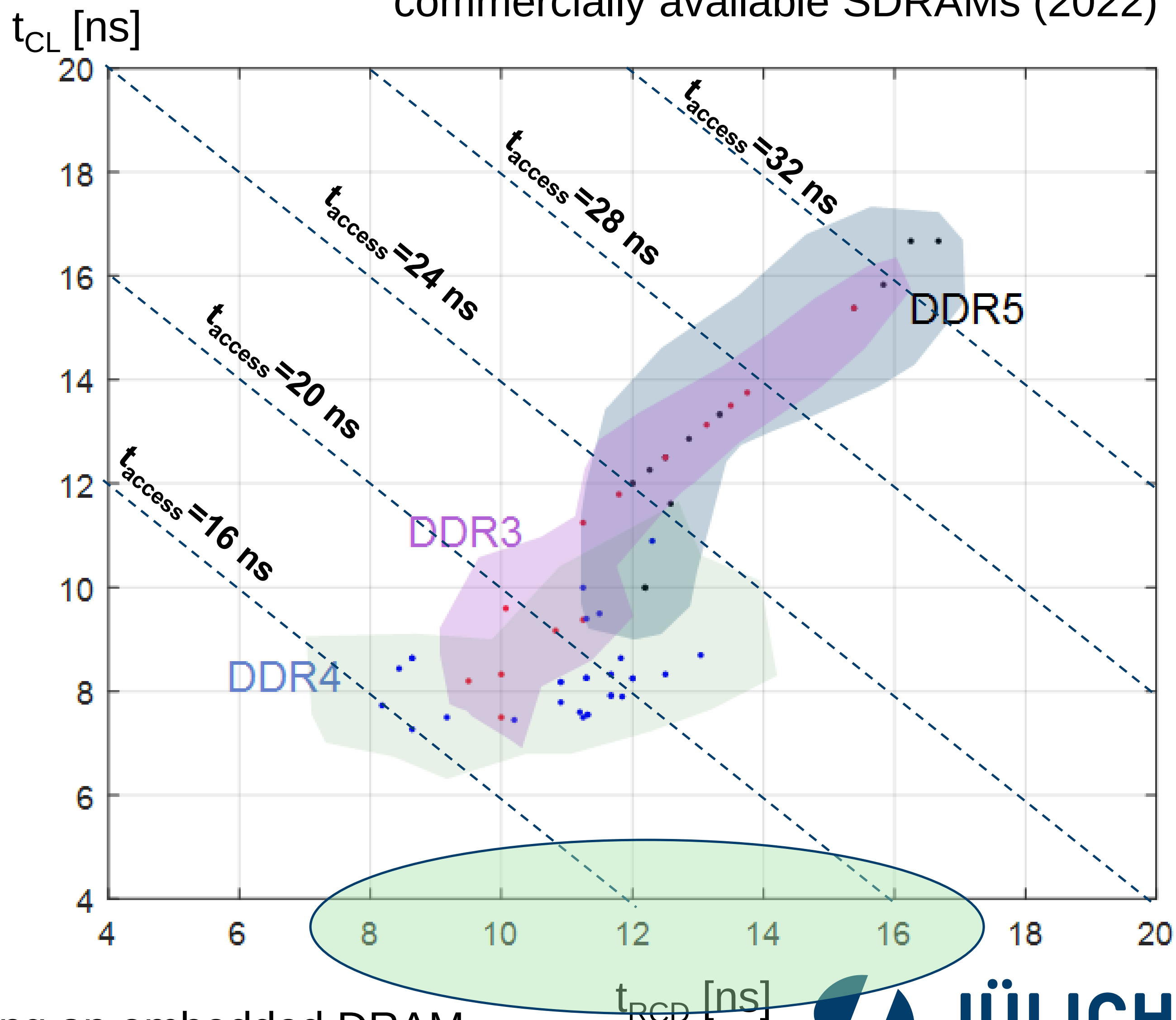
- No significant trend in access latency reduction

- Neuromorphic Computing:

  - **Plasticity** will incur significant traffic to the memory system

  - latency issue could be solved e.g. by using an embedded DRAM technology in a dedicated accelerator circuit

- Performance data from 54 different commercially available SDRAMs (2022)



CONCLUSION

- Components in conventional HPC systems are optimized for both:

large data packets and **large communication & memory bandwidth**

→ results in **large latencies** for …. memory accesses and packet transmission

- Requirements for a future accelerated neuromorphic compute platform

+ communication of **data packets** comprising **small size** (spikes)

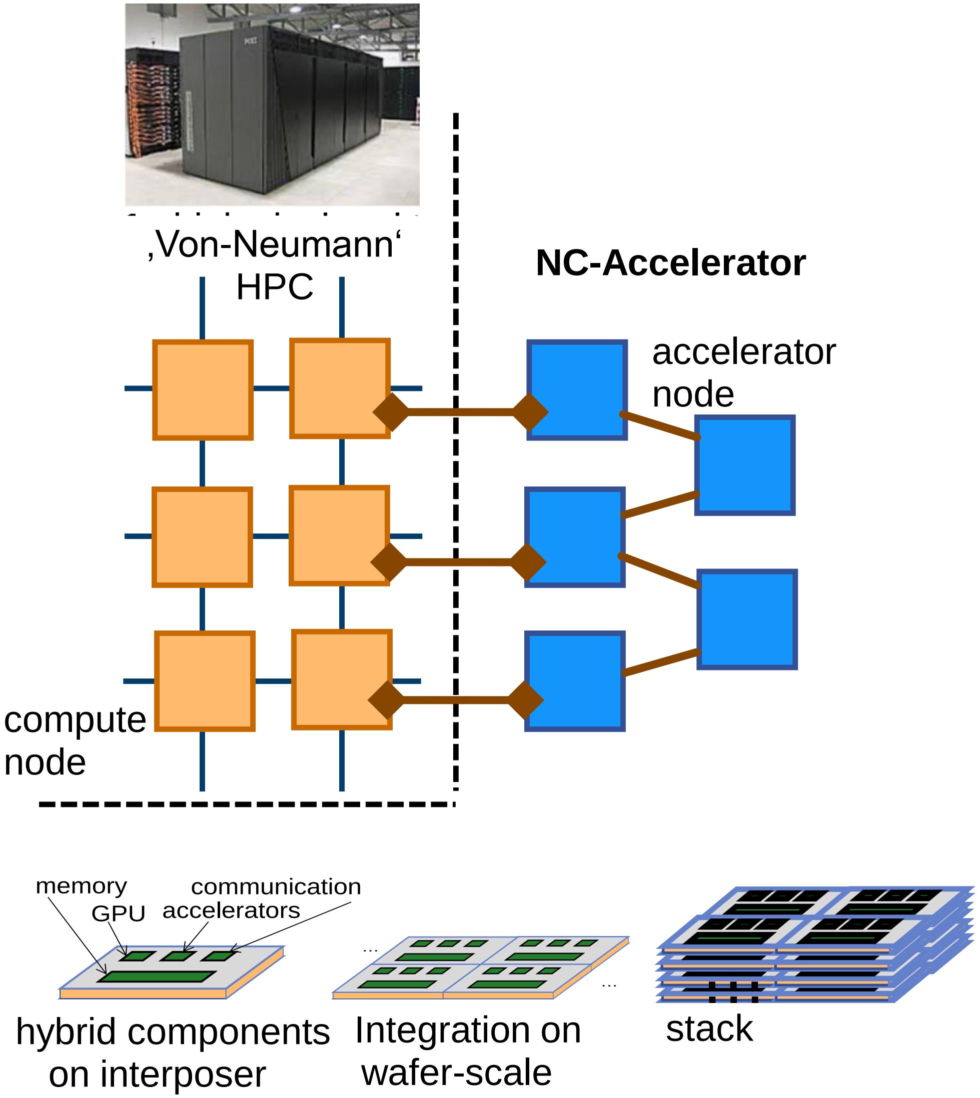
+ fully random memory access to **small data packets** (e.g. synapse parameters) with very restricted locality properties (caches won‘t work efficiently)

+ **ultra-short latencies for communication and memory access**

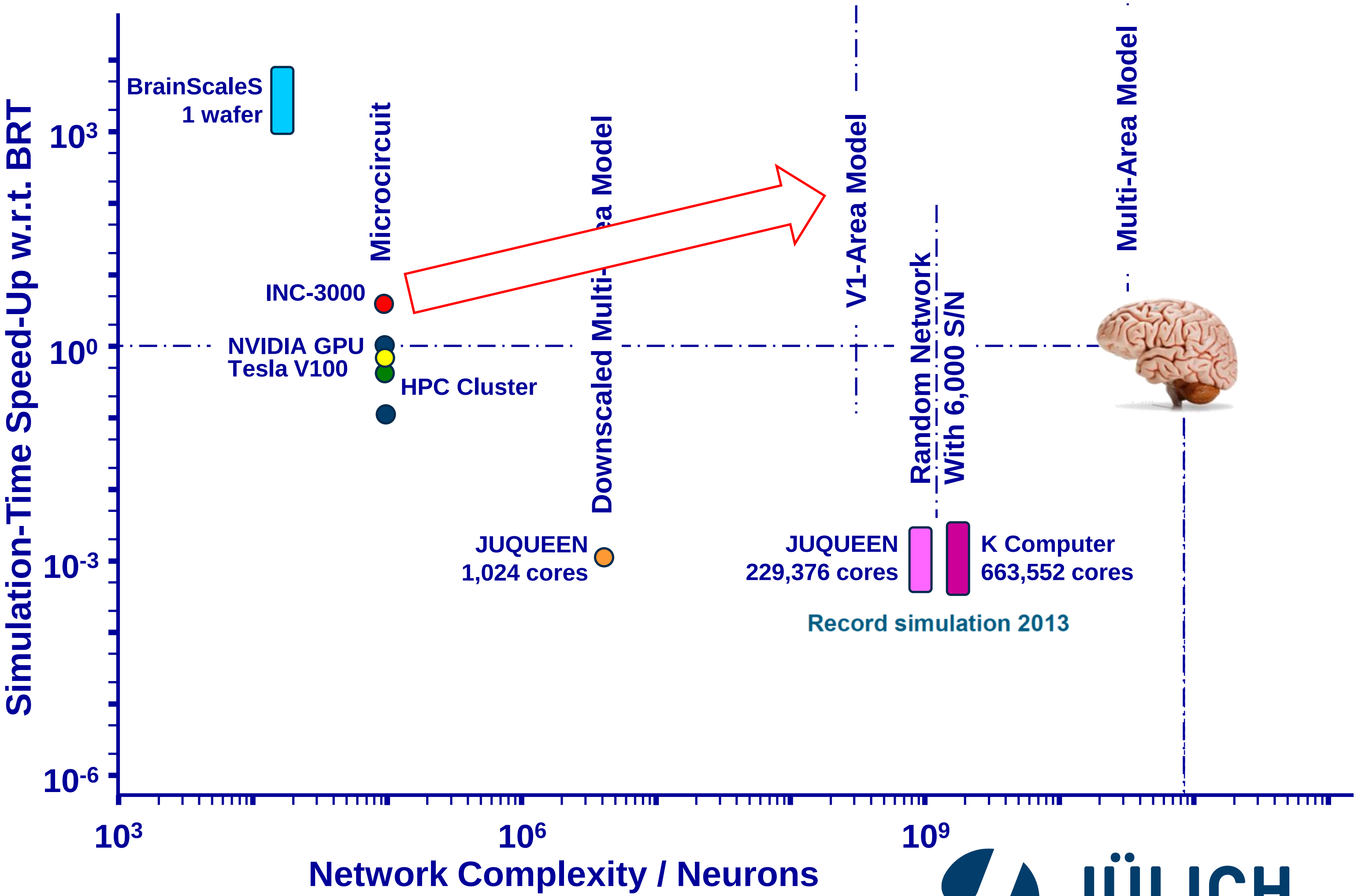
The requirements for future NC-Platform and HPC are complementary

- New memory architecture *: near-memory computation with short memory latencies*
- Hierarchical networks with ultra-short communication latency
- *High-package density (2.5D/3D stacked silicon on interposer)*

THE LONG-TERM GOAL: SUSTAINABLY STAY AHEAD



State-of-the-Art Simulation Time (0.1-ms time grid)



„Hybrid Neuromorphic-von-Neumann general purpose computing“

JÜLICH
Forschungszentrum