Leveraging Sparsity of SRNNs for Reconfigurable and Resource-Efficient Network-on-Chip

Manu Rathore

and Garrett S. Rose

The University of Tennessee, Knoxville

NICE'2024

04.24.24

TENN LAB NEUROMORPHIC ARCHITECTURES. LEARNING. APPLICATIONS.

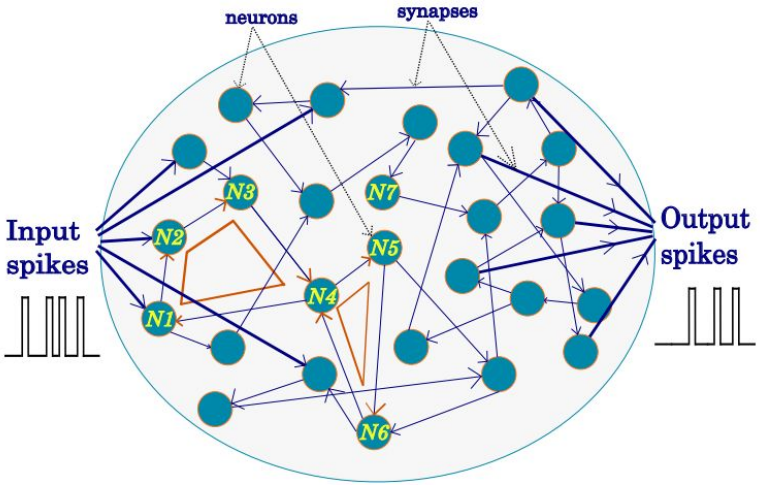
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Overview

- Introduction
- Motivation
- Proposed NoC Architecture
- Leveraging Sparsity using the NoC
- Implementation and Results
- Conclusion

Introduction

- Spiking Recurrent Neural Network (SRNN)
  - resource-efficient low-power solution
    - Less neurons can achieve high computational performance
  - nonlinear signal processing and control applications
- Utilizing analog timing information of spike data
- Reconfigurable implementation on hardware challenging
  - Network-on-Chip (NoC) to support reconfigurable connectivity

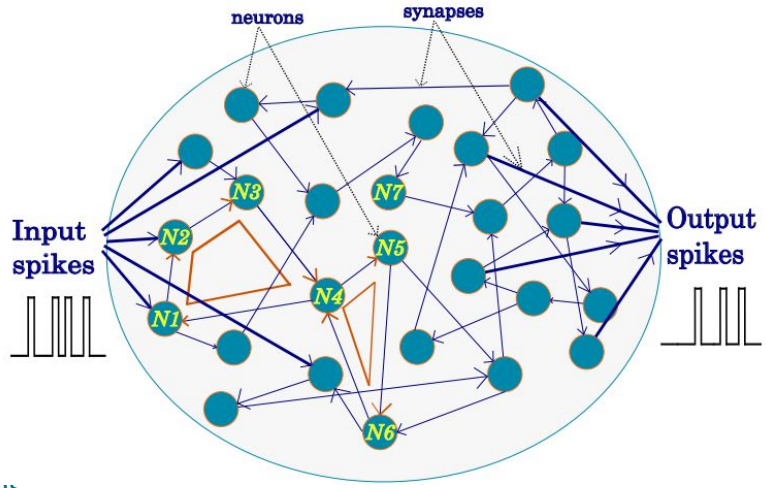


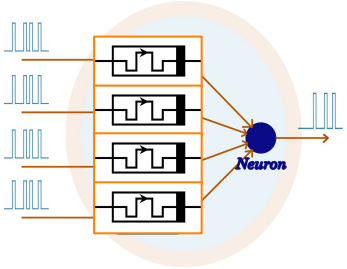
Building block:

Neuron Core with multiple synapses

Introduction

- Reconfigurable NoC requirements:
  - ✓ Connections between any two arbitrary neurons
  - ✓ Low-power and area-efficient
  - ✓ High fidelity and minimal degradation
  - ✓ Preserving spike timing and synchronization information
  - ✓ Scalable
  - ✓ Flexibility of design space to include multiple viable routing paths
- NoC be circuit-switched or packet-switched
- Constrained by physical wiring limits



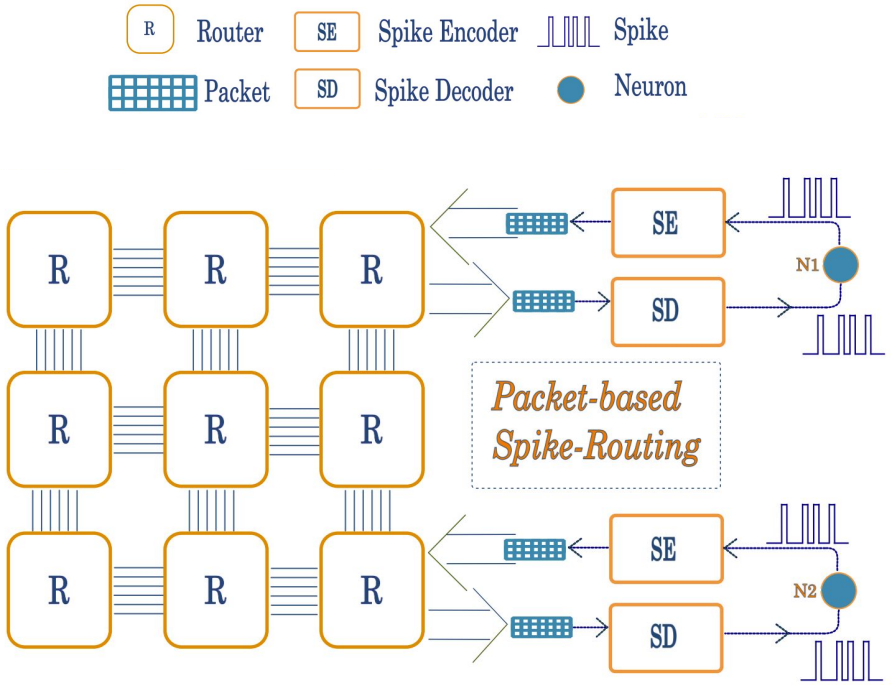


Building block:

Neuron Core with multiple synapses

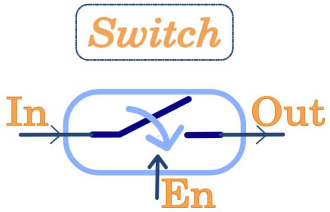
Introduction

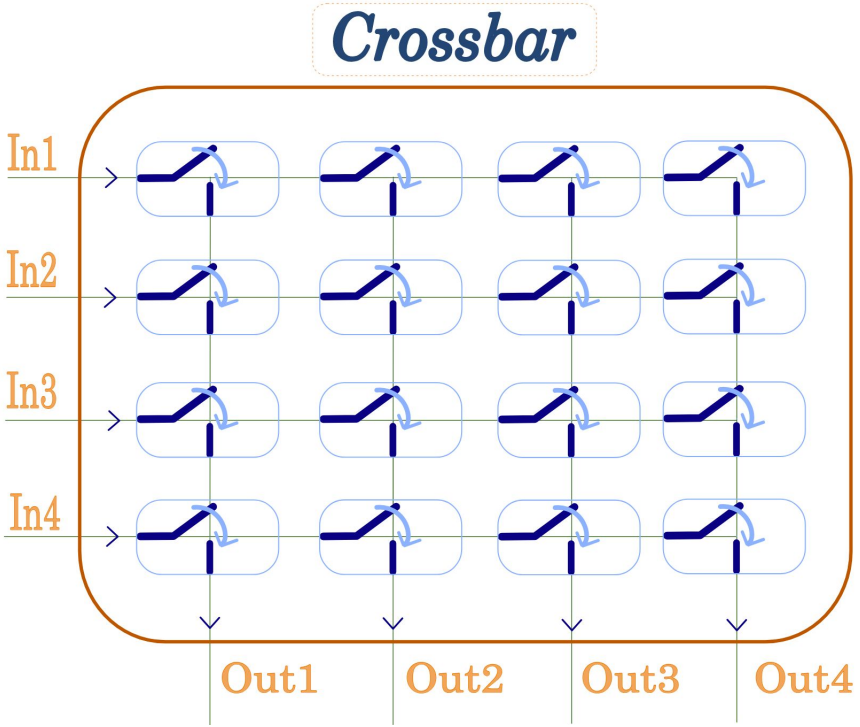
- Existing Solutions : multi-bit packet-based
- While scalable, added performance overhead for proximal neurons
  - Spike encoder/decoder
  - Routers
  - Extra wires for addresses
  - Advantage of spiking low-power data compromised



Introduction

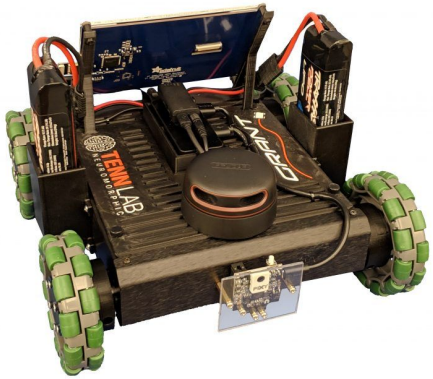
- Existing Solutions : Circuit-switched crossbars
- Simple and easy to implement, but
  - Only one possible datapath between a set of input-output
  - Scales multiplicatively
    - No. of switches = input x output

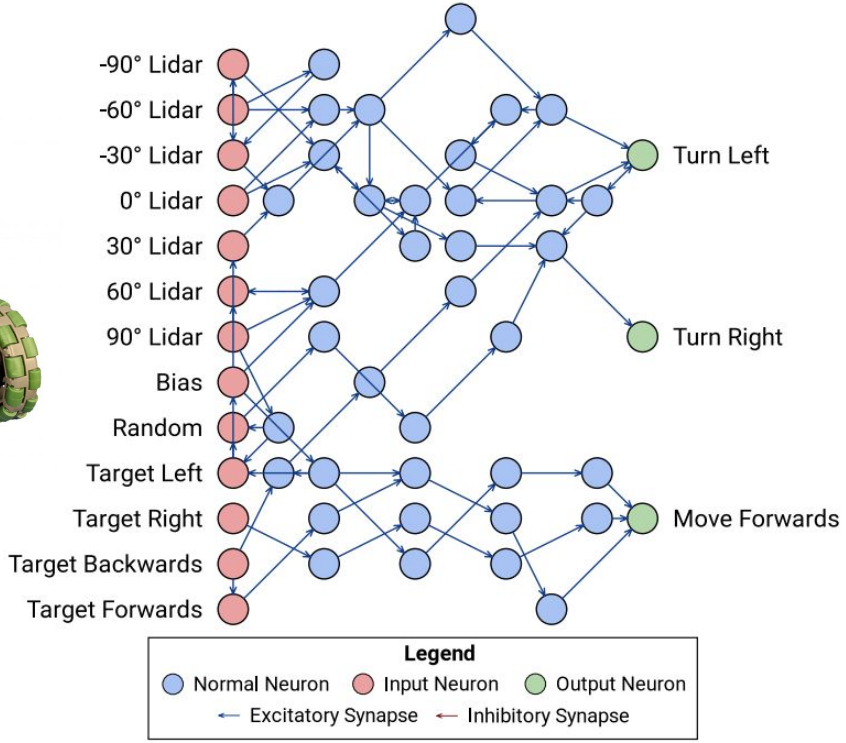




Motivation

- Network structures that inform NoC architecture
- Neuromorphic Targeter[1]
- Small network
  - Packet-based approach unnecessary
- Sparse Connectivity
  - Crossbar NoC assumes fully-concurrently connected network

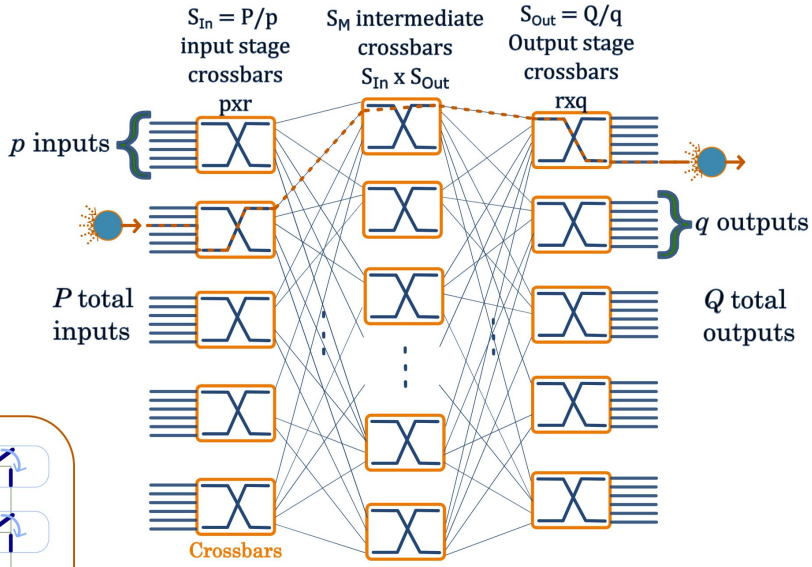


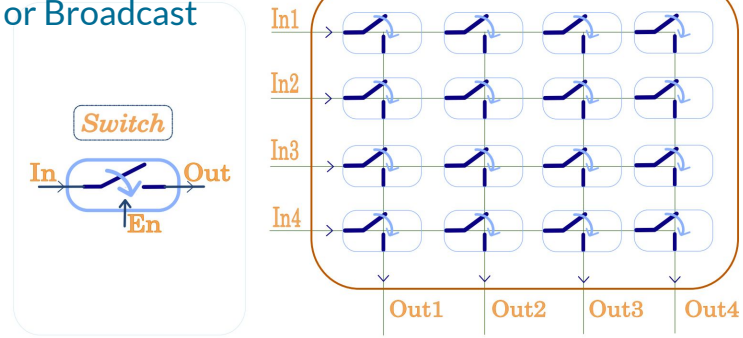


[1] - Jonathan D. Ambrose et al. “GRANT: Ground-Roaming Autonomous Neuromorphic Targeter”. IJCNN 2020

Proposed NoC

- Spike-routing Circuit-Switched Network-on-chip
- SpiCS-Net: “spikes-as-spikes”
- Circuit-Switching : establish direct wired connection
- Clos topology[1] : multiple smaller crossbars; achieve full connectivity with less switches
- Unicast, Multicast or Broadcast
- Delay agnostic





[1] - Charles Clos “A study of non-blocking switching networks”. BSTJ 1953

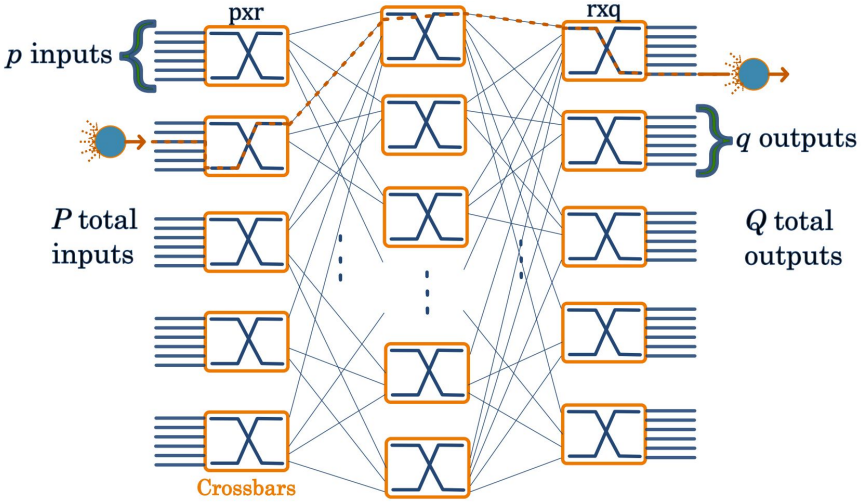
SpiCS-Net

Problem of connecting P inputs to Q outputs ➡ Problem of connecting p inputs to q outputs

Conditions on SpiCS-Net parameters for multicast

non-blocking implementation,

$$r <= p + q - 1, \quad P/p < Q/q, \quad r > p, \quad r > q$$



SpiCS-Net Design for specific SRNN size

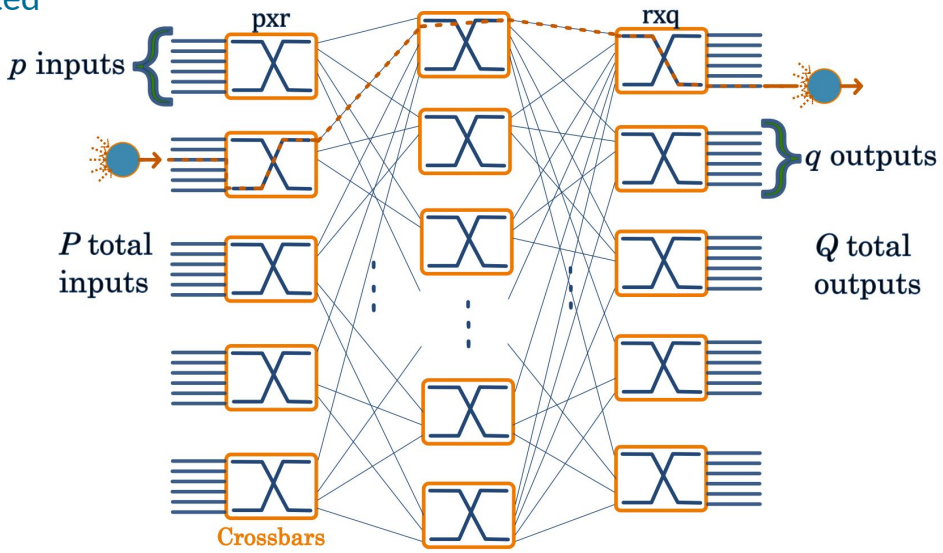
N_n = number of neurons

C_{max} = Maximum number of connections for fully-connected SRNN

$$C_{max} = 2 * \binom{N_n}{2} = N_n(N_n - 1)$$

Limiting by fan-in and fan-out per neuron

$$C_{max} = N_n * max(S_{in}, S_{out})$$

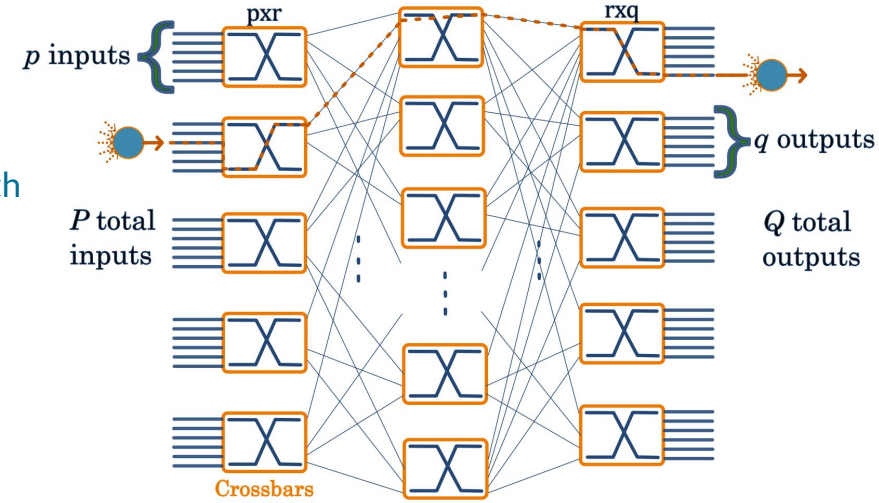


SpiCS-Net Design for specific SRNN size

The maximum possible unicast connections through the NoC are given by,

$$C_{max} = max(P, Q)$$

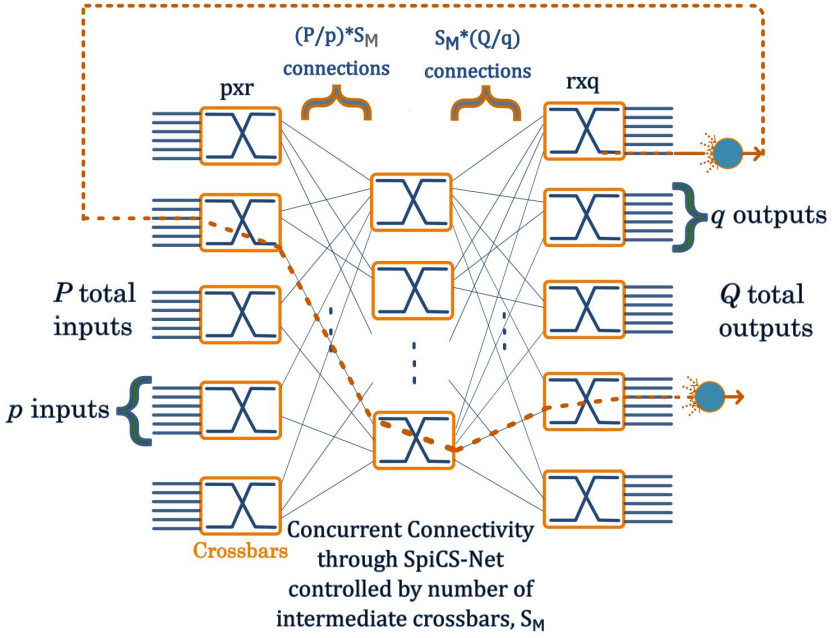
This can be related back to C_{max} for specific sized SRNN, with neuron core fan-in and fan-out.



Manu Rathore and Garrett S. Rose “SpiCS-Net : Circuit Switched Network on Chip for Area-Efficient Spiking Recurrent Neural Networks”. IEEE VLSID 2024

Leverage Network Sparsity using SpiCS-Net

- Concurrent Connectivity: Max connections established simultaneously
- Tuning Concurrent Connectivity by modifying number of crossbars in middle stage
- Without affecting the capability of connection between any set of neurons



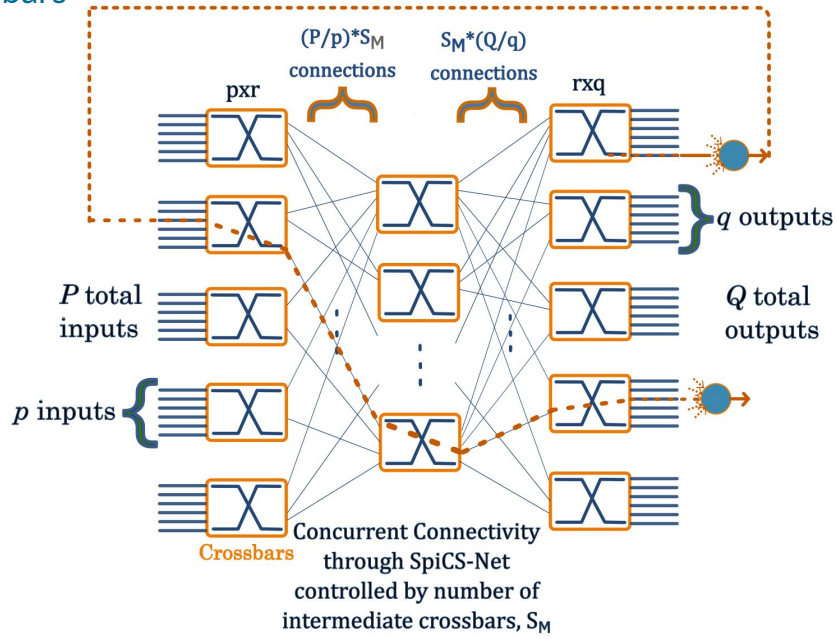
Leverage Network Sparsity using SpiCS-Net

Concurrent Connectivity for S_M *bar* crossbars instead of S_M crossbars

$$CC_{\bar{S}_M} = C_{max_{nw}} - max\left(\frac{P}{p}(S_M - \bar{S}_M), \frac{Q}{q}(S_M - \bar{S}_M)\right)$$

Concurrent Connectivity percentage,

$$CC_{percent} = \left(\frac{CC_{\bar{S}_M}}{C_{max_{srnn}}}\right) * 100$$

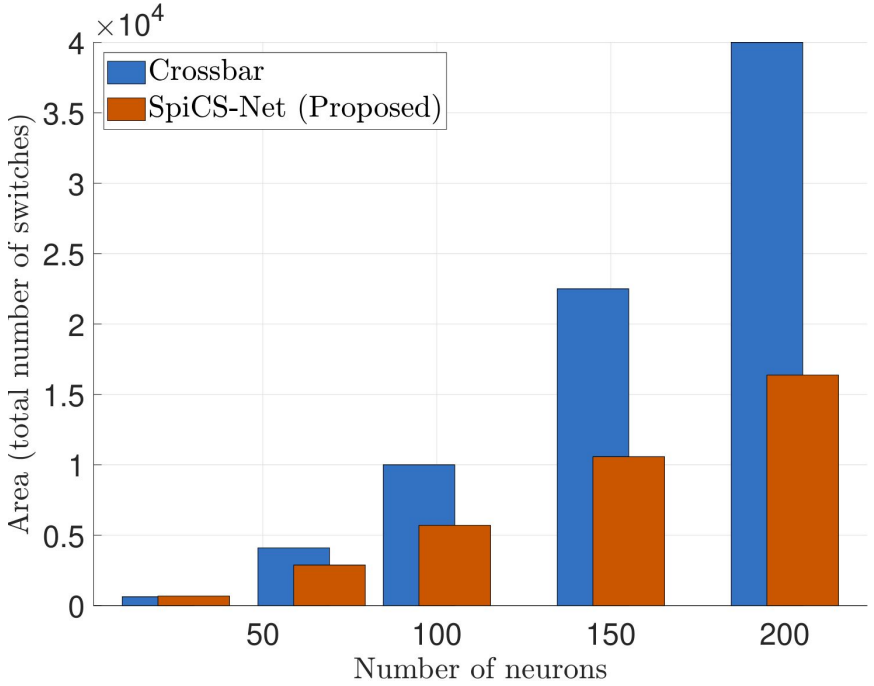


Main ideas

- Circuit-Switched Clos Network
  - Area and Power efficient when compared to packet-based approaches for proximal neurons
- Leveraging Sparseness
  - Blocking property of Clos topology for further area/power savings on-chip
  - Not all connections that are supported by the NoC need to be established at the same time

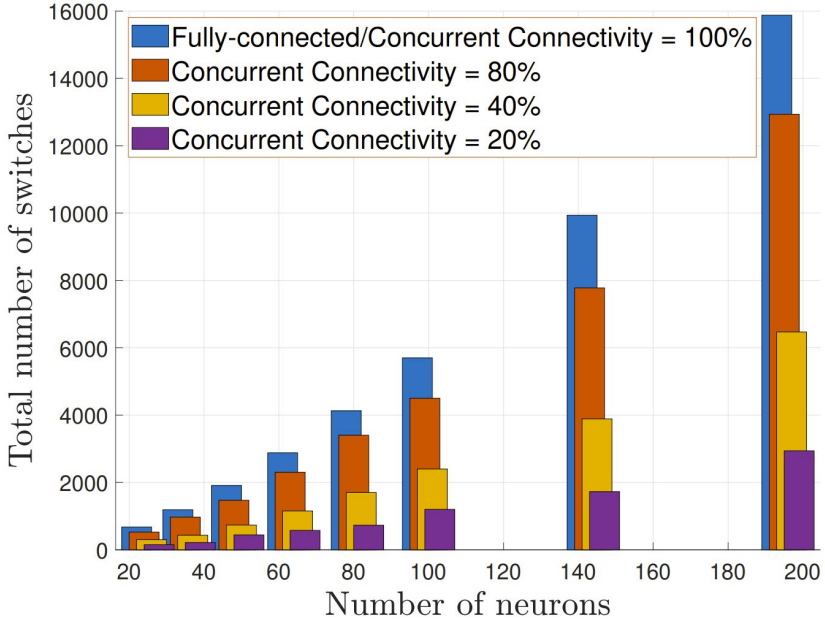
SpiCS-Net Implementation

- Switch can be designed to handle digital/analog spikes
- Digital Implementation details:
  - MUX based implementation in System Verilog
  - Synthesized to 65 nm IBM CMOS10LPE
  - Results from post-layout simulations

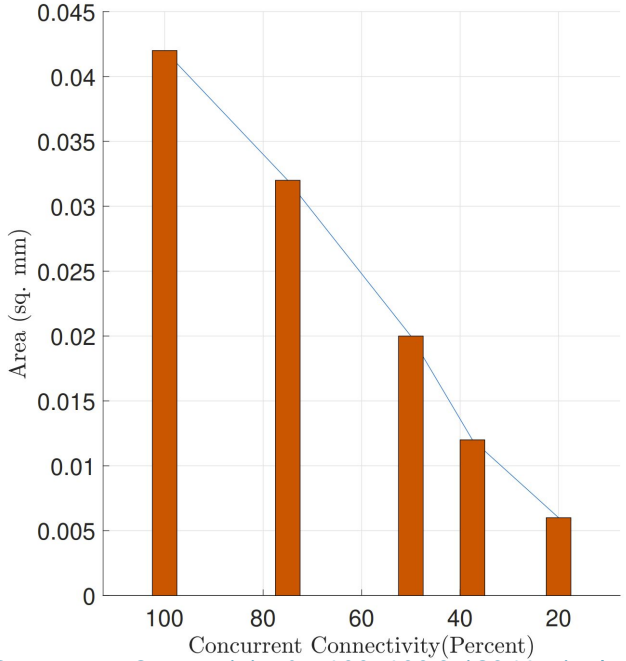


Leverage Network Sparsity using SpiCS-Net

- Number of switches relates directly to area, power and memory requirements on the chip



Switches vs SRNN size for various levels of Concurrent Connectivity



Area vs Concurrent Connectivity for 128x128 SpiCS-Net in the 65nm process

Leverage Network Sparsity using SpiCS-Net

- Area on-chip for varying Concurrent Connectivity comparison with packet-based Network-on-Chip Architectures

Design	Area(mm^2)	Relative Area
SpiCS-Net* 128x128 $CC_{percent}$= 37.5%	0.012	1x
SpiCS-Net* 128x128 $CC_{percent}$= 75%	0.032	2.6x
SpiCS-Net* 128x128 $CC_{percent}$= 100% (non-blocking)	0.042	3.5x
SpiCS-Net* 128x128 (strictly non-blocking)	0.08	6.6x
ClosNN 128x128 (45nm) [1]	0.904	75x
H-NoC 400 neurons (65nm) [2]	0.587	15.65x (scaled)

* SpiCS-Net (65nm)

[1] Hojabr et a. IEEE Transactions on Computers 2017
[2] Carrillo et al. IEEE Transactions on Parallel and Distributed Systems 2012

Results

	3DNoC-SNN[1]	ClosRNN [2]	H-NoC [3]	Crossbar	SpiCS-Net (this work)
Technology	45 nm	45nm	65 nm	65 nm	65 nm
Size	3x3x4	128x128	400 neurons	128x128	128x128
Switching Technique	Packet	Packet	Packet	Circuit	Circuit
Packet Size	31-bit	32 bit	48 bit	1-bit	1-bit
Structure	3D Packet-based	Clos Packet-based	Hierarchical Star-Mesh	Crossbar	Clos Circuit-switched
Area (sq. mm)	0.031 (per Router)	0.904	0.587	0.2	0.042
Power Consumption	10.13 mW (Inverted Pendulum)	0.85 mW (ECG)	13.16 mW (Wisconsin)	-	7.5 uW (Mackey-Glass: Reservoir Computer)
Throughput	0.0313 spike/node/cycle	-	3.3×10^9	4.7×10^9	3.6×10^9

[1] Vu et al. JETC 2019
[2] Hojabr et a. IEEE Transactions on Computers 2017
[3] Carrillo et al. IEEE Transactions on Parallel and Distributed Systems 2012

Results

- Up to 4.5x savings in area compared to packet-based NoCs with Nonblocking Connectivity
  - No packet-handling circuit overhead
- Up to 6.3x savings in area compared to packet-based NoCs with 75% Concurrent Connectivity
  - Leverage sparsity
- 9% higher throughput
  - Spikes transmitted per second
- Substantial savings in power compared to packet-based approaches in literature
  - Fully- Combinational NoC without any switching activity other than the spike itself
  - No dynamic clock-associated power

Conclusion

- Spiking Recurrent Neural Networks offer high computation power even with less neurons owing to recurrent connectivity
- Reconfigurable NoC for these systems pose unique challenges
- Proposed SpiCS-Net architecture is highly efficient circuit-switched and delay agnostic approach for proximal neurons
- Dedicated wires : no spike-collision or loss concerns
- Can be tailored for analog and digital spikes


Thank you!