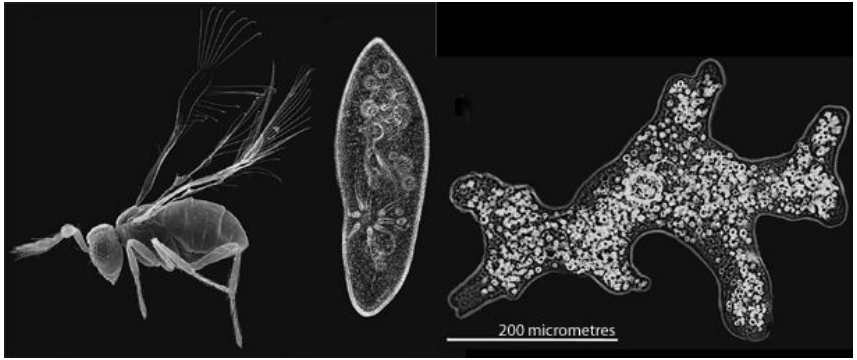
GridNorth

A Spiking Neural Network Chip

Bryan Ingwersen, Jake Leporte, Martin Dawson, Patrick Harkins
CSE 40462 Final Presentation
December 10th, 2021

Inspiration

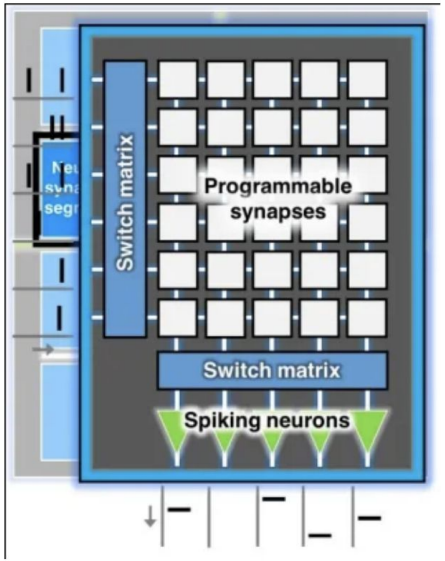


- Fairy Wasp - an organism only 200 micrometers (.2 mm) long that hosts only 7,400 neurons
- The neurons do not communicate via electrical signals as most neurons do, they are connected directly to each other

Intel - “Guided by the principles of biological neural computation, neuromorphic computing intentionally departs from the familiar algorithms and programming abstractions of conventional computing so it can unlock orders of magnitude gains in efficiency and performance compared to conventional architectures.”

Existing Implementations

- IBM TrueNorth
- Intel Loihi 2
- SpiNNaker
- Innatera



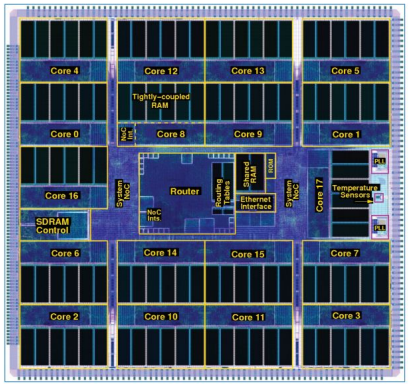
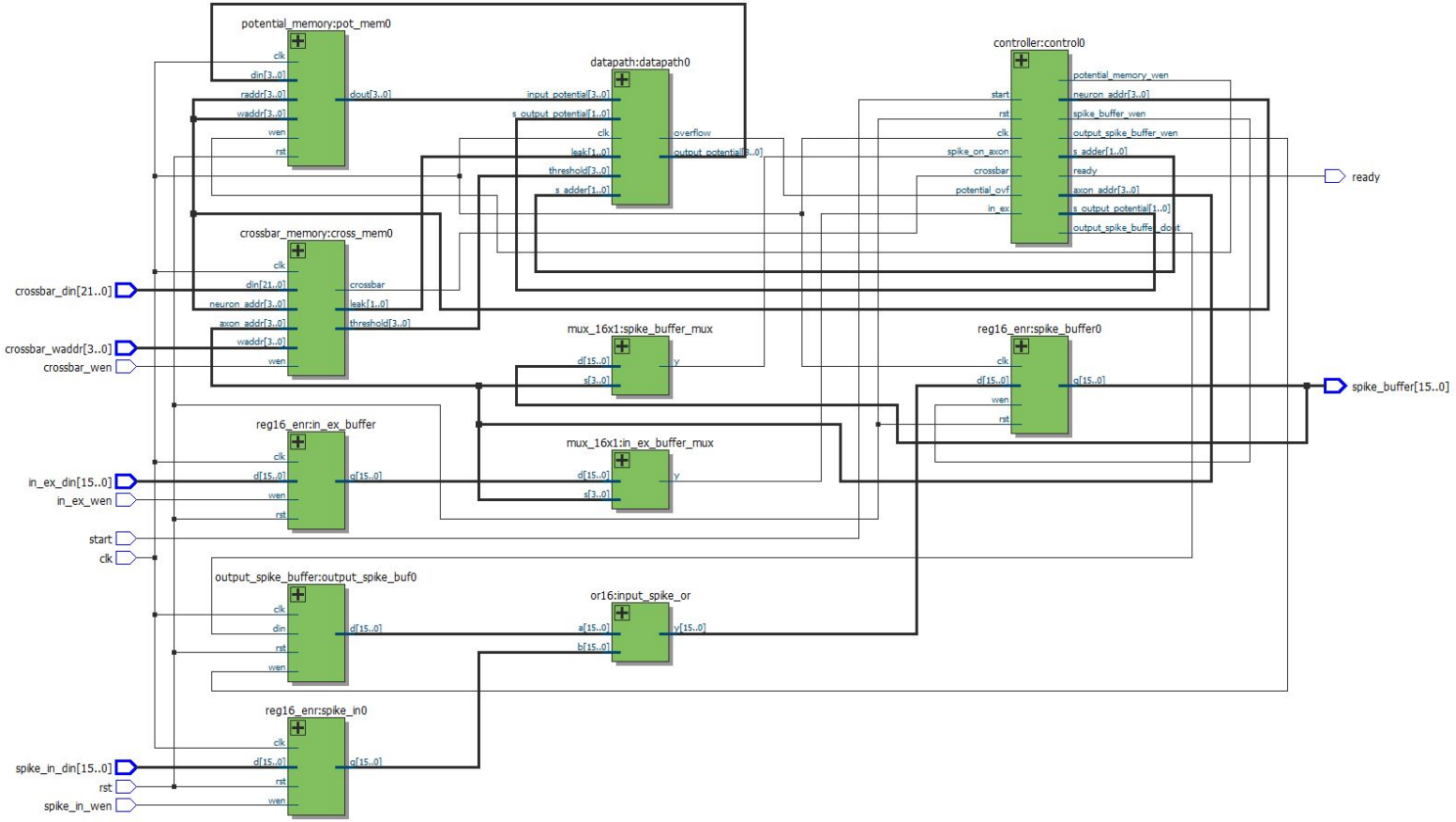


Figure 2.8. The SpiNNaker chip floor plan.

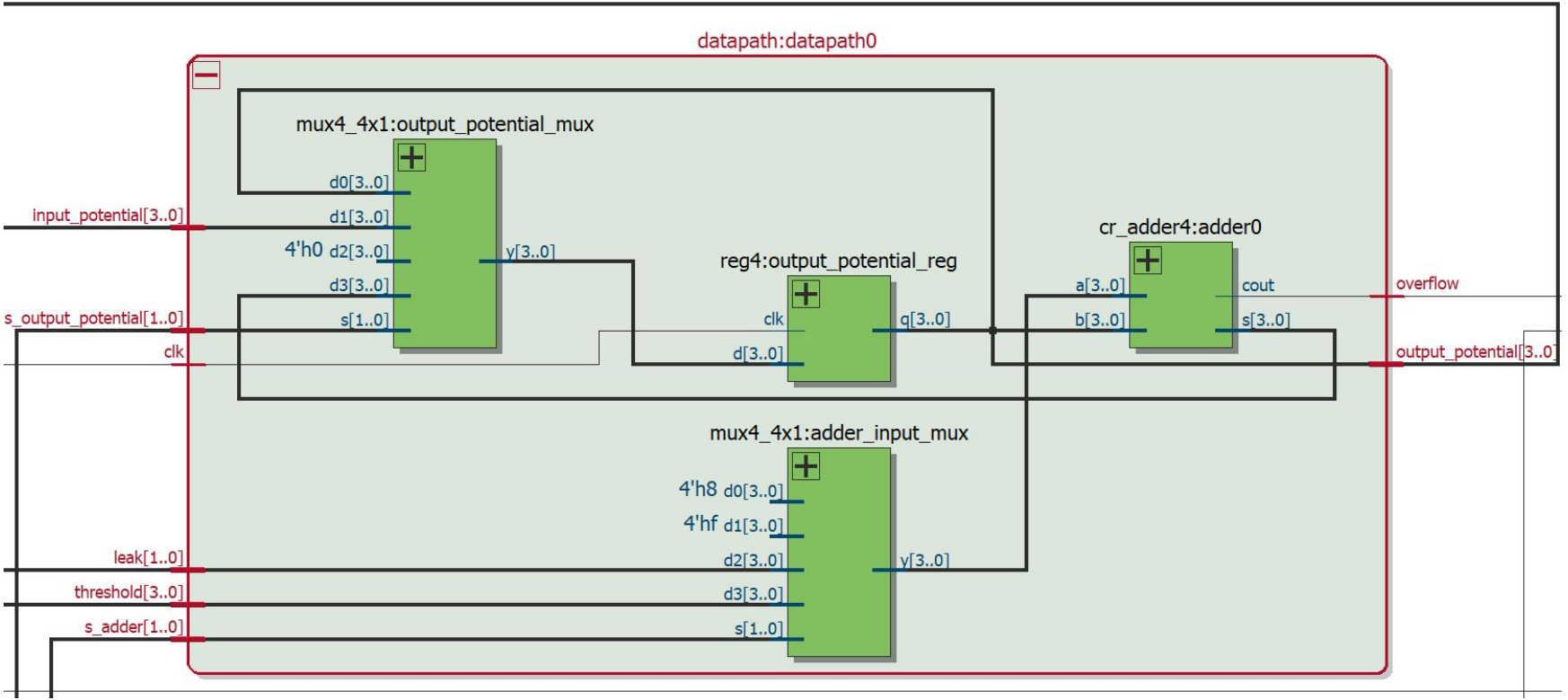
Our Approach: Not Quite TrueNorth™

- Our architecture is inspired by the architecture of IBM TrueNorth
- "Virtual Crossbar Array"- SRAM representing up to 256 synaptic connections
- A single datapath for performing the arithmetic operations of each neuron
- A single chip contains 16 neurons, all of which can be connected to each of the 16 other neurons through the virtual crossbar array
- Spikes from the neurons act as our "input" and "output," meaning chips could, in principle, be "tiled" with neurons connected to neurons in other chips

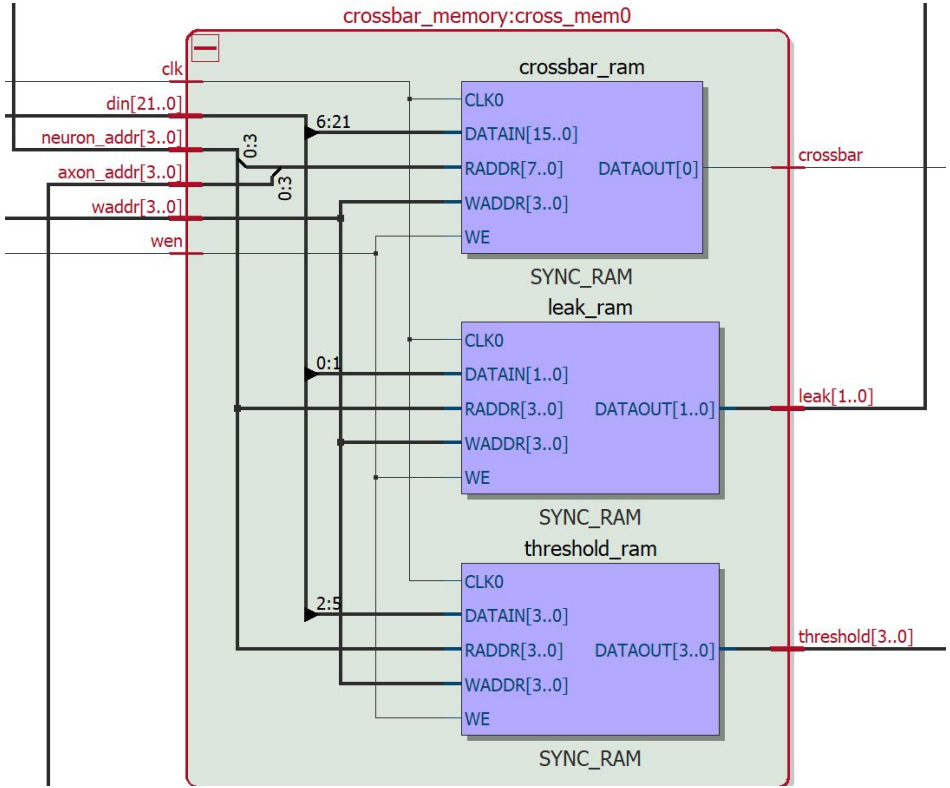
Our Architecture: Top Level



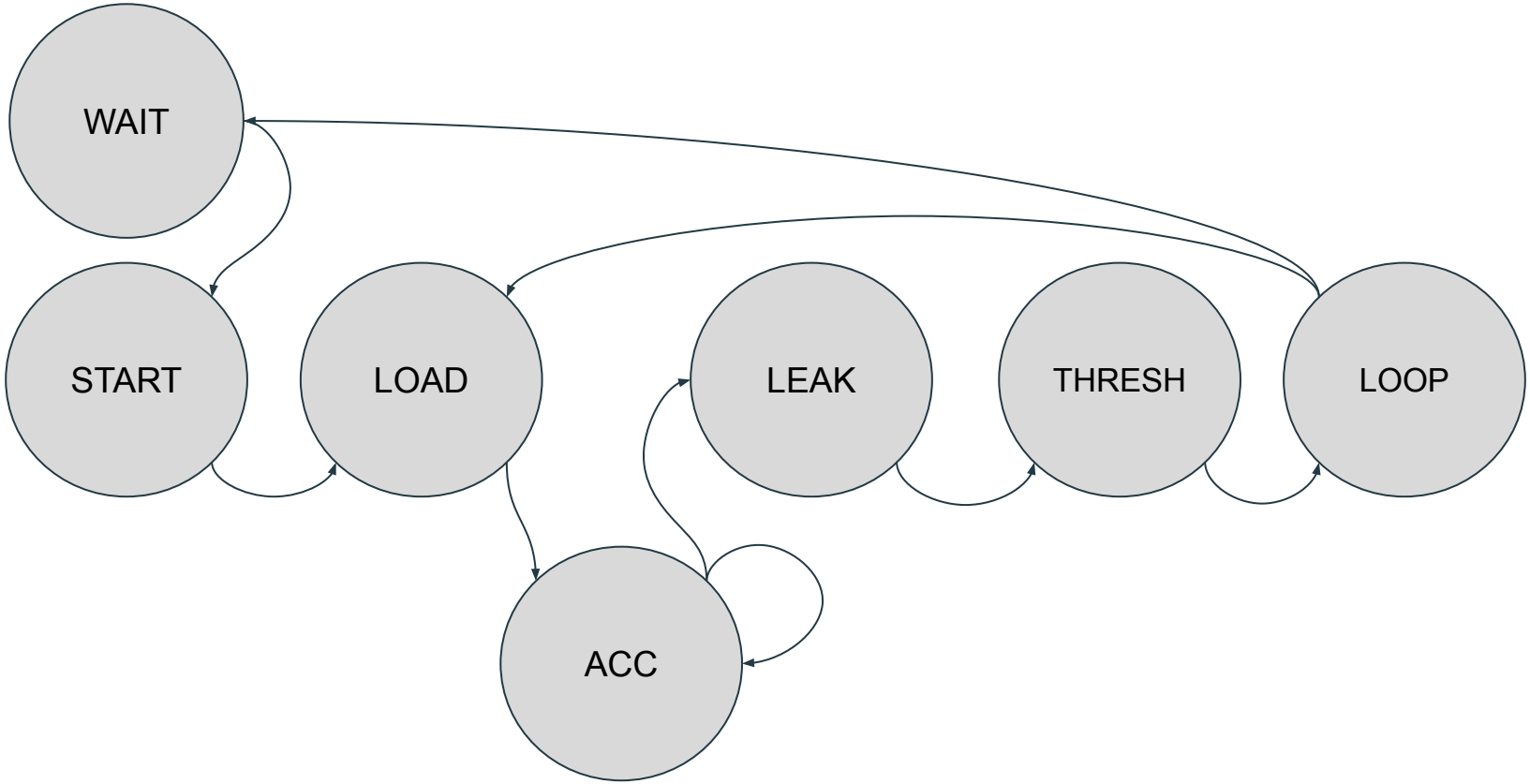
Our Architecture: Neuron Datapath



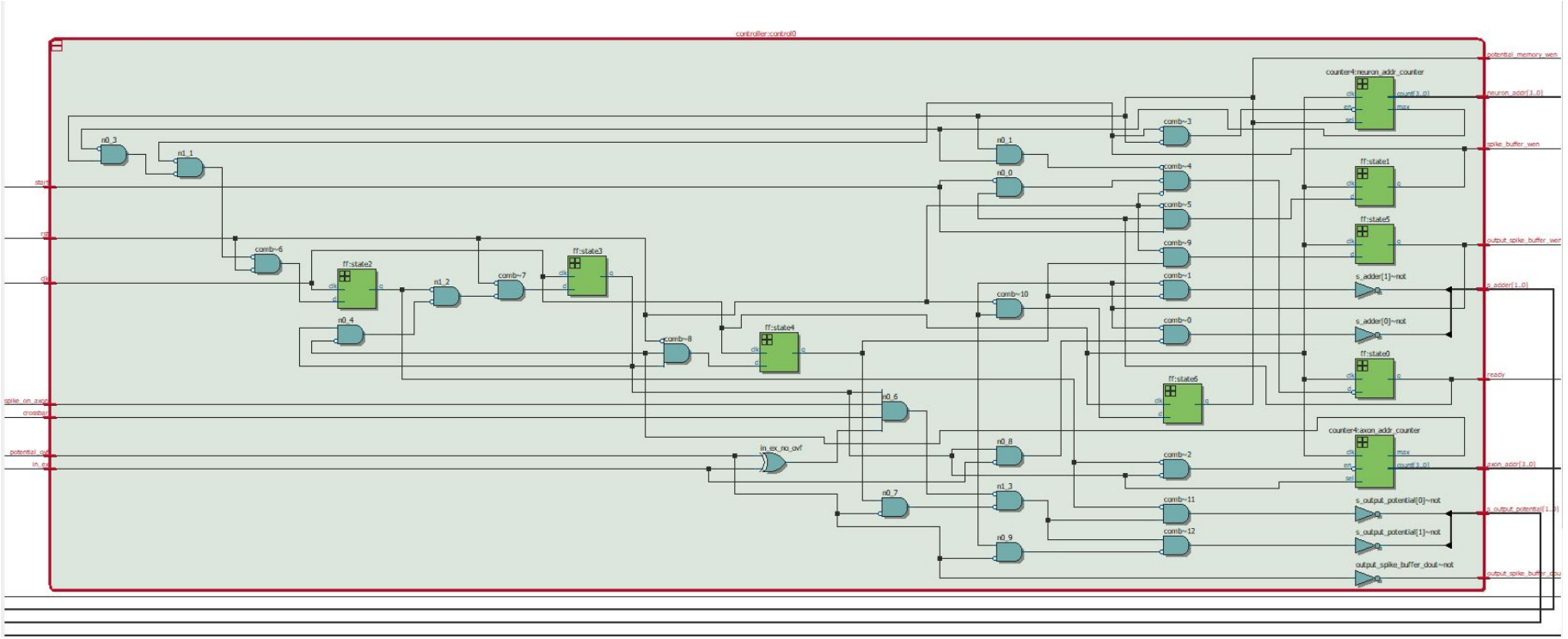
Our Architecture: Crossbar Memory



Our Architecture: Controller States



Our Architecture: Controller



Verilog Model

- Fully structural Verilog (implemented directly with transistor models, or with primitive gates (AND, OR, etc.)
- Controller, Datapath, RAM modules for network parameters and membrane potentials, and a few registers (for I/O and for spikes)

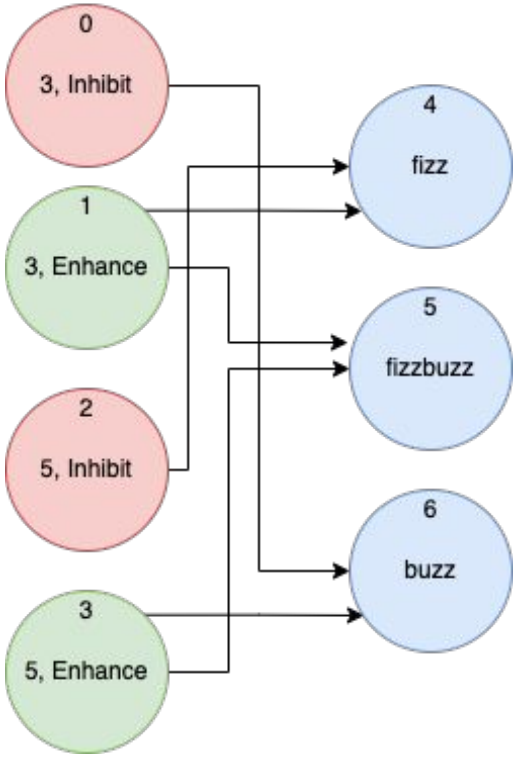
Python Model

- In order to test our Verilog model, we also created a model in python
- This model is highly simplified and abstracted
- SpikingNeuron class - encapsulates the threshold, leak, potential, spike, and inhibit/enhance signal of a single Neuron
    - Holds references to axons (other SpikingNeuron objects) to accept spikes
- JSON IO to configure the each Neuron and the crossbar connections

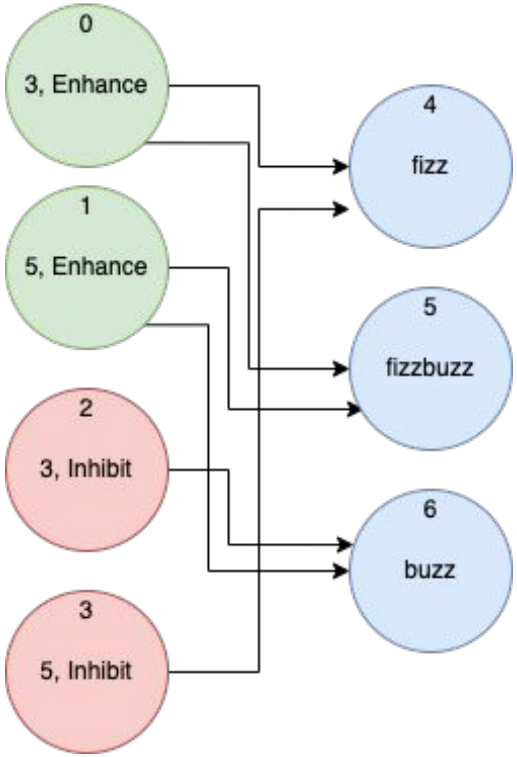
Python Tests - ½+½ and fizzbuzz

- Debugging fizzbuzz in the python model revealed a significant bug in GridNorth
- Our overflow detection makes the order of the incoming spikes order dependent
  - We throw out operations that would result in overflow/underflow
  - For example:
    - Potential = 0, Threshold = 1
    - Receive incoming inhibit (Potential--), this operation thrown out
    - Receive incoming enhance (Potential++), Potential = 1
    - Spike Generated! Although Potential should be 0.

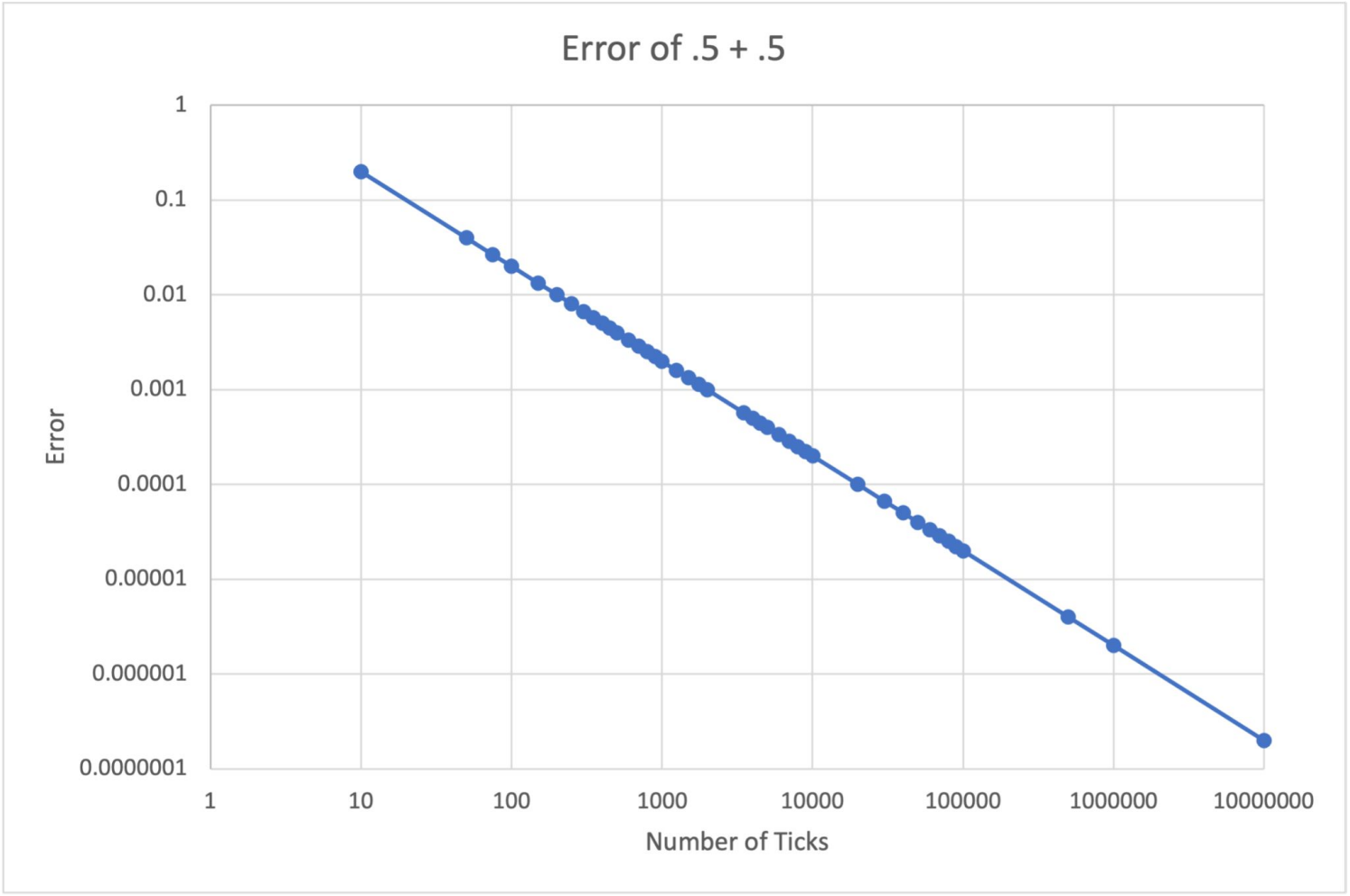
Original Solution



Correct Solution




```
fizzbuzz
16
17
fizz
19
buzz
fizz
22
23
fizz
buzz
26
fizz
28
29
fizzbuzz
31
32
fizz
34
buzz
fizz
37
38
fizz
buzz
41
fizz
43
44
fizzbuzz
```

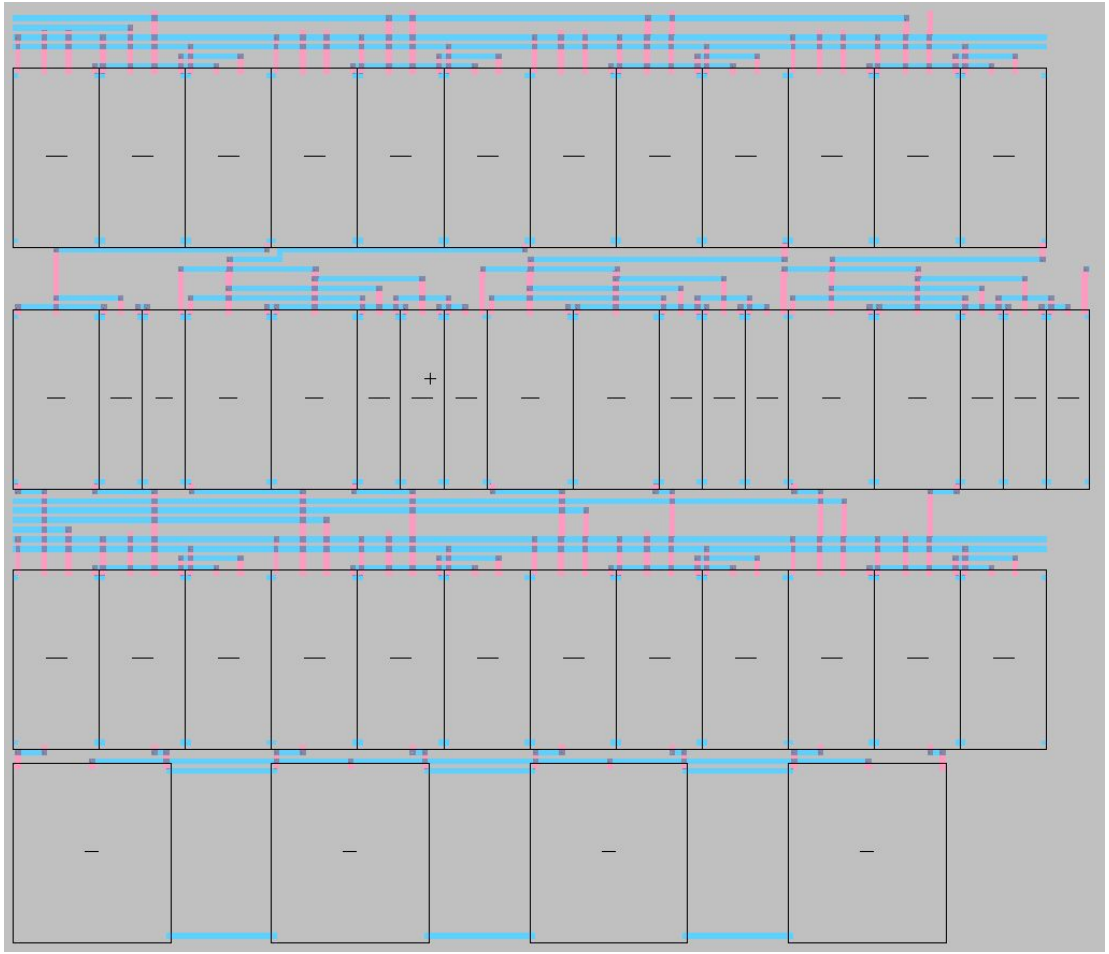

Error of .5 + .5
1
0.1
0.01
0.001
0.0001
0.00001
0.000001
0.0000001
Error
1
10
100
1000
10000
100000
1000000
10000000
Number of Ticks

Datapath Layout

- Used 5 basic blocks: INVERTER, NAND, MUX2, XOR, Flip Flop
- Built higher level blocks: MUX4, Half Adder, Full Adder
- Used lambda size of basic blocks from 3 µm layouts (from book)
- Designed wiring between blocks following rules of 4λ wide metal with 4λ spacing

Datapath Layout

- Organized in 4 layers:
  - 4 bit 4x1 mux
  - 4 bit adder
  - 4 bit 4x1 mux
  - 4 flip flops
- Dimensions
  - 900λ x 780λ
- Area:
  - Total: 702,000λ^2
  - Blocks: 495,000λ^2
  - Percent Blocks: 70.5%
  - Percent Other: 29.5%



Area Estimation

- Sources: Area Estimation table from the book, pg. 52
- Our layout for the Datapath used around $1000\lambda^2$ transistor without interconnect- even more with interconnect
- We'll use the high end of the "random logic" density estimate for controller area and register area estimations, and the SRAM density estimate for our on-chip SRAMs

	Controller	Datapath	Shift Register	Other Registers	Mux & Or
Area (λ^2)	1014000	702000	912000	3312000	504000

	Potential Memory	Crossbar Memory
Area (λ^2)	64000	352000

Total Logic Area	6444000
Total SRAM Area	416000
Total Area	6860000

Area Estimation

- Total Estimated Chip Area: 6,863,000λ^2
- ND Process: 6.863 mm^2
- 65 nm process: 0.007 mm^2

Retrospective

Github

https://github.com/patrick-harkins/GridNorth