Mnemosyne: Neural Network for Pattern Recognition

Group B

Matt Walker, Joe Montgomery, Luke Hoban, CJ Ganier

What is a Neural Network?

- Model for biological learning
- Learns by example
- Excels at pattern recognition

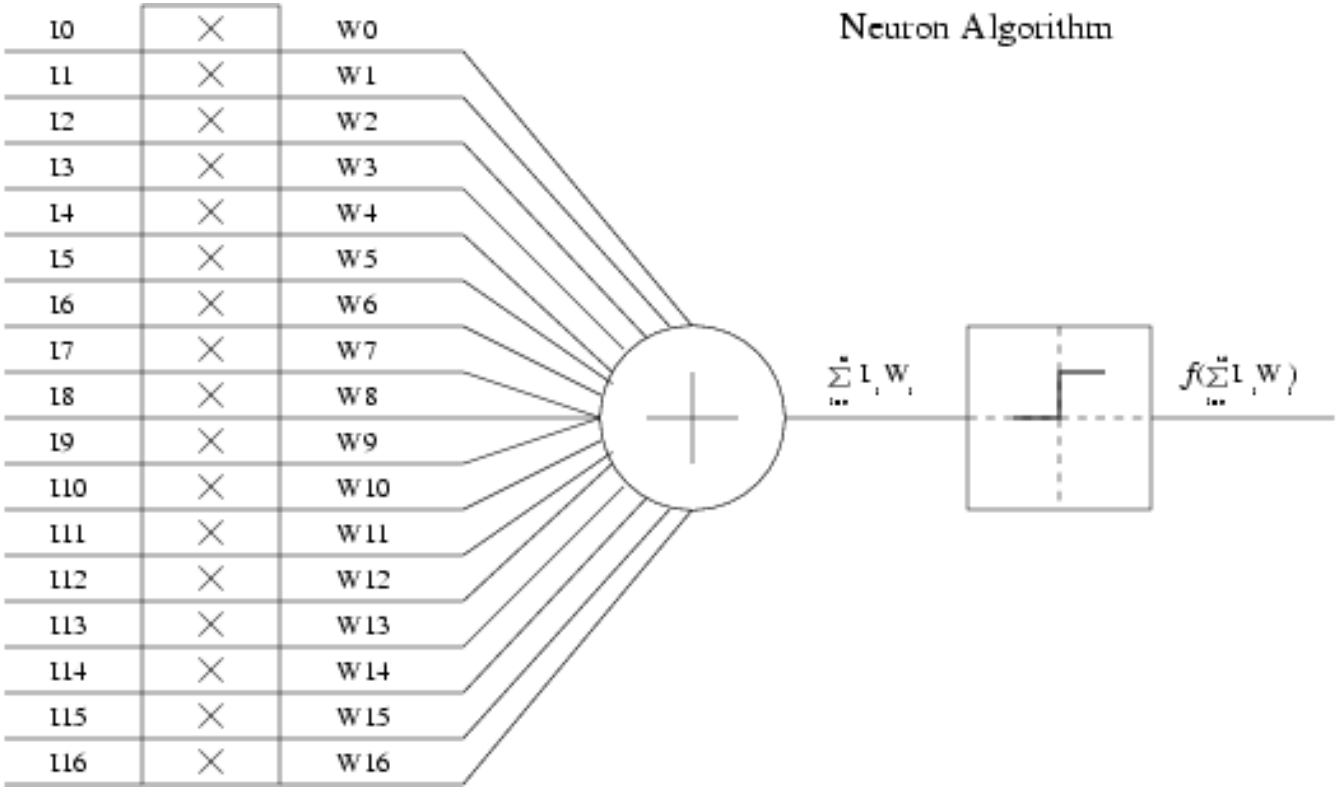
General Learning Algorithm

- First train, then test the network
- Train:
  - Give examples of “good” and “bad” patterns
  - Network learns by changing weights
  - Uses positive and negative reinforcement
- Test:
  - Release net into real world; let it make decisions

Network Structure

- Neuron:
  - Decision making node
  - State is either on or off
- Axon:
  - Connects nodes
  - Has a weight, or importance

General Algorithm



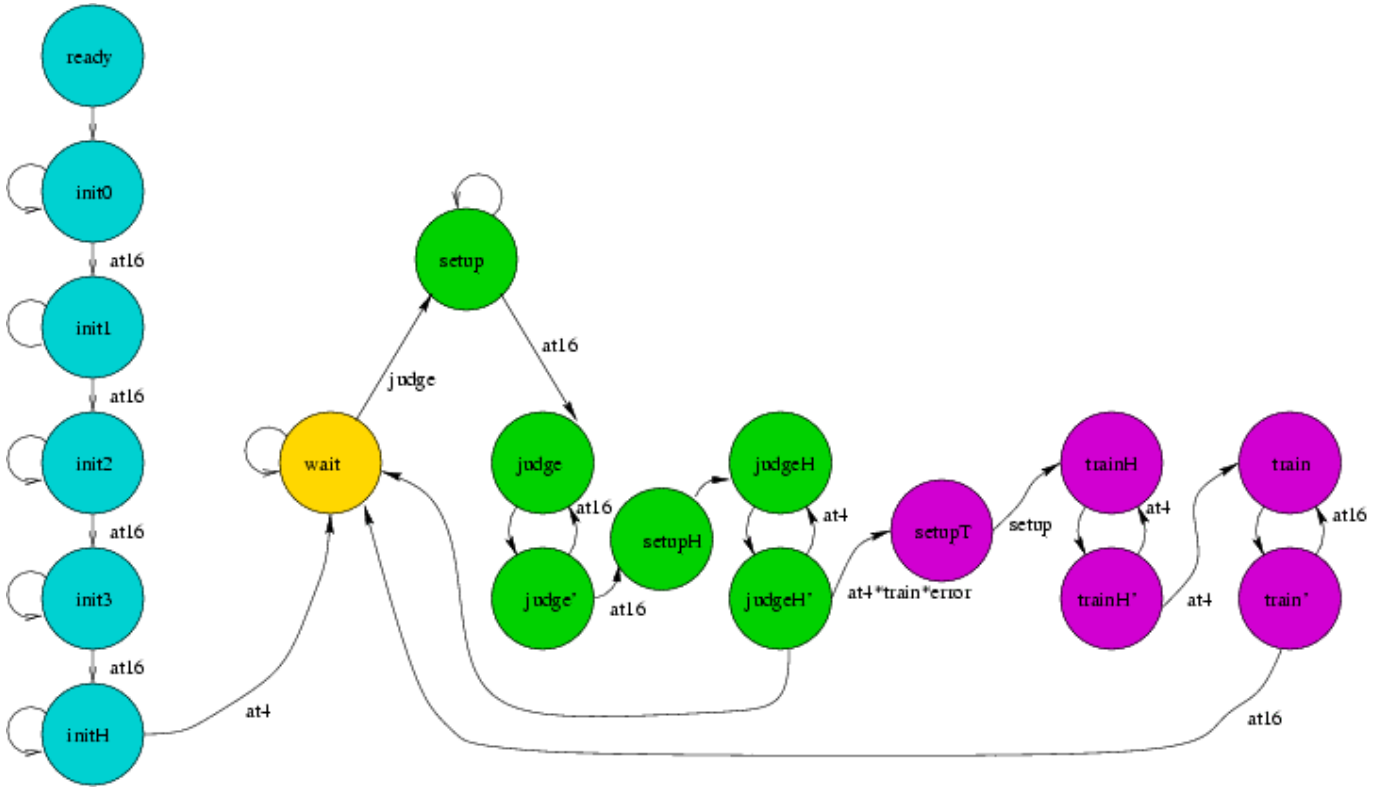
Our Implementation

- 17 one bit inputs per neuron
- Four input neurons
- One output neuron
- Four bit weights
- Step activation function

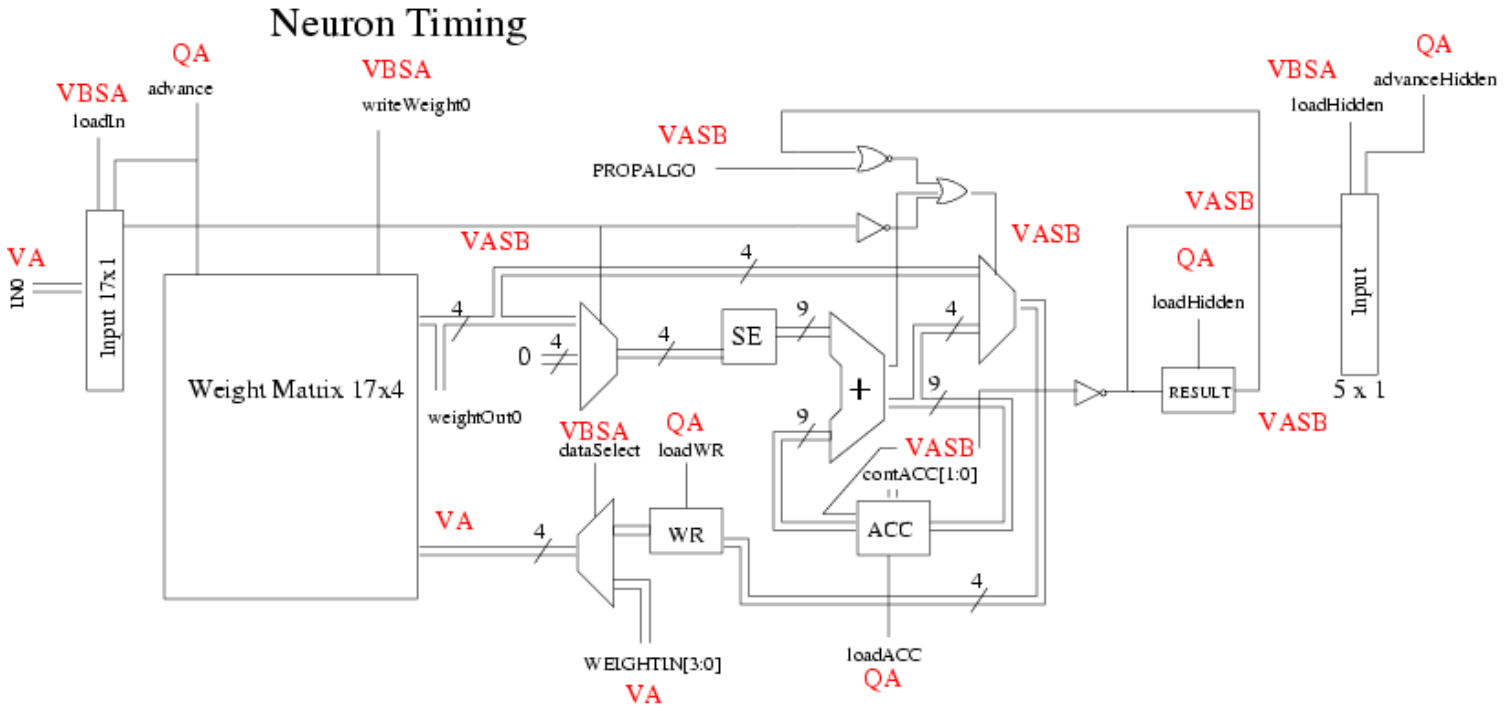
Our Algorithm

- Assign random weights to each axon
- For each case:
  - Feed in inputs
  - Propagate firing values forward
  - Update weights if training

Control Flow

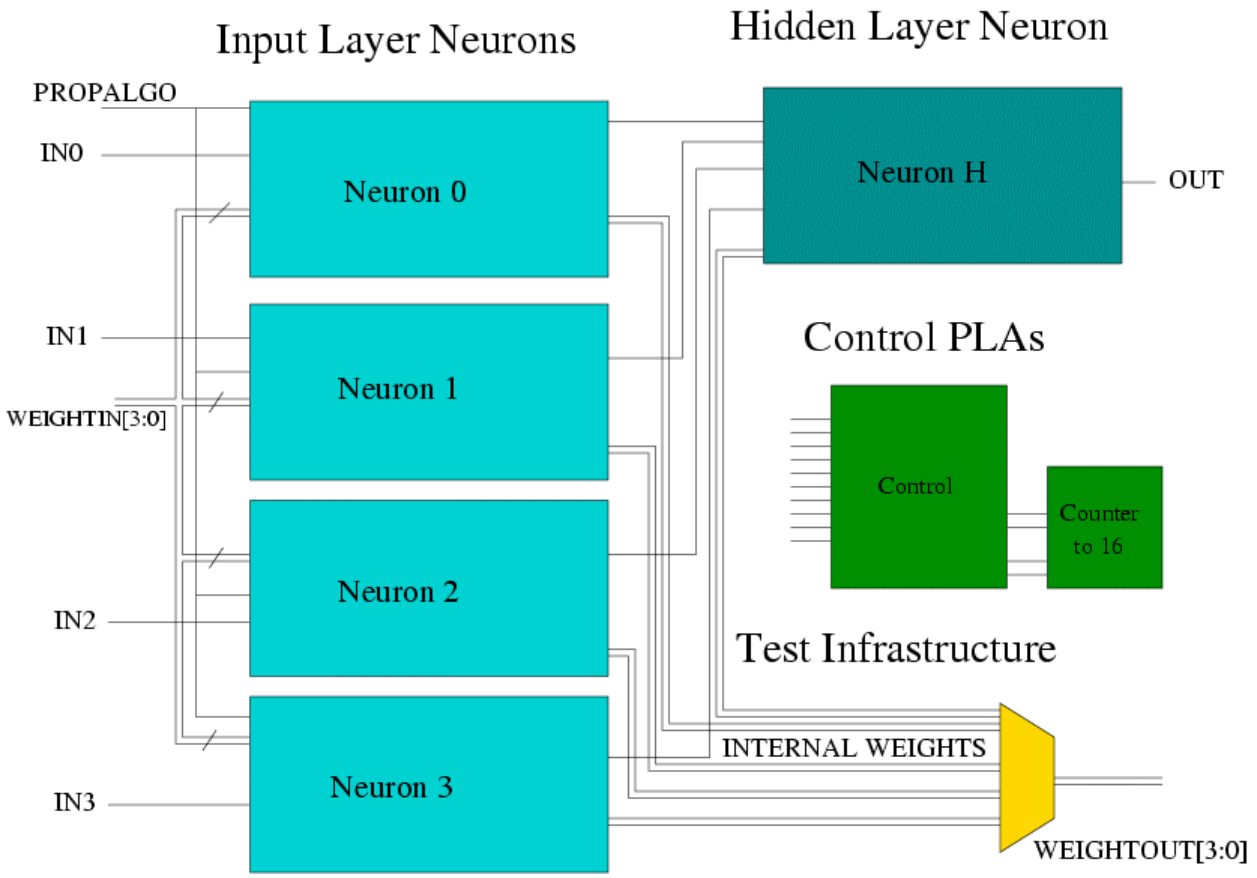


Timing Diagram



Control Timing is either VBSA or QA

Block Diagram

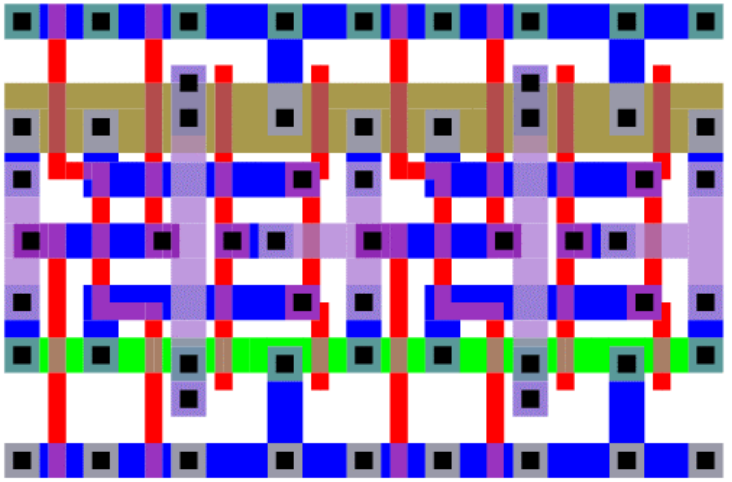


Subcells

- Five neurons
  - 17 4bit weights
  - 17bit input arrays
  - Miscellaneous registers
- Requires over 800 latches
  - Original latches 100x40 lambda
  - Occupied 80% of chip in 1.5*u*m process
  - Discrete transistors inefficient

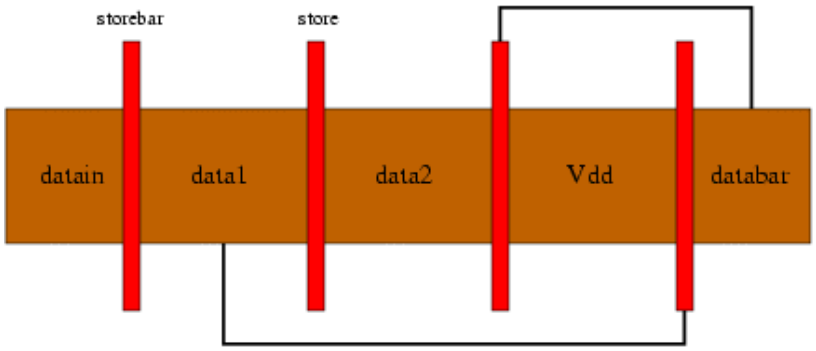
The Latch

- New latch is half as large
  - 800 occupy only 20% of chip in 0.5*u*m process
  - Eliminates discrete transistors
- Graph theory
  - Simple combinational method in Weste
  - Generalized to sequential logic

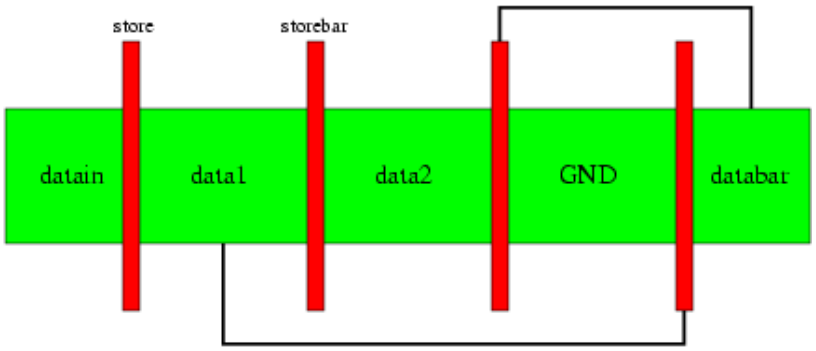


Graph Theory

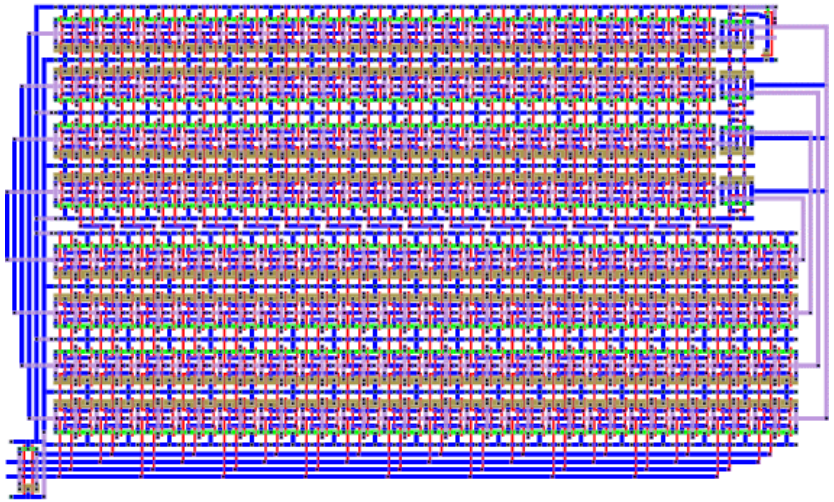
P-Type Graph



N-Type Graph



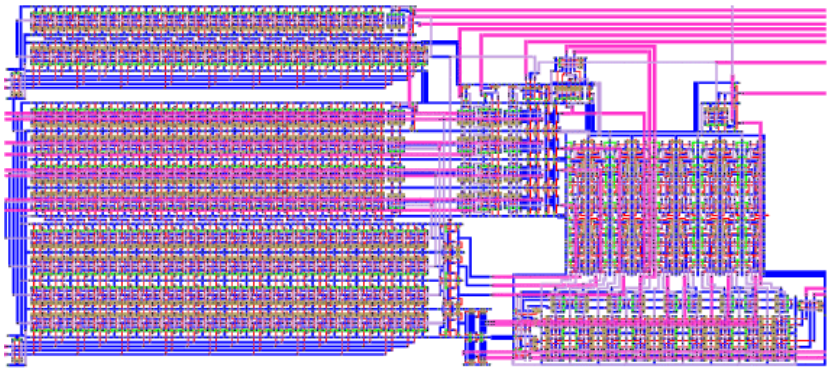
Weight Matrix



- Shifting memory
  - Takes advantage of serial access in accumulator-style process
  - Latch overlaps significantly for tiling savings
  - Dense
- No use of metal 3

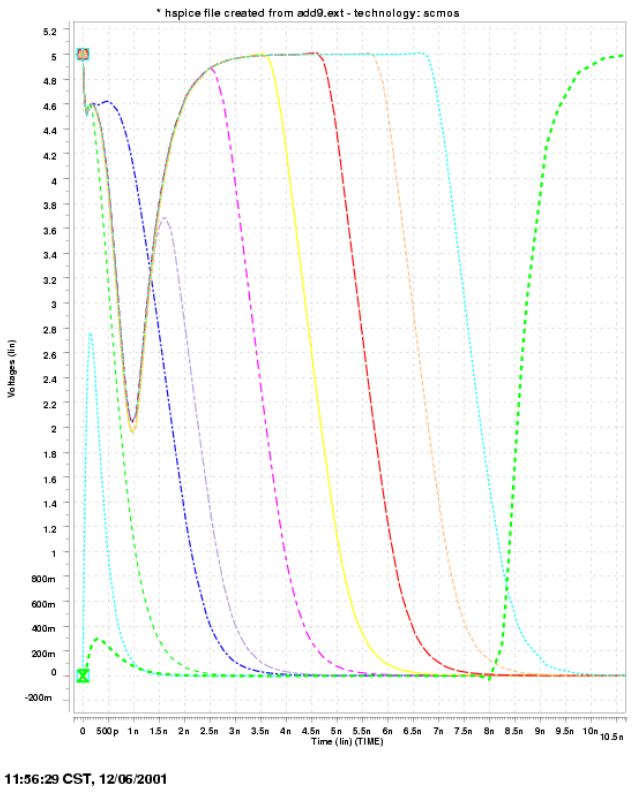
Neuron

- Input stage
  - Input array
  - Weight matrix
- Accumulating adder
- Data path structures
  - Muxes
  - Write register
- Routing
  - Subcells done without m3
  - m3 to the boundaries

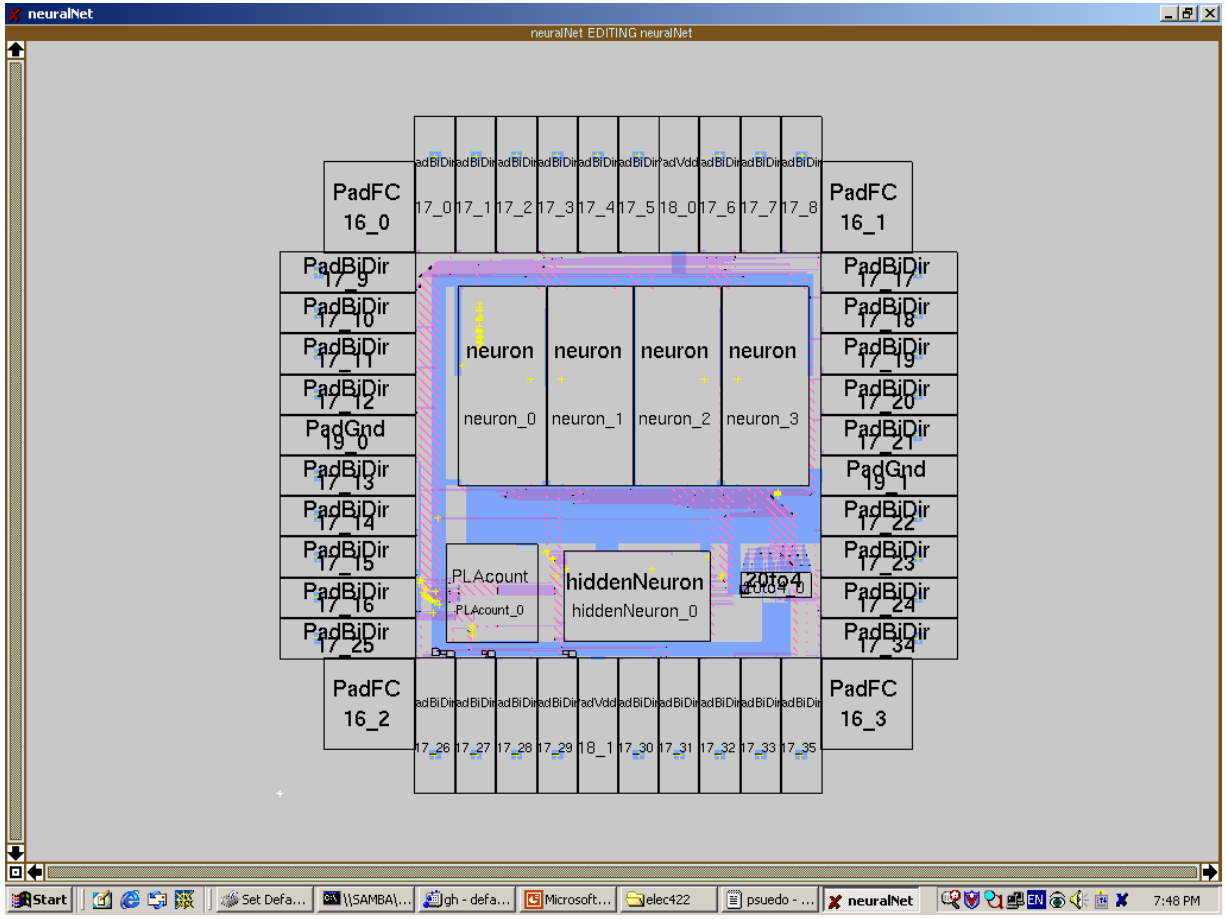


System Timing

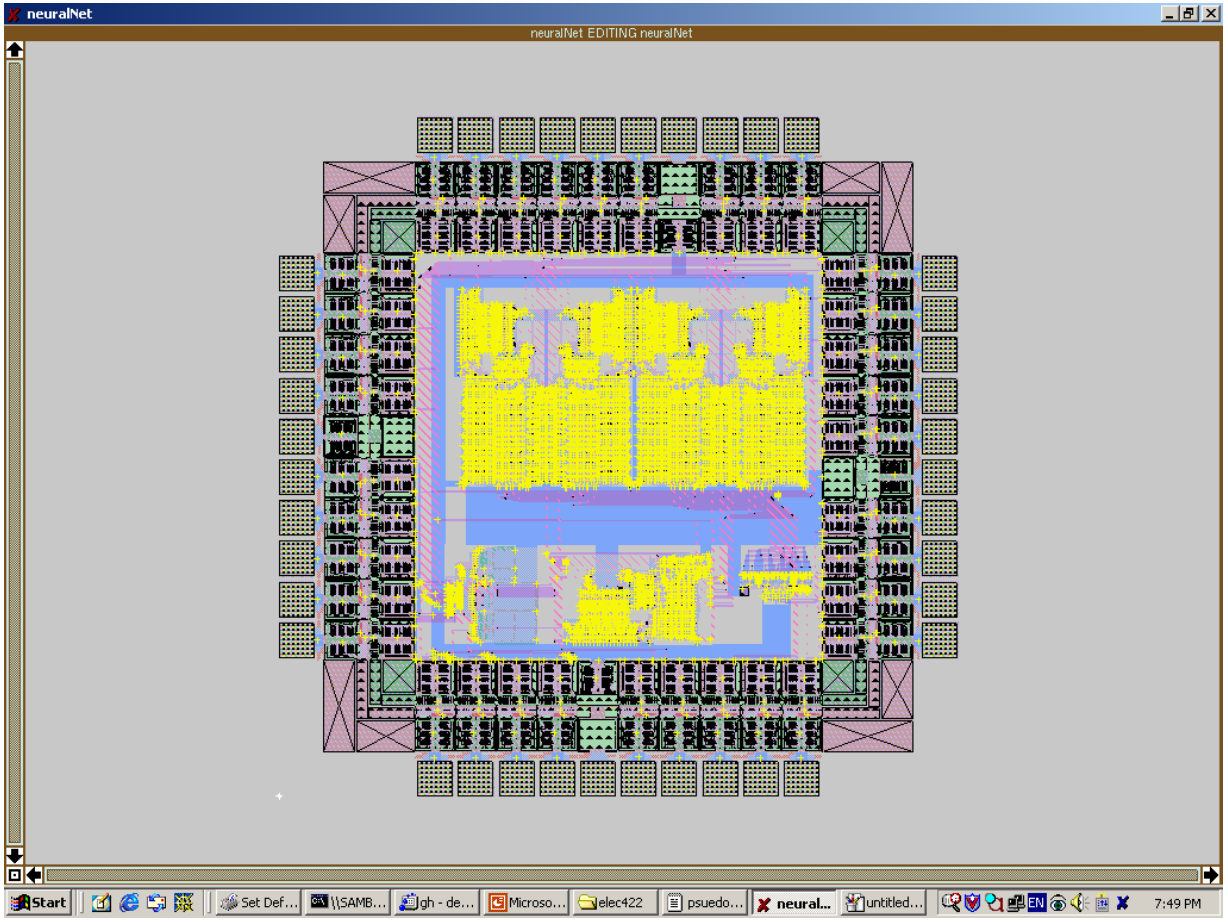
- Longest path 20ns
- Estimated frequency 50MHz
- 108 cycles per training
- 460,000 trainings per second



Floorplan


neuralNet
neuralNet EDITING neuralNet
PadFC 16_0
PadFC 16_1
PadFC 16_2
PadFC 16_3
PadBiDir
PadGnd 19_0
PadGnd 19_1
neuron
neuron_0
neuron_1
neuron_2
neuron_3
PLAcount
PLAcount_0
hiddenNeuron
hiddenNeuron_0
20to4
Start
Set Defa...
\\SAMBA\...
gh - defa...
Microsoft...
elec422
psuedo - ...
neuralNet
7:48 PM

Status - DONE


neuralNet
neuralNet EDITING neuralNet
Start
Set Def...
\\SAMB...
gh - de...
Microso...
elec422
psuedo...
neural...
untitled...
7:49 PM

Algorithm Design

- Standard neural network algorithm
  - Weights are real numbers (continuous valued)
  - Activation function is complex non-linear function
  - Multiplication required
- We designed a stripped down but effective algorithm
- Required extensive verification and modification

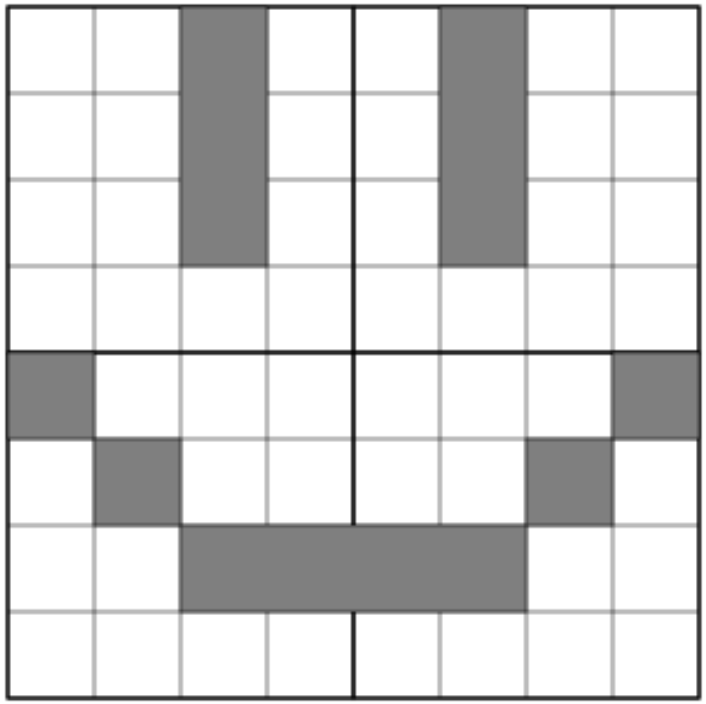
Simulation

- Developed a complete simulation of the algorithm
- Parameterized over all of the design decisions
- Automatically generates IRSIM test vectors for simulated runs
- Used to verify that we can train neural net to correctly recognize patterns 98% of the time

Uses

- Reading ZIP codes
- Pattern recognition on 64 bit inputs
- Limitations
  - Only 1 output bit (Yes/No)
  - Single bit inputs (Black/White Image)
  - Only 64 bit input (8x8 Image)
  - 4 bit weights (limited precision learning)

Conclusion



Neuron 1 Input: 0010 0010 0010 0000 1

Neuron 2 Input: 0100 0100 0100 0000 1

Neuron 3 Input: 1000 0100 0011 0000 1

Neuron 4 Input: 0001 0010 1100 0000 1