

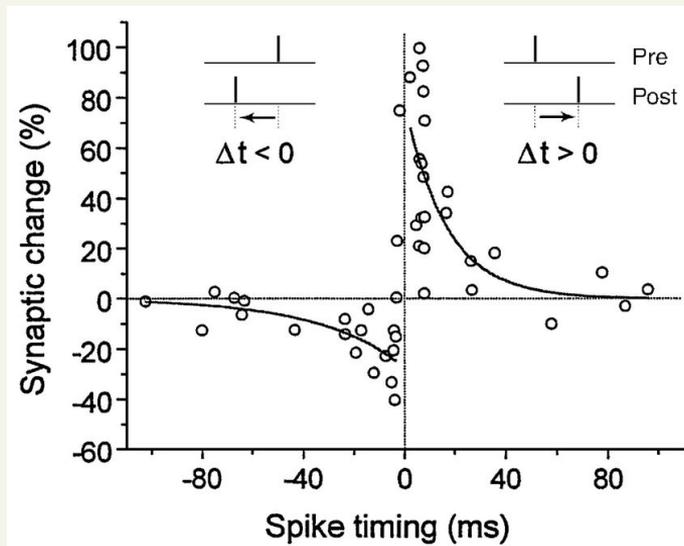
ECE 274 Final Presentation

# Triplet STDP Verilog Implementation

Nik Belle, Patrick Truesdale, Dakota Barnes, Logan Unger

## Pair-based STDP Recap

- Recall from lecture that STDP learning adjusts the weights of synapses based on the timing delta between pre- and post-synaptic spikes
- However, it is limited since it only looks at one timing delta, which limits the spike sequences that can learn well



## Pair-Based STDP Shortcomings

- “Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity” by Pfister and Gerstner points out that pair-based STDP inaccurately models experimental data
- Property 1: Frequency independence
  - ◆ Pair-based applies same weight change regardless of repetition rate
- Property 2: Symmetry under burst
  - ◆ Pre-post-pre firings and post-pre-post yield same weight change

# Triplet STDP Proposal

- Propose a triplet rule
- Introduce slow traces  $r_2$  and  $o_2$  to be rate/burst integrators (used alongside standard fast traces)
- Lets us understand how much recent activity there has been
- Frequency dependence:
  - ◆ At high repetition rates, the slow trace still has spike history. At low rates, it collapses to pair-based STDP.
- Burst asymmetry:
  - ◆ Two traces will weight LTP and LTD differently for pre-post-pre and post-pre-post scenarios

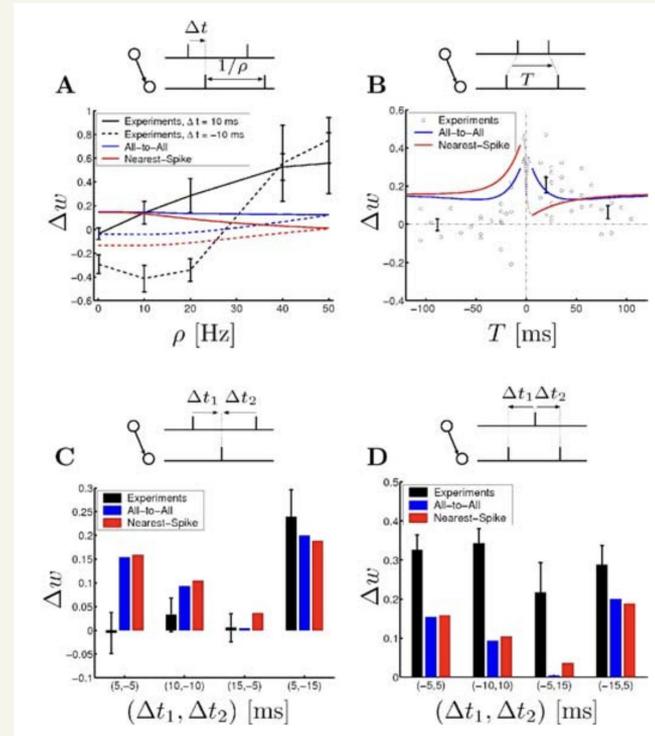


Figure 3: Triplet Experiment vs Pair-STDP Data (Pfister and Gerstner, 2006)

## Project Goal

- Implement a triplet STDP rule in Verilog for a 2-layer SNN with 2 output neurons
- Train and test using 5x5 pixel “zero” and “one” images
- Use HW4 as a baseline to understand how STDP provides improvement
- Evaluate:
  - ◆ Output neuron spiking
  - ◆ Learned weight maps
- Analyze PPA differences between pair-based and triplet STDP in the same SNN

## Triplet Implementation: Changes from HW4 SNN

- Replace binary shift registers with 4-bit decaying trace variables
  - ◆ Fast traces ( $r1, o1$ ) halve each cycle
  - ◆ Slow traces decay linearly by  $-2/\text{cycle}$
- Expanded weights from 2-bit to 4-bit
- Added a leak term
- Voltage parameters scaled 4x to match wider weight range.

## Triplet Implementation: Learning Rule

Did pre fire?                      Did post fire before that?

↓    ↓

LTP →  $\Delta w^+ = r_1[i] \cdot A_2^+ + [r_1[i] \cdot o_2/16] \cdot A_3^+$

LTD →  $\Delta w^- = o_1 \cdot A_2^- + [o_1 \cdot r_2[i]/16] \cdot A_3^-$

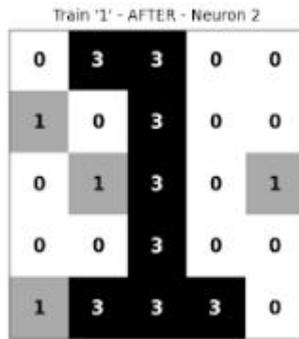
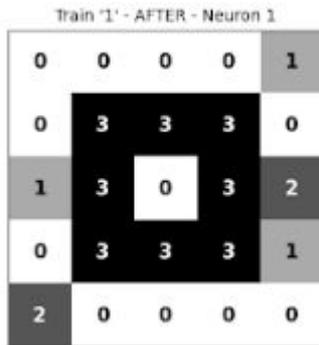
- First term = standard pair-based update
- Second term = triplet modulation via slow trace
- Divide by 16 to normalize the product of two 4 bit traces back to pair-term scale
- Pair parameters are symmetric
- Triplet parameters asymmetric (depression is 4x potentiation)

## Case 0: HW4 and Expanded Spike Trains

→ 20 bit trains, 2 bit weights/traces, pair-based, no leak

- White: 01000000100000000010
- Black: 01010100010101000101

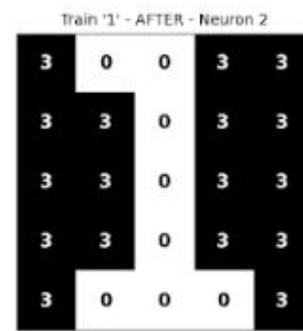
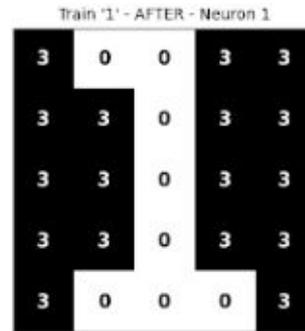
Pair	Train 0	Train 1	Test 0	Test 1
N1	2	0	2	0
N2	0	2	0	2



→ 40 bit trains, 2 bit weights/traces, pair-based, with leak

- White: 000000000000100000000000010000000001000
- Black: 000010000100001000010000100001000010000100010

Pair	Train 0	Train 1	Test 0	Test 1
N1	3	2	3	1
N2	0	2	3	1



# Case 1: Expand Weight/Trace Depth to 4 Bits

→ 40 bit trains, 4 bit weights/traces, pair-based, with leak

- White: 0000000000001000000000000010000000001000
- Black: 0000100001000010000100001000010000100010

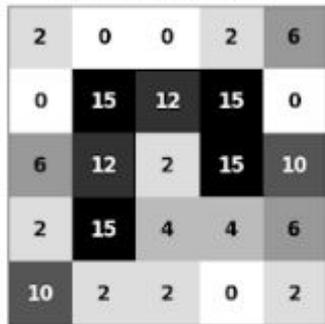
Pair	Train 0	Train 1	Test 0	Test 1
N1	2	0	2	1
N2	0	2	0	2

→ 40 bit trains, 4 bit weights/traces, triplet, with leak

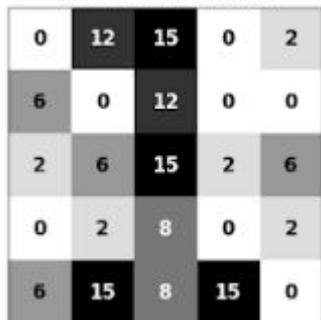
- White: 0000000000001000000000000010000000001000
- Black: 0000100001000010000100001000010000100010

Triplet	Train 0	Train 1	Test 0	Test 1
N1	2	0	2	1
N2	0	2	0	2

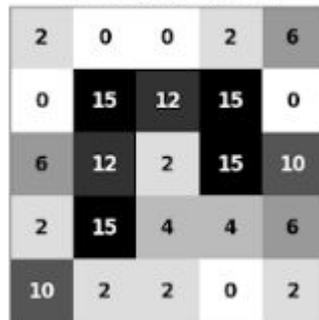
Train '1' - AFTER - Neuron 1



Train '1' - AFTER - Neuron 2



Train '1' - AFTER - Neuron 1



Train '1' - AFTER - Neuron 2

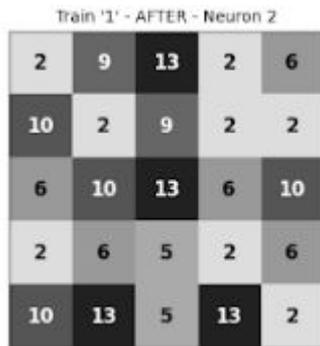


## Case 2: Spike Frequency Changes

→ 40 bit trains, 4 bit weights/traces, pair-based, with leak

- White: 001000000000100000000001000000000100000000
- Black: 0010010000100100001001000010010000100100

Pair	Train 0	Train 1	Test 0	Test 1
N1	3	3	4	4
N2	0	1	0	2



→ 40 bit trains, 4 bit weights/traces, triplet, with leak

- White: 001000000000100000000001000000000100000000
- Black: 0010010000100100001001000010010000100100

Triplet	Train 0	Train 1	Test 0	Test 1
N1	3	2	4	1
N2	0	3	0	3

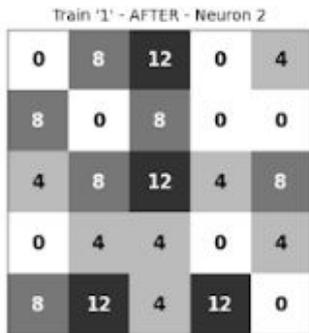
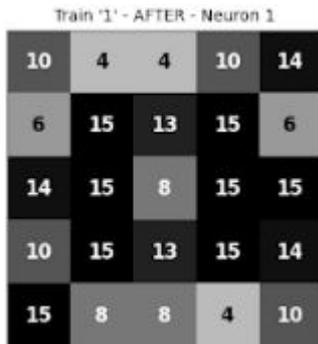


## Case 3: High-Frequency Burst Spike Trains

→ 40 bit trains, 4 bit weights/traces, pair-based, with leak

- White: 000010000000010000000010000000100000000000
- Black: 00011100000000111000000011100000001110000

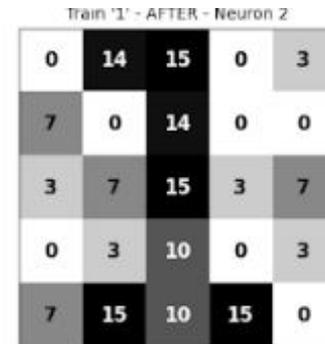
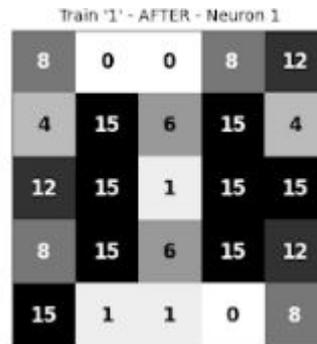
Pair	Train 0	Train 1	Test 0	Test 1
N1	4	4	5	4
N2	0	0	0	0



→ 40 bit trains, 4 bit weights/traces, triplet, with leak

- White: 000010000000010000000010000000100000000000
- Black: 00011100000000111000000011100000001110000

Triplet	Train 0	Train 1	Test 0	Test 1
N1	4	2	4	0
N2	0	2	0	4



# Hardware Optimization Validation

## PPA

- ASIC flow with Librelane for RTL → GDS
  - ◆ Yosys for synthesis, OpenROAD for pnr
- Targeting SkyWater 130nm HD standard cell library
- 15ns Clock Period

## Functional Regression

- 17 spike pattern benchmark (3 categories)
- Grade by
  - ◆ Classification accuracy
  - ◆ Spike count margin
  - ◆ Post-training weight map cosine similarity to baseline

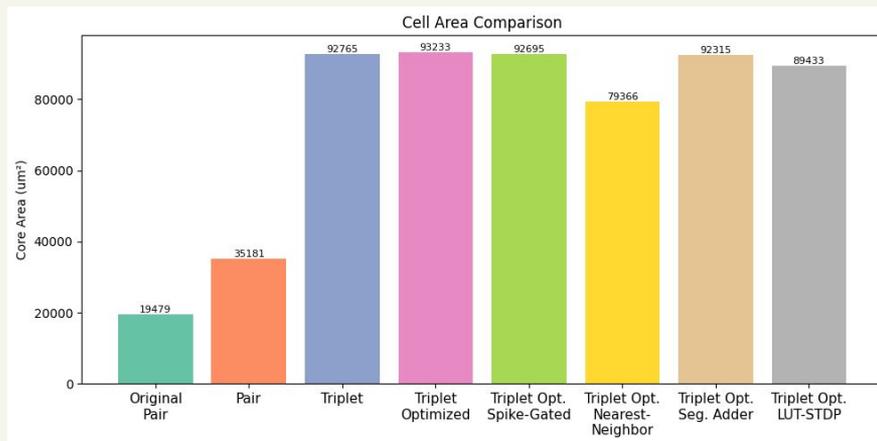
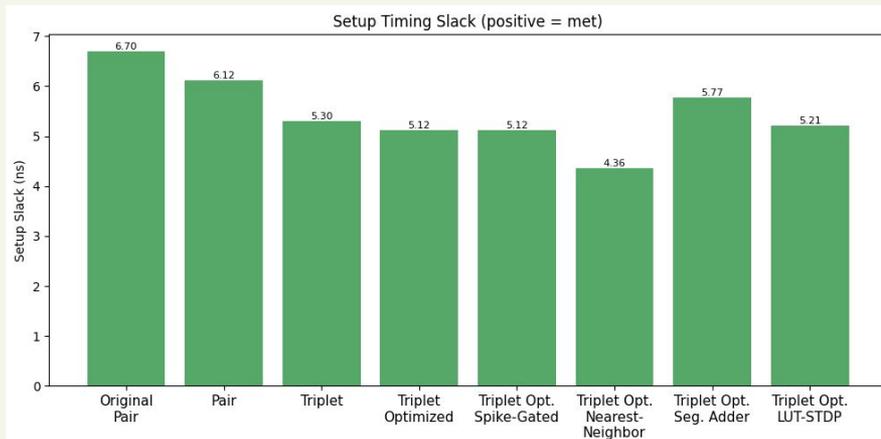
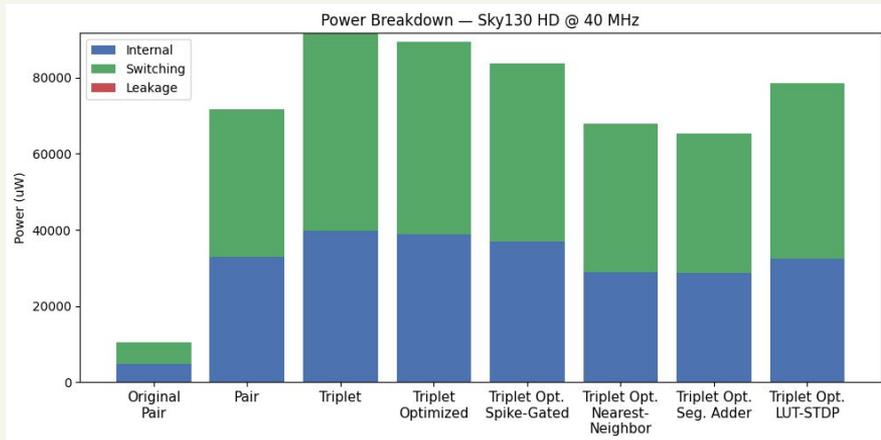
# Hardware Optimization Strategies

- Nearest-Neighbor Trace Simplification
- Approximate Segmented Adder
- Spike-Gated Clock Enable
- LUT-Based STDP

	Design	Accuracy	Mean Margin	Cosine Sim	$\Delta$ Accuracy	$\Delta$ Margin
0	Original Pair	0/17	2.35	0.7956	-17	-2.41
1	Pair	6/17	5.29	0.9398	-11	+0.53
2	Triplet	17/17	4.76	1.0000	baseline	baseline
3	Triplet Optimized	17/17	4.76	1.0000	+0	+0.00
4	Triplet Opt. Spike-Gated	17/17	4.76	1.0000	+0	+0.00
5	Triplet Opt. Nearest-Neighbor	11/17	5.88	0.9598	-6	+1.12
6	Triplet Opt. Seg. Adder	11/17	4.71	0.9088	-6	-0.06
7	Triplet Opt. LUT-STDP	17/17	4.76	1.0000	+0	+0.00

Pattern	Case	Original Pair	Pair	Triplet	Triplet Optimized	Triplet Opt. Spike-Gated	Triplet Opt. Nearest-Neighbor	Triplet Opt. Seg. Adder	Triplet Opt. LUT-STDP
0	C1-6	1	FAIL	PASS	PASS	PASS	PASS	FAIL	PASS
1	C1-9	1	FAIL	PASS	PASS	PASS	PASS	FAIL	PASS
2	C2-7	1	FAIL	PASS	PASS	PASS	PASS	FAIL	PASS
3	C2-28	1	FAIL	PASS	PASS	PASS	PASS	FAIL	PASS
4	C2-30	1	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
5	C2-32	1	FAIL	PASS	PASS	PASS	PASS	PASS	PASS
6	C1-2	2	FAIL	FAIL	PASS	PASS	FAIL	FAIL	PASS
7	C2-31	2	FAIL	FAIL	PASS	PASS	FAIL	FAIL	PASS
8	C3-Ex1	3	FAIL	FAIL	PASS	PASS	FAIL	PASS	PASS
9	C3-Ex2	3	FAIL	FAIL	PASS	PASS	FAIL	PASS	PASS
10	C3-Ex3	3	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
11	C3-Ex4	3	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
12	C3-Ex5	3	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
13	C3-Ex6	3	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
14	C3-Ex7	3	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS
15	C3-Ex8	3	FAIL	FAIL	PASS	PASS	FAIL	PASS	PASS
16	C3-Ex9	3	FAIL	FAIL	PASS	PASS	FAIL	PASS	PASS
17	TOTAL		0/17	6/17	17/17	17/17	17/17	11/17	17/17

# PPA Results



## Further Work and Conclusion

- Could have
  - ◆ Increased image size
  - ◆ Increased image fidelity
  - ◆ Increased dataset size
  - ◆ Experiment with wider variety of spike trains
- Decided to limit the scope to focus on STDP triplet implementation and searching for useful applications of it
- Results agree with the paper

---

**Thank you!**