Introduction

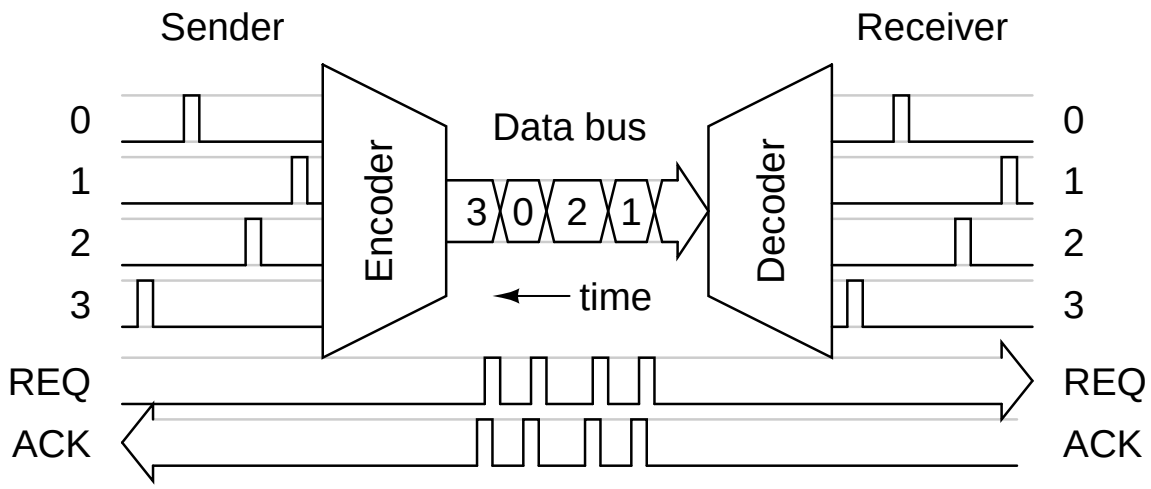
- The goal of neuromorphic engineering is to design and implement microelectronic systems that emulate the structure and function of the brain.

- Address-event representation (AER) is a communication protocol originally proposed as a means to communicate sparse neural events between neuromorphic chips.

- Previous work has shown that AER can also be used to construct large-scale networks with arbitrary, configurable synaptic connectivity.

- Here, we further extend the functionality of AER to implement arbitrary, configurable synaptic plasticity in the address domain.

Address-Event Representation (AER)

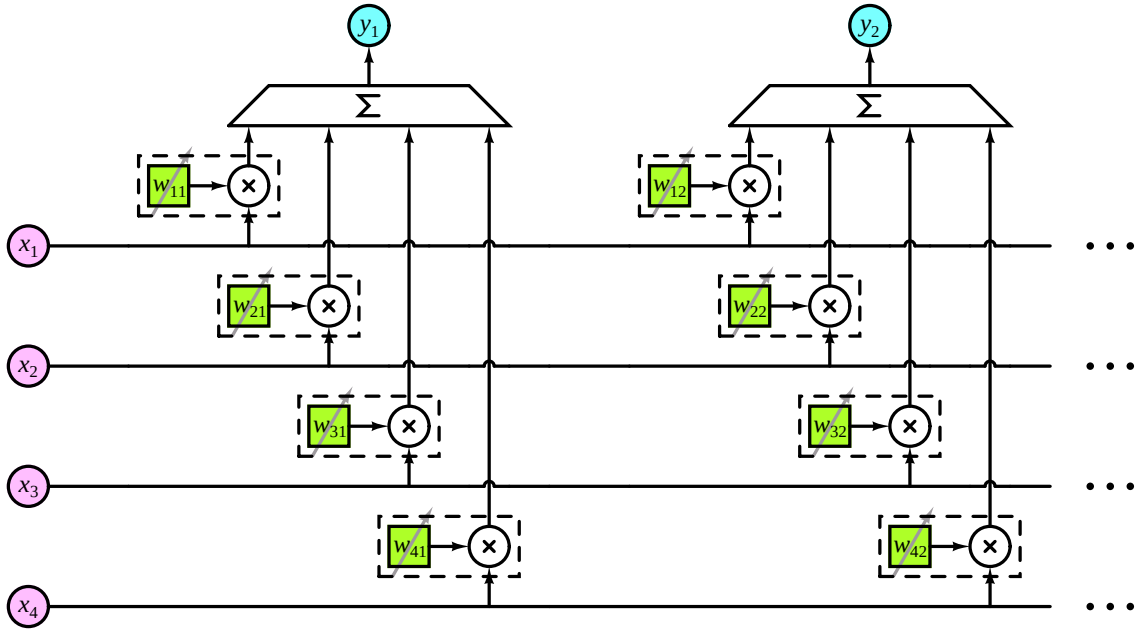


(Mahowald, 1994; Lazzaro et al., 1993)

- The AER communication protocol emulates massive connectivity between cells by time-multiplexing many connections on the same data bus.
- For a one-to-one connection topology, the required number of wires is reduced from N to $\sim \log_2 N$.
- Each spike is represented by:
  - Its location: explicitly encoded as an address.
  - The time at which it occurs: implicitly encoded.

Learning on Silicon

- Adaptive hardware systems commonly employ learning circuitry embedded into the individual cells.
- Executing learning rules locally requires inputs and outputs of the algorithm to be local in both *space* and *time*.
- Implementing learning circuits locally increases the size of repeating units.
- This approach can be effective for small systems, but it is not efficient when the number of cells increases.



Address Domain Learning

- By performing learning in the address domain, we can:
  - Move learning circuits to the periphery.
  - Create scalable adaptive systems.
  - Maintain the small size of our analog cells.
  - Construct arbitrarily complex and reconfigurable learning rules.
- Because any measure of cellular activity can be made globally available using AER, many adaptive algorithms based on incremental outer-product computations can be implemented in the address domain.
- By implementing learning circuits on the periphery, we reduce restrictions of locality on constituents of the learning rule.
- Spike timing-based learning rules are particularly well-suited for implementation in the address domain.

Enhanced AER

- In its original formulation, AER implements a one-to-one connection topology.

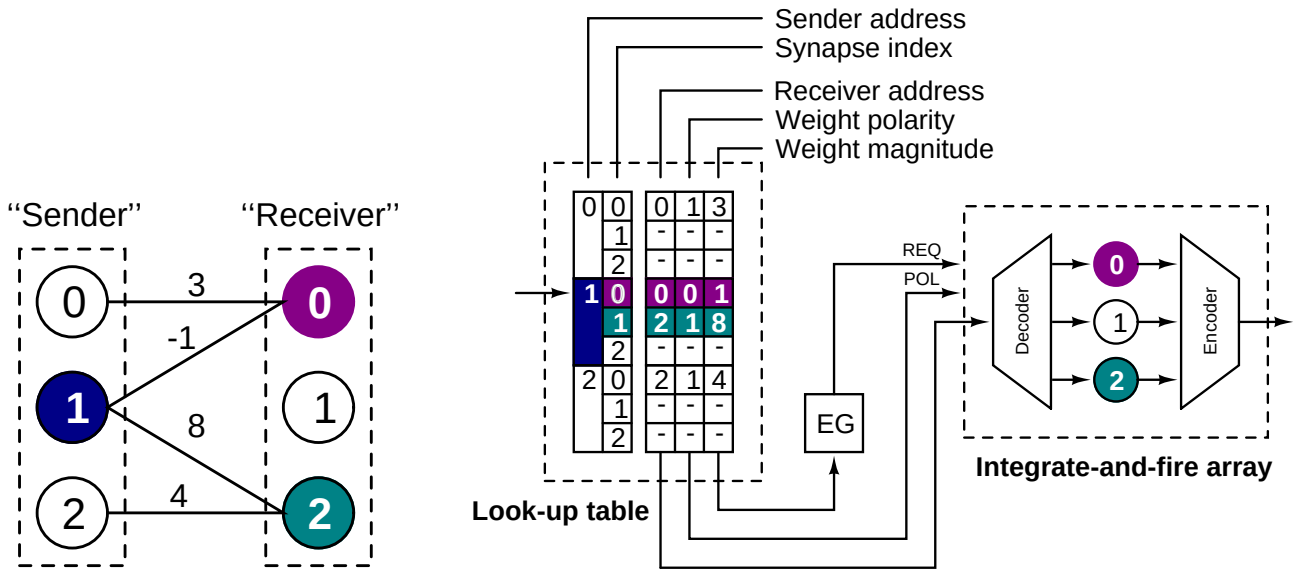
- To create more complex neural circuits, convergent and divergent connections are required.

- The connectivity of AER systems can be enhanced by routing address-events to multiple receiver locations via a look-up table (Andreou et al., 1997; Diess et al., 1999; Boahen, 2000; Higgins & Koch, 1999).

- Continuous-valued synaptic weights can be obtained by manipulating event transmission (Goldberg et al., 2001):

W	$=$	n	$\times$	p	$\times$	q
Weight		Number of spikes sent		Probability of transmission		Amplitude of postsynaptic response

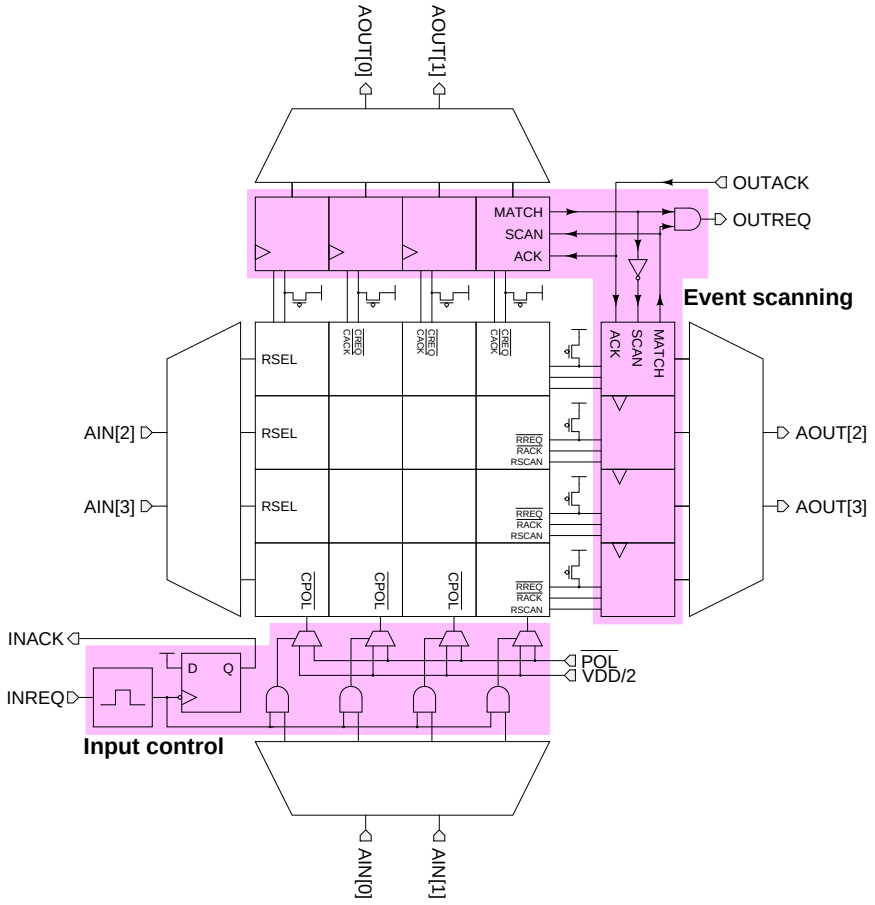
Enhanced AER: Example



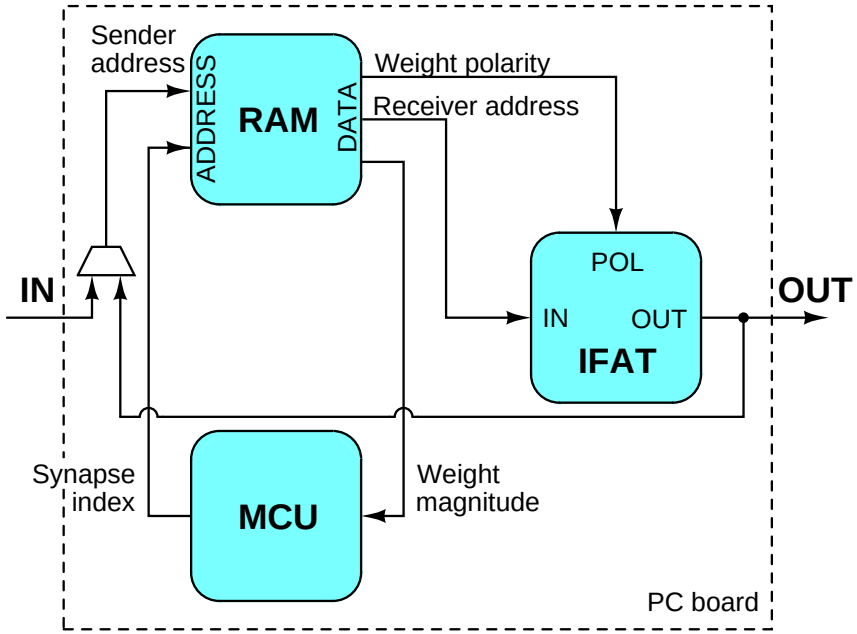
- A two-layer neural network is mapped to the AER framework by means of a look-up table (LUT).
- The event generator (EG) sends as many events as are specified in the weight magnitude field of the LUT.
- The integrate-and-fire array transceiver (IFAT) spatially and temporally integrates events.

Architecture

IFAT

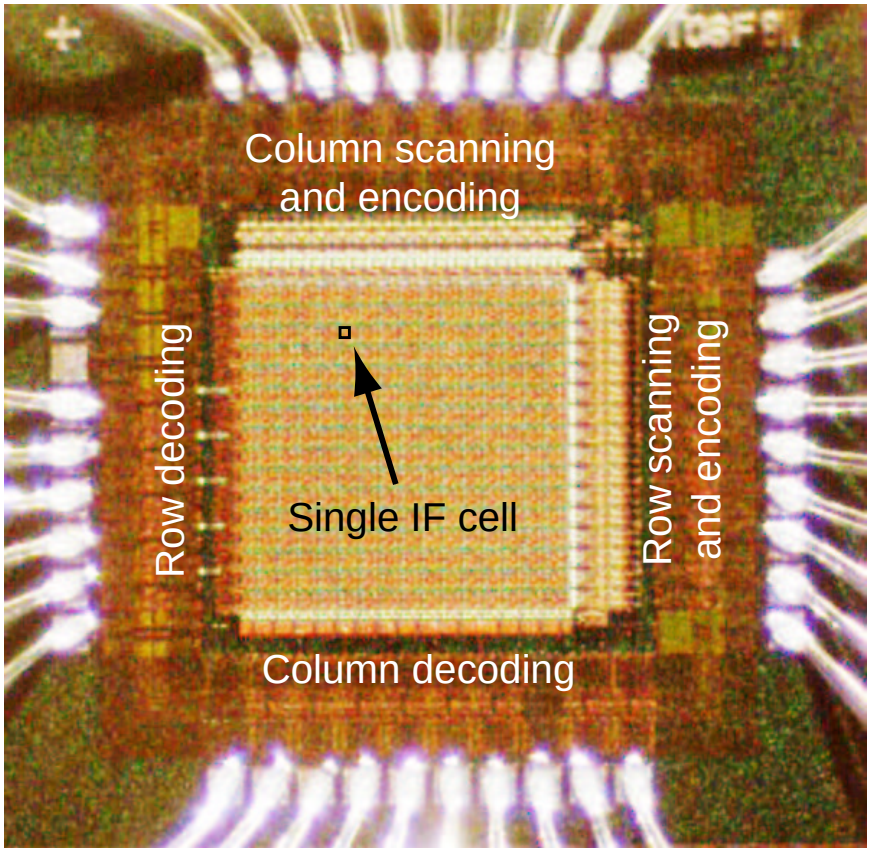


System

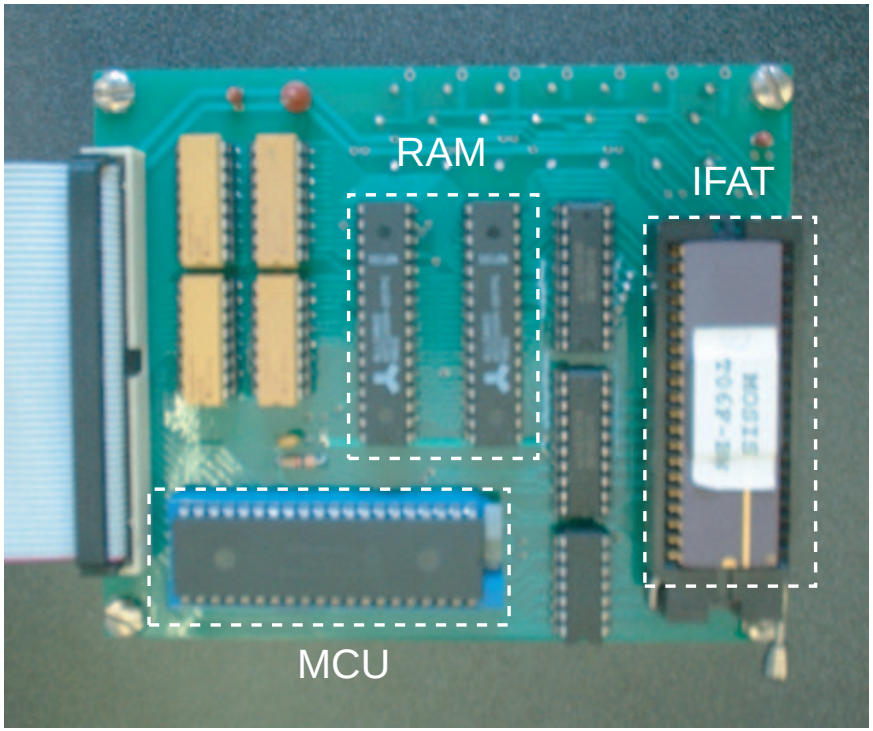


Implementation

IFAT

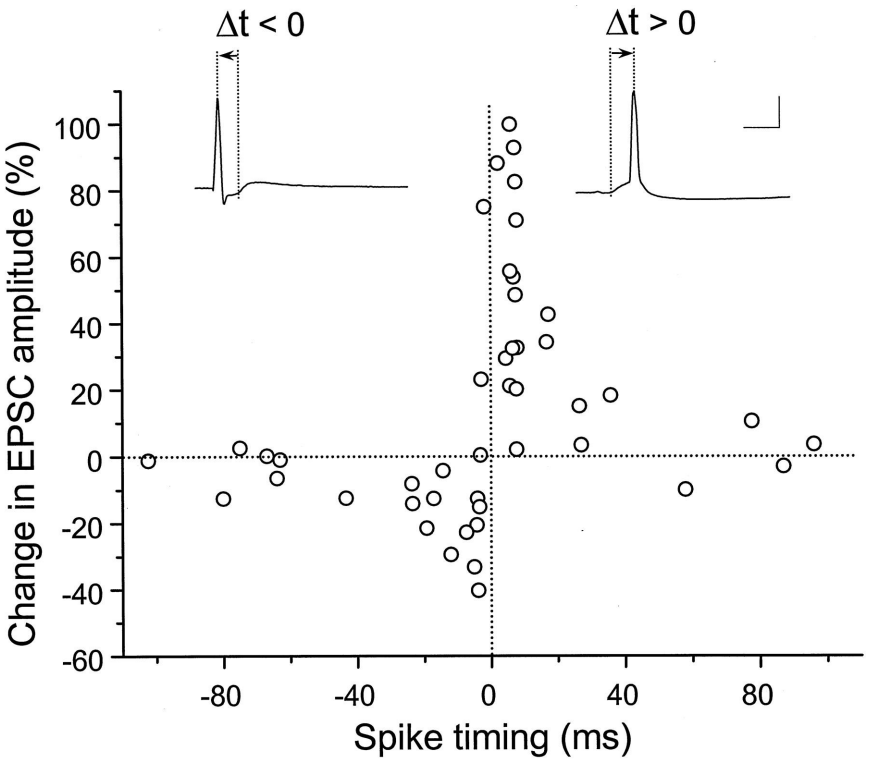


System



Spike Timing-Dependent Plasticity

- In spike timing-dependent plasticity (STDP), changes in synaptic strength depend on the time between each pair of presynaptic and postsynaptic events.
- The most recent inputs to a postsynaptic cell make larger contributions to its membrane potential than past inputs due to passive leakage currents.
- Postsynaptic events immediately following incoming presynaptic spikes are considered to be causal and induce weight increments.
- Presynaptic inputs that arrive shortly after a postsynaptic spike are considered to be anti-causal and induce weight decrements.



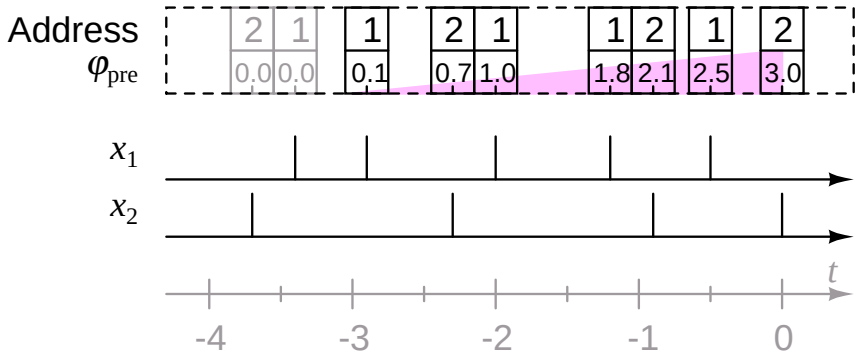
From (Bi & Poo, 1998)

Address Domain STDP: Event Queues

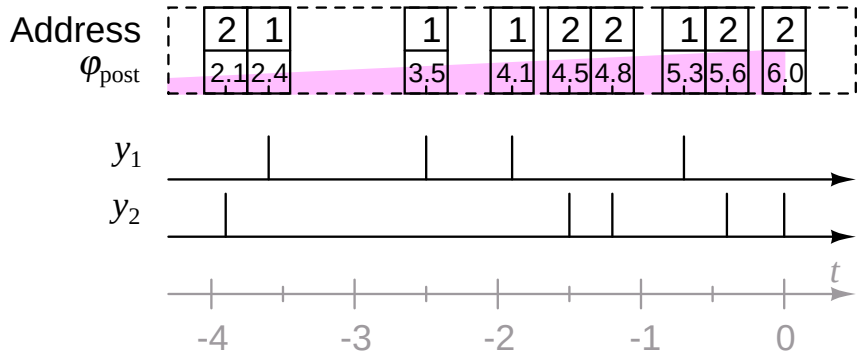
- To implement our STDP synaptic modification rule in the address domain, we augmented our AER architecture with two event queues, one for presynaptic events and one for postsynaptic events.

- When an event occurs, its address is entered into the appropriate queue along with an associated value φ initialized to τ_+ or τ_-. This value is decremented over time.

Presynaptic queue



Postsynaptic queue

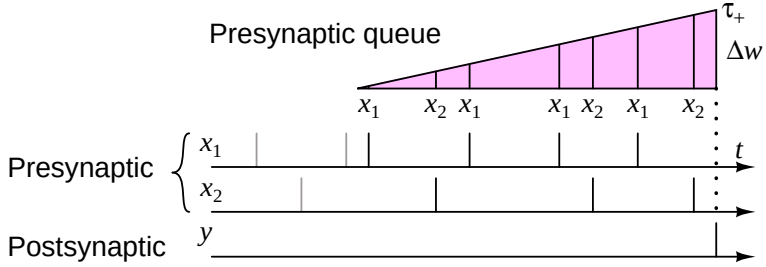


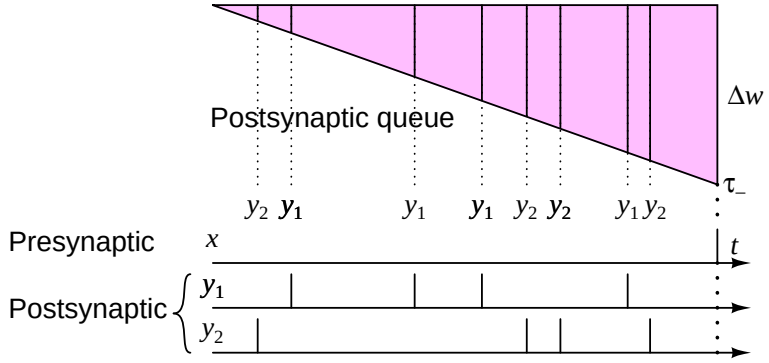
$$\varphi_{\text{pre}}(t-t_{\text{pre}}) = \begin{cases} \tau_+ - (t - t_{\text{pre}}) & \text{if } t - t_{\text{pre}} \geq \tau_+ \\ 0 & \text{if } t - t_{\text{pre}} < \tau_+ \end{cases}$$

$$\varphi_{\text{post}}(t-t_{\text{post}}) = \begin{cases} \tau_- - (t - t_{\text{post}}) & \text{if } t - t_{\text{post}} \geq \tau_- \\ 0 & \text{if } t - t_{\text{post}} < \tau_- \end{cases}$$

Address Domain STDP: Weight Updates

- Weight update procedure:





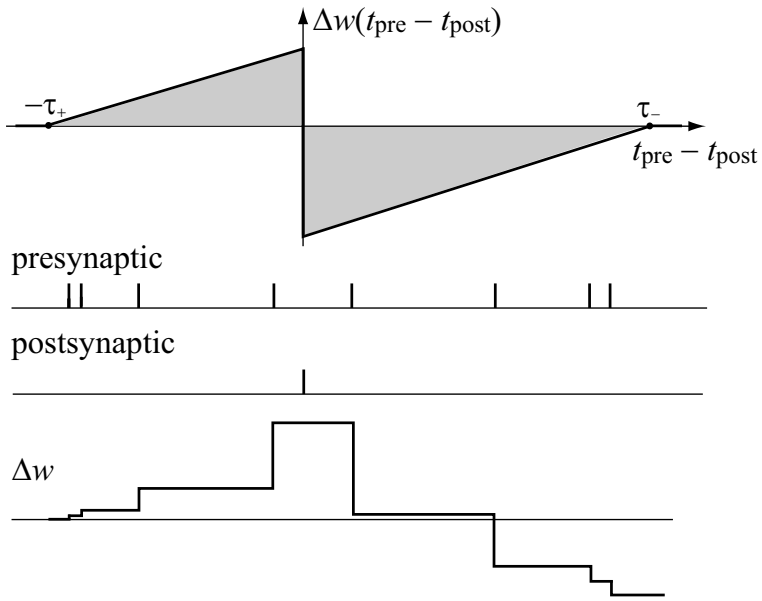
For each postsynaptic event, we iterate backwards through the presynaptic queue to find the causal spikes and increment the appropriate weights in the LUT.

For each presynaptic event, we iterate backwards through the postsynaptic queue to find the anti-causal spikes and decrement the appropriate weights in the LUT.

- The magnitude of the weight updates are specified by the values stored in the queue.

$$\Delta w = \begin{cases} -\eta \cdot \varphi_{\text{post}}(t_{\text{pre}} - t_{\text{post}}) & \text{if } 0 \le t_{\text{pre}} - t_{\text{post}} \le \tau_- \\ +\eta \cdot \varphi_{\text{pre}}(t_{\text{post}} - t_{\text{pre}}) & \text{if } -\tau_+ \le t_{\text{pre}} - t_{\text{post}} \le 0 \\ 0 & \text{otherwise} \end{cases}$$

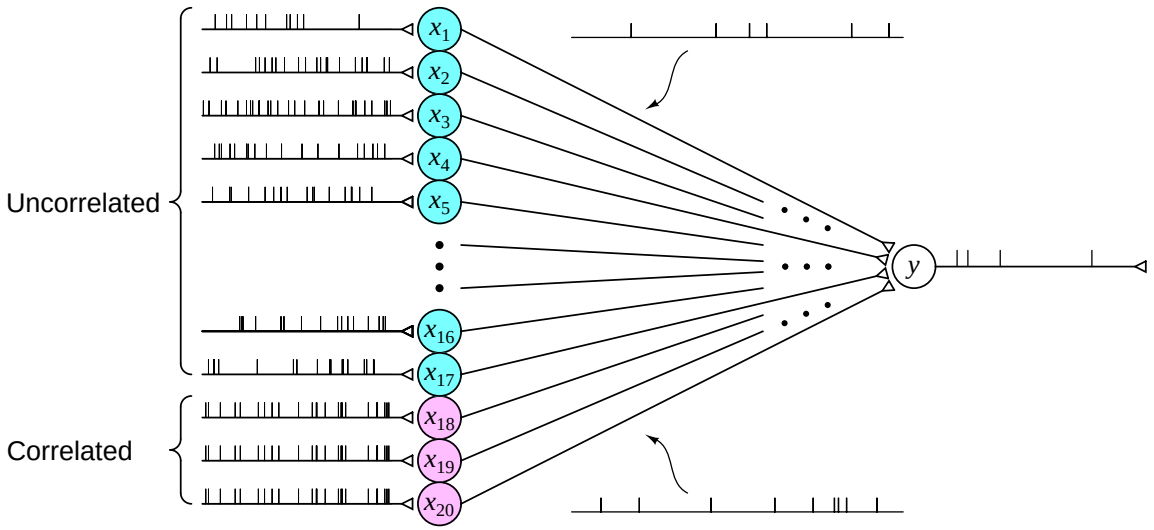
Address Domain STDP: Details



- For stable learning, the area under the synaptic modification curve in the anti-causal regime must be greater than that in the causal regime. This ensures convergence of the synaptic strengths (Song et al., 2000).

- In our implementation of STDP, this constraint is met by setting $\tau_- > \tau_+$.

Experiment: Grouping Correlated Inputs



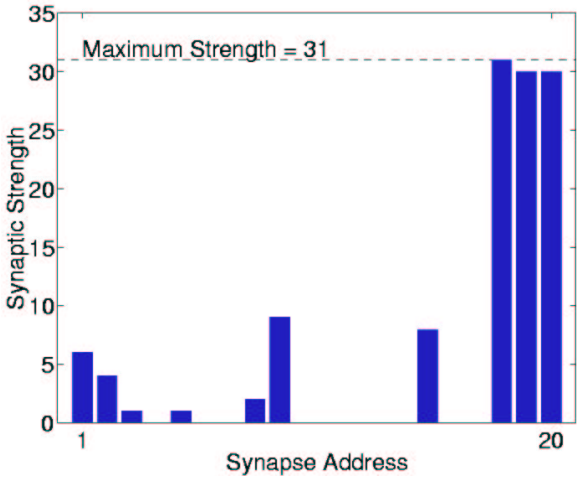
- Each of the 20 neurons in the input layer is driven by an externally supplied, randomly generated list of events.

- Our randomly generated list of events simulates two groups of neurons, one correlated and one uncorrelated. The uncorrelated group drives input layer cells $x_1 \ldots x_{17}$, and the correlated group drives input layer cells $x_{18} \ldots x_{20}$.

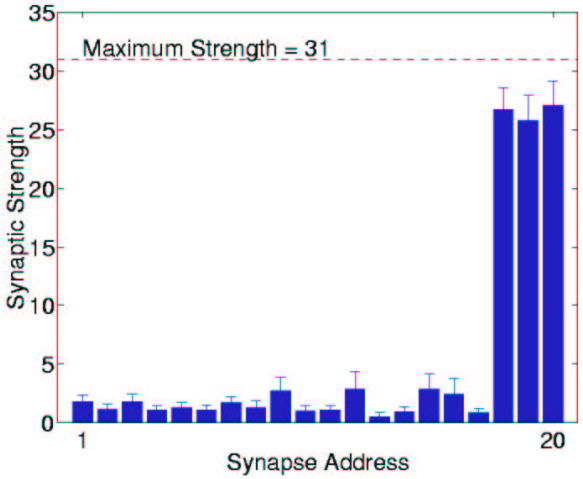
- Although each neuron in the input layer has the same average firing rate, neurons $x_{18} \ldots x_{20}$ fire synchronous spikes more often than any other combination of neurons.

Experimental Results

Single Trial



Average Over 20 Trials



- STDP has been shown to be effective at detecting correlations between groups of inputs (Song et al., 2000). We demonstrate that this can be accomplished in hardware in the address domain.

- Given a random starting distribution of synaptic weights for a set of presynaptic inputs, a neuron using STDP should maximize the weights of correlated inputs and minimize the weights of uncorrelated inputs.

- Our results illustrate this principle when all synaptic weights are initialized to a uniform value and the network is allowed to process 200,000 input events.

Conclusion

- The address domain provides an efficient representation to implement synaptic plasticity based on the relative timing of events.
  - Learning circuitry can be moved to the periphery.
  - The constituents of learning rules need not be constrained in space or time.
- We have implemented an address domain learning system using a hybrid analog/digital architecture.
- Our experimental results illustrate an application of this approach using a temporally-asymmetric Hebbian learning rule.

Extensions

- The mixed-signal approach provides the best of both worlds:
  - Analog cells are capable of efficiently modelling sophisticated neural dynamics in continuous-time.
  - Nearest-neighbor connectivity can be incorporated into an address-event framework to exploit the parallel processing capabilities of analog circuits.
  - Storing the connections in a digital LUT affords the opportunity to implement learning rules that reconfigure the network topology on the fly.
- In the future, we will combine all of the system elements on a single chip. The local embedding of memory will enable high bandwidth distribution of events.