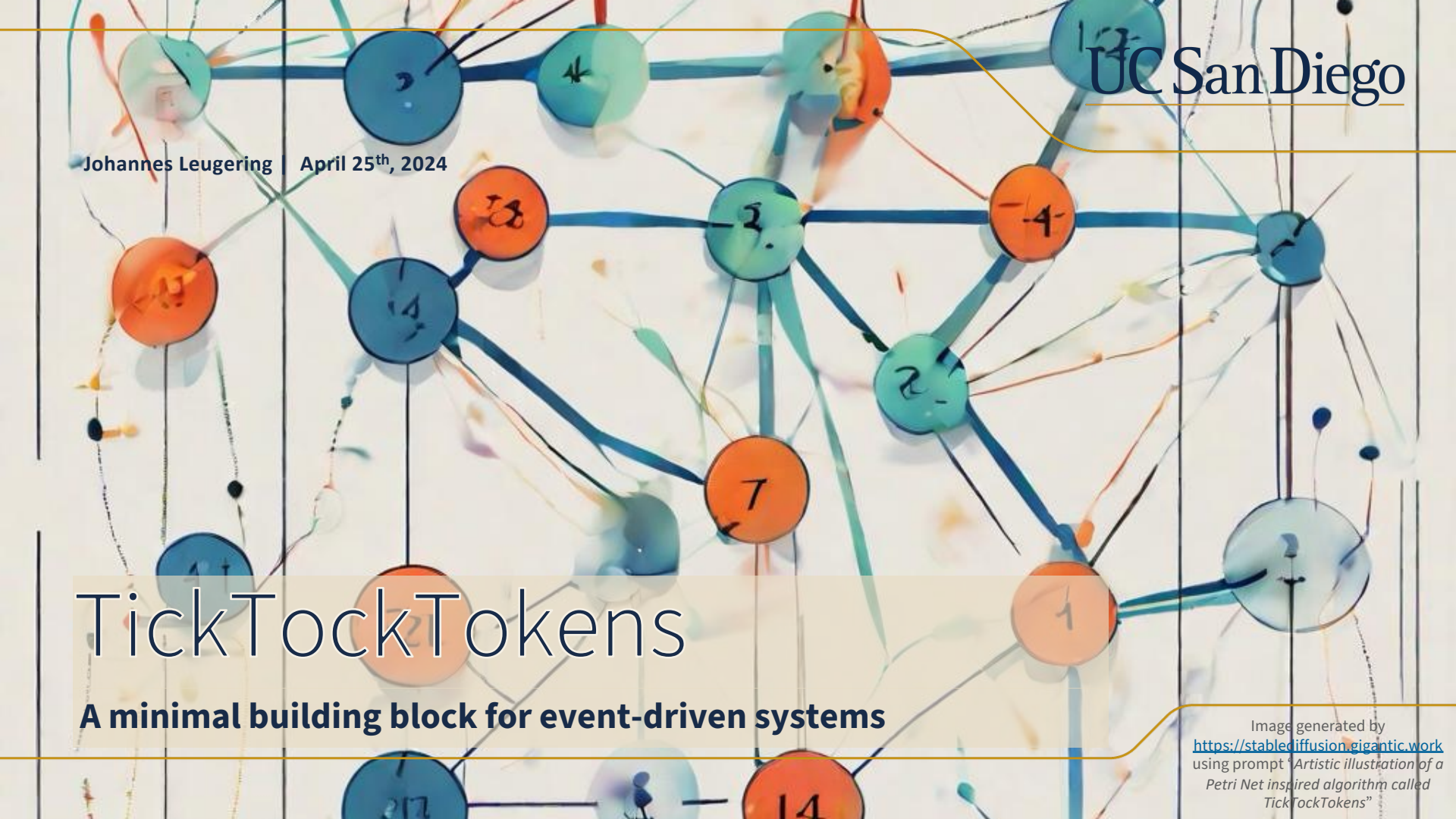


UC San Diego

Johannes Leugering | April 25th, 2024

TickTockTokens

A minimal building block for event-driven systems

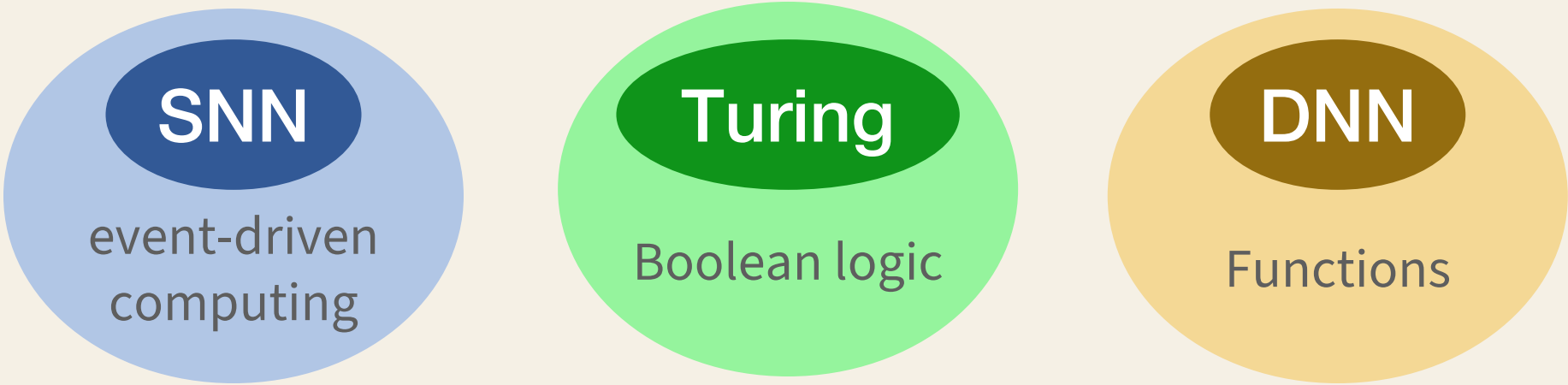
Image generated by https://stablediffusion.gigantic.work using prompt "*Artistic illustration of a Petri Net inspired algorithm called TickTockTokens*"

“[…] Spiking neural networks […] efficient […] event-driven computing […]”

- me, at some point

"[…] Spiking neural networks […] efficient […] event-driven computing […]"

- me, at some point



UC San Diego

What is the equivalent of ***Boolean logic*** for **event-driven** computing?!

"[…] Spiking neural networks […] efficient […] event-driven computing […]"

- me, at some point

UC San Diego

Computing with events

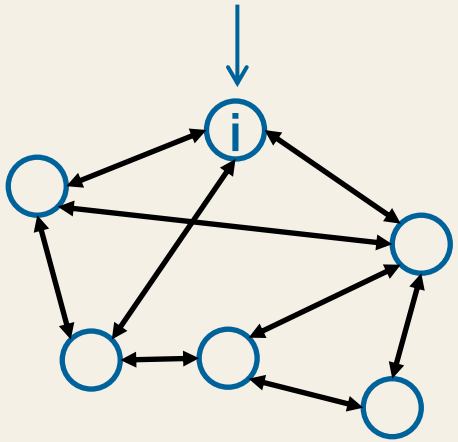
A very simple proposal

- System of interconnected *nodes*
- Each node:
  - *binary "visible" state*
  - state changes communicated via *events**

    **Event: message at time t, containing only #ID*
  - *local, hidden state + dynamics*

"Hidden" state: $\left(c_i^{good} \in \mathbb{N}, c_i^{bad} \in \mathbb{N}, T \in \mathbb{R}\right)$

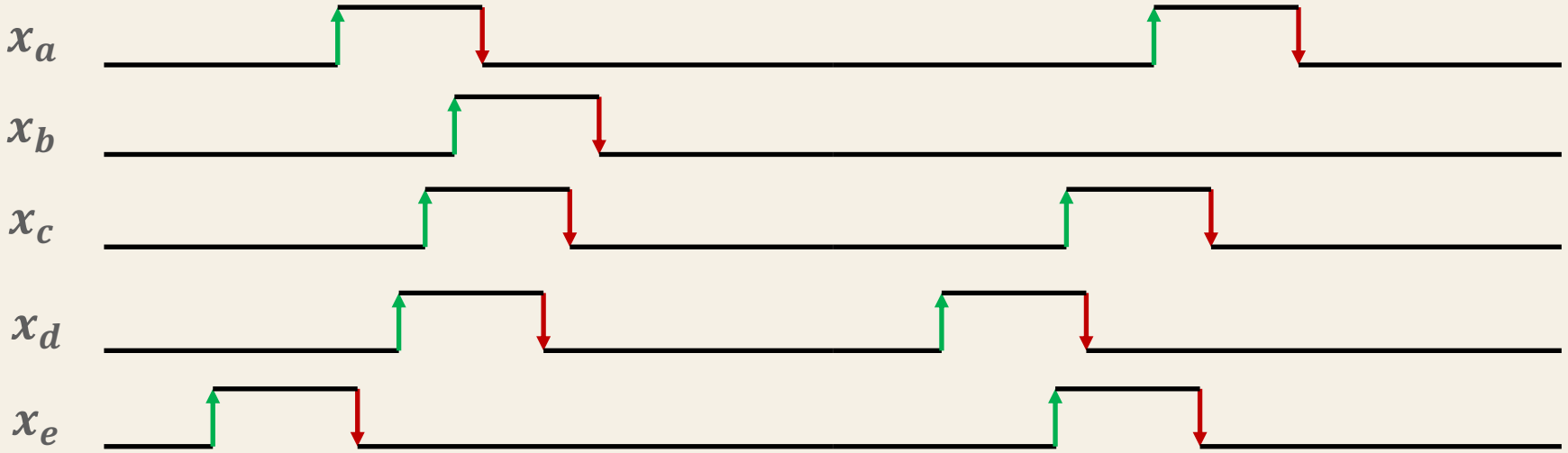
"Visible" state: $x_i = f(c_i^{good}, c_i^{bad}, T) \in \{\text{on}, \text{off}\}$



UC San Diego

Computing with events

Boolean conditions for switching



UC San Diego

Computing with events

Boolean conditions for switching

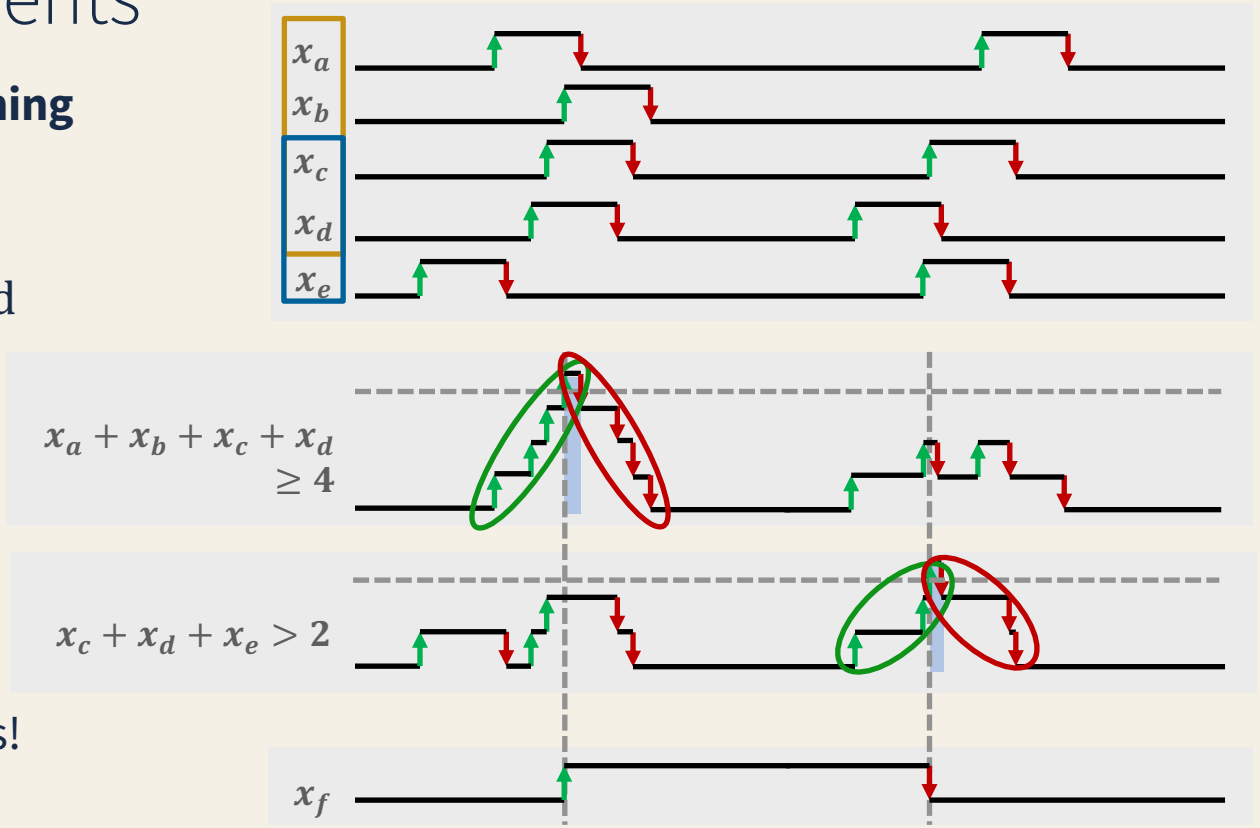
- Turn **on** for **good** input:

$$c_i^{good} = \sum_j w_{ij}^{g} \cdot x_j \geq \theta_{good}$$

- Turn **off** for **bad** input:

$$c_i^{bad} = \sum_j w_{ij}^{b} \cdot x_j > \theta_{bad}$$

- → S-R latch with logic
- Just count up & down edges!
- But: not *"autonomous"* yet!



UC San Diego

Computing with events

Internal timeout → time-scale

- When turning **on**:

$$T \leftarrow \tau$$

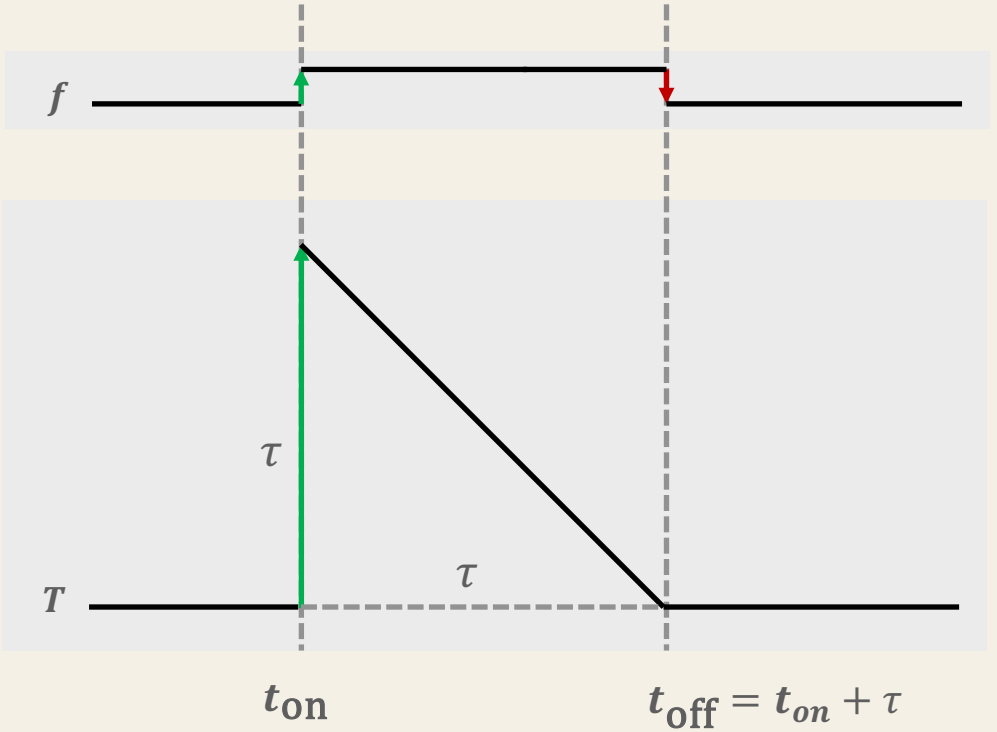
- Internal **countdown** timer:

$$\frac{d}{dt}T = -1 \quad \text{or} \quad \frac{d}{dt}T = -f(T)$$

- Turn **off** if:

$$T \leq 0 \quad \text{or} \quad T \leq \Theta$$

- Autonomous "**off**" events!
- *"async. monoflop with reset"*



UC San Diego

TickTockTokens

Single node

- Single node: input pulse → output pulse

UC San Diego

TickTockTokens

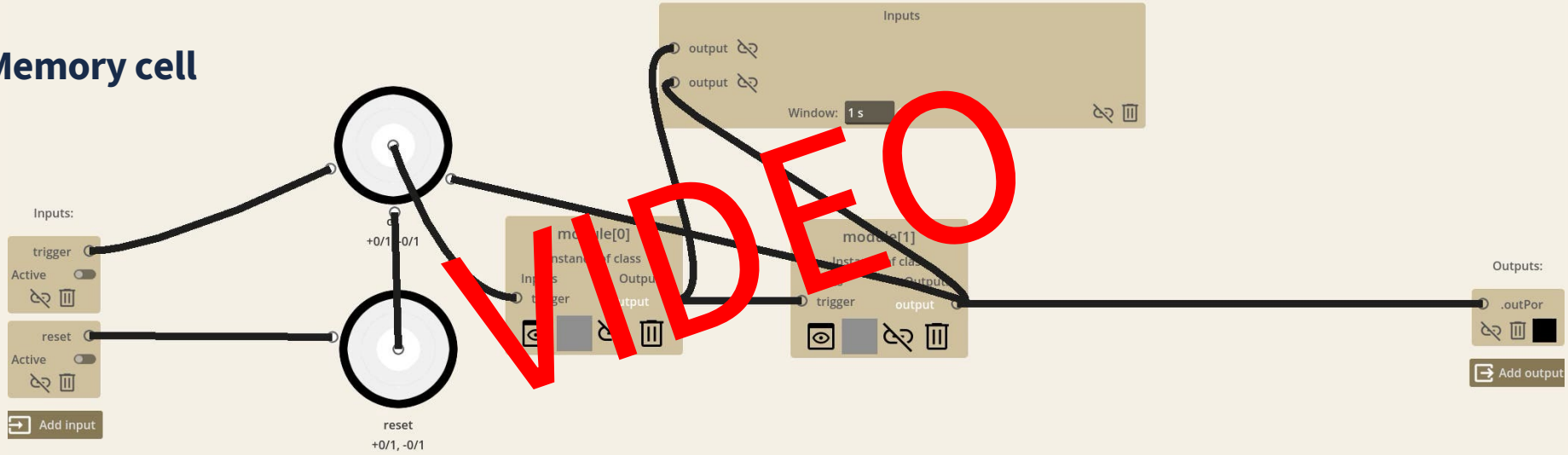
Generic pulse generator

- Three nodes: pulse generator
- Input pulse → output pulse with fixed delay & duration

UC San Diego

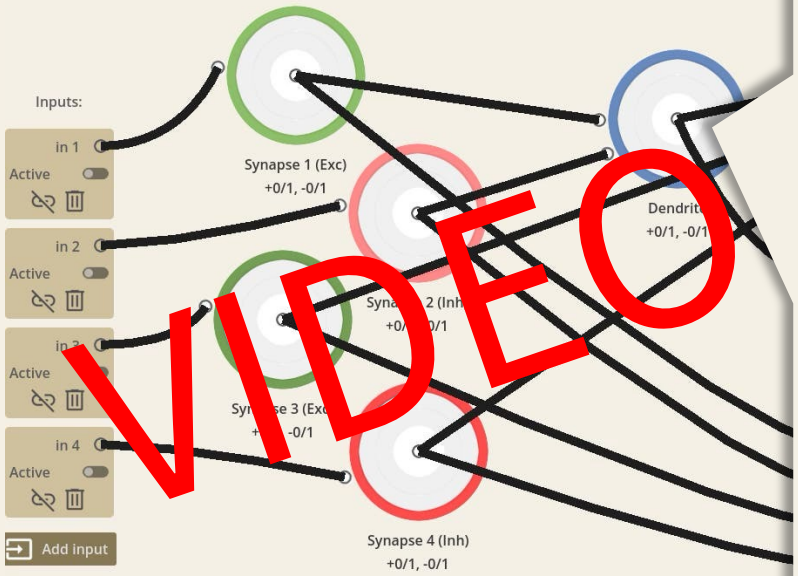
TickTockTokens

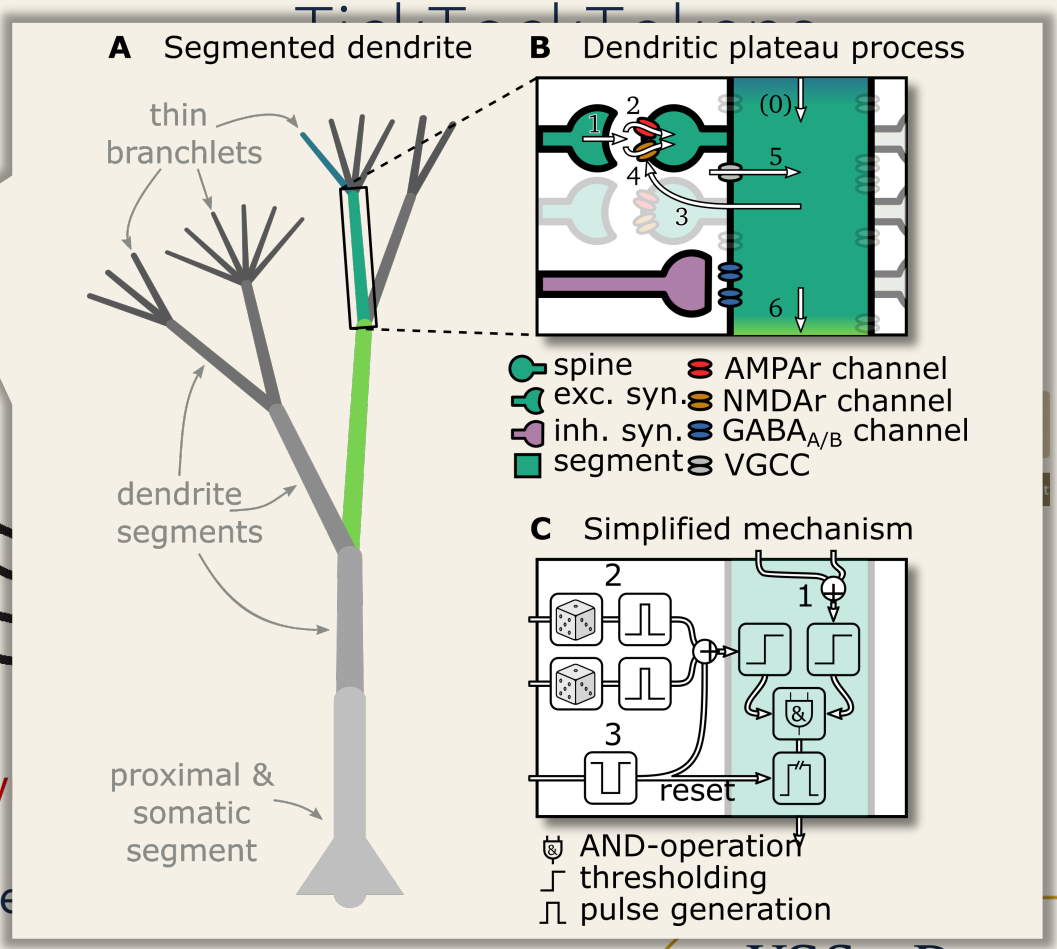
Memory cell



- Two cross-coupled pulse generators = oscillator/memory
- Exc. input pulse → starts oscillation / sets memory
- Inh. Input pulse → stops oscillation / resets memory

UC San Diego





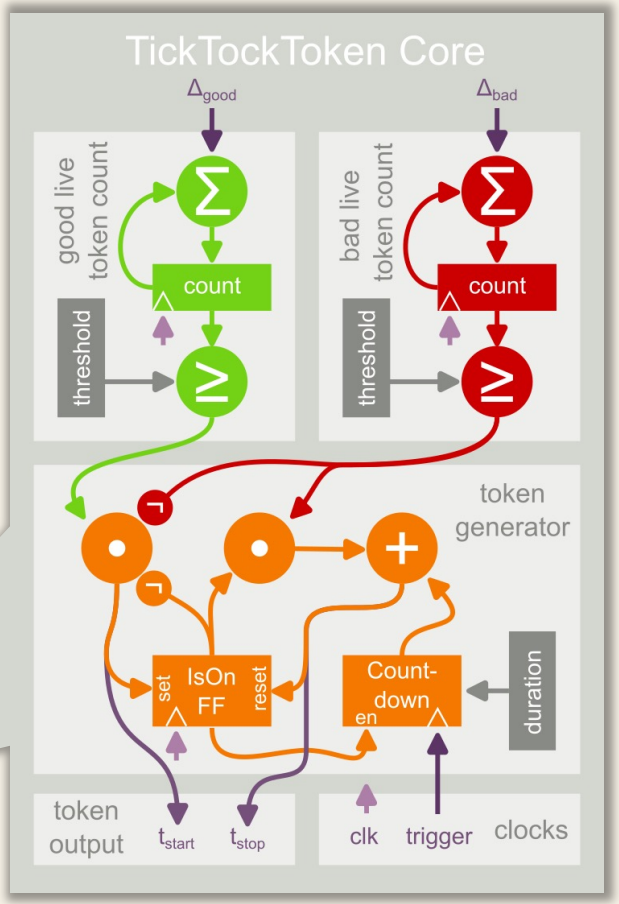
- 6 nodes = 2 exc. synapses, 2 inh. sy
- Implements a simple 2-compartme

UC San Diego

TickTockTokens

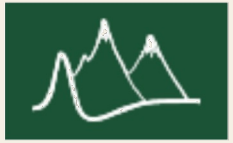
A very simple (minimal?) model

- Turn **on** $(x_j \leftarrow 1, T \leftarrow \tau)$ if $\mathrm{c_{on}} \wedge \neg \mathrm{c_{off}}$
- Turn **off** $(x_j \leftarrow 0)$ if $\mathrm{c_{off}} \wedge \neg \mathrm{c_{on}}$
- $\mathrm{c_{on}} = \neg \mathrm{x_j} \wedge \sum_i w_{ji}^{\mathrm{g}} \cdot x_i \geq \theta_{\mathrm{good}}$
- $\mathrm{c_{off}} = \mathrm{x_j} \wedge \left(\sum_i w_{ji}^{\mathrm{b}} \cdot x_i > \theta_{\mathrm{bad}} \vee T \leq 0 \right)$
- one “hidden layer” → arbitrary constraints (→ paper)
- Could be implemented in
  - Analog electronics?!
  - Digital electronics → Tiny(Neuromorphic)Tapeout



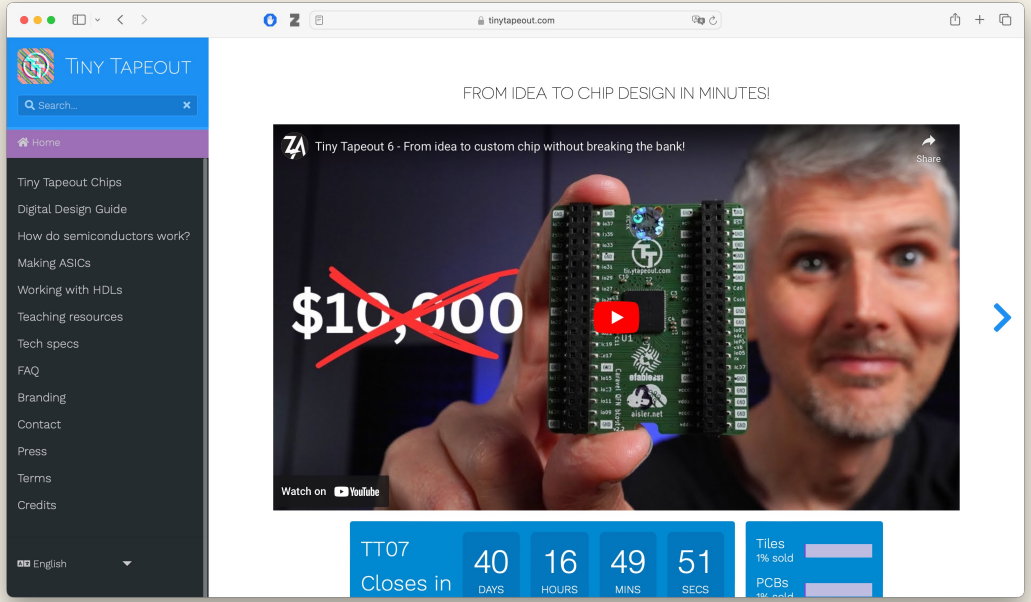
UC San Diego

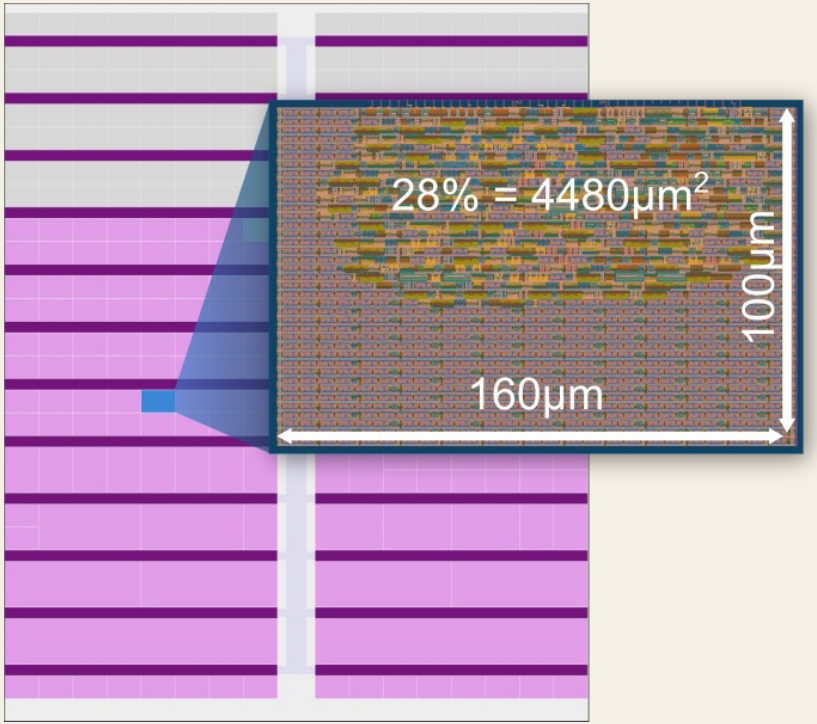
TickTockTokens



Telluride 2023

Implementation in hardware





Design #7 of TinyTapeout 5 (ETA June 15th)
(https://tinytapeout.com)

UC San Diego

Key take-aways

- ~~Monoflops are all you need!~~
- discrete state → Δstate → events
- an LIF neuron is ***almost*** enough, but:
- event-based processing needs autonomous dynamics
  → at minimum: monoflop
  - maybe rebound spikes?
  - multiple neurons?
  - multiple time-constants?
  - …
- *(TickTockTokens as building block?)*

Thank you!

INTEGRATED SYSTEMS NEUROENGINEERING

Telluride 2023

TickTockTokens paper:

https://jleugeri.github.io/docs/ticktocktokens.pdf

HW @ GitHub:
https://github.com/jleugeri/tnt-ticktocktokens

Simulator @ GitHub:

https://github.com/jleugeri/ticktocktokens

Dendrite paper:

https://tinyurl.com/dendritic-computing