Phase 1: Gather Spikes

Goal: Find which synapses need to be processed

PHASE 1a: External Events (Axon Spikes). ONLY COLLECTS ADDRESSES (POINTERS); synapse data not read yet
- external_events_processor reads Present BRAM
- Generates exec_bram_spiked[15:0] mask
  -
- hbm_processor reads axon pointers from HBM Region 1
  -
- pointer_fifo_controller demuxes pointers → 16 FIFOs
  -

do phase 1a and 1b happen one hbm row at a time? are they in parallel (as in 1a and 1b happen at the same time, or hbm rows simultaneously)
- No, phase 1a and 1b are sequential.
- All HBM rows are fetched at a time, but rows are processed one at a time.

PHASE 1b: Internal Events (Neuron Spikes from previous cycle)
- internal_events_processor reads URAM states / membrane potentials
- Generates exec_uram_spiked[15:0] mask
  -
- hbm_processor reads neuron pointers from HBM Region 2
  -
- pointer_fifo_controller continues filling 16 FIFOs
  -

Output: 16 pointer FIFOs filled with addresses

Phase 2: Synaptic Processing & Neuron Updates

Goal: Read synapses, accumulate weights, update neurons

- pointer_fifo_controller: Round-robin read from 16 FIFOs
- hbm_processor: Pop pointer → read synapses from Region 3
- hbm_processor: Parse synapses → forward to IEP
- internal_events_processor: Accumulate weights into neurons
- internal_events_processor: Apply neuron model (leak, etc.)
- internal_events_processor: Threshold check → spike?
- internal_events_processor: Write updated state to URAM
- Spikes → spike_fifo_controller → command_interpreter

Output: Updated neuron states in URAM, new spikes generated

If neurons spiked: Loop back to Phase 1b
If no more spikes: Timestep complete

Spike Output

1) spike_fifo_controller round-robbin muxes spikes from individual spk#FIFOs to spk2ci FIFO
2) command_interpreter takes spikes from spk2ci FIFO and batches spikes into 512-bit packets; places them into tx FIFO
- *Example packet format:*
  - [511:480] = 0xEEEEEEEE (spike output opcode)
  - [479:463] = spike_0 (17-bit neuron ID)
  - [462:446] = spike_1 (17-bit neuron ID)
  - ...
  - [#:0] = padding (unused for < 14 spikes)

3) pcie2fifo takes tx FIFO packets and pushes them thrugh PCIe to Host

AXON IDs

- Each axon name (like 'a0') is registered in the connectome during compilation
- The connectome assigns each axon a **core type index** (sequential numbering)
- *Example:* 'a0'→0, 'a1'→1, 'a2'→2
- Host Side (Python):
  - Creates one-hot encoding for axons
    -
  - Packs into 512-bit PCIe TLP packets sent to pcie2fifos
  - Each packet contains 32 × 16-bit "axon events"
- pcie2fifos.v recieves 512 bit PCIe TLP and sends to rxFIFO
- command_interpreter takes rxFIFO and unpacks 512-bit TLP into 32, 16 bit chunks (aka rows of BRAM) into Future BRAM.

KEY TERMS

GROUP (neuron group) = a collection of neurons managed by a single URAM bank. The hardware has 16 URAM banks (URAM0-URAM15), so there are 16 groups numbered 0-15. Each group can hold approximately 8,192 neurons (131,072 total ÷ 16 banks). When we say "group 0 has spikes," we mean neurons stored in URAM bank 0 have spiked.

MASK (spike mask) = a bit vector where each bit indicates whether a group has spiking activity.
- *Example:*
  - exec_bram_spiked[15:0] = 0x0007 means bits 0, 1, and 2 are set, indicating groups 0, 1, and 2 have spiking neurons. Each bit corresponds to one of the 16 URAM banks (neuron groups)

DEMUXING = Taking one input and routing it to one of multiple outputs based on a selector.
- In Phase 1, `pointer_fifo_controller` demuxes HBM data (containing 16 pointers) to 16 separate pointer FIFOs based on the spike mask.

MUXING = Taking multiple inputs and selecting one as the output.
- In Phase 2, `pointer_fifo_controller` muxes (round-robin reads) from 16 pointer FIFOs to select which pointer to process next.

ROUND-ROBIN READING = A fair scheduling algorithm that cycles through resources (typically neuron groups in FIFO) in sequential order.
- The `pointer_fifo_controller` reads from ptr0 → ptr1 → ptr2 → ... → ptr15 → ptr0 (looping back), ensuring all groups get equal processing time.

FIFO = A queue data structure where the first item added is the first item removed. The hardware uses multiple types of FIFOs:
- **Pointer FIFOs (ptr0-ptr15)** = 16 FIFOs that store 32-bit HBM addresses pointing to synapse data locations. Each FIFO corresponds to one neuron group.
- **Spike FIFOs (spk0-spk7)** = 8 FIFOs that store 17-bit neuron addresses (neuron IDs that spiked). These aggregate spikes before sending to the host.
- **rxFIFO/txFIFO** = PCIe communication buffers (512-bit wide) for receiving commands from and sending results to the host.

One-Hot / Multi-Hot Encoding = A binary representation using a fixed range of bits, where each bit position corresponds to one item in a set. The bit value (0 or 1) indicates whether that item is active.
- In the FPGA, spike masks use 16 bits for 16 neuron groups:

Bit positions: [15][14][13]...[4][3][2][1][0]
Neuron groups: 15 14 13 ... 4 3 2 1 0

- *Example:* If bits 3 and 4 are set to 1:
-
exec_bram_spiked = 0x0018 (binary: 0000000000011000)
↑↑
bit 4 bit 3

Meaning: Groups 3 and 4 have spiking neurons
Groups 0, 1, 2, 5-15 have no spikes