Phase 1: Gather Spikes

Goal: Find which synapses need to be processed (aka which presynaptic axons or neurons spiked and their outgoing connections).
==> *returns synapse pointers across the 16 pointer FIFOs*

PHASE 1a: External Events (Axon Spikes)
- *external_events_processor* reads Present BRAM & generates exec_bram_spiked[15:0] mask
- *hbm_processor* prepares axon pointers from HBM Region 1
- *pointer_fifo_controller* demuxes pointers → 16 FIFOs

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* prepares neuron pointers from HBM Region 2
- *pointer_fifo_controller* continues filling 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

KEY TERMS

GROUP (neuron group) = a collection of nneurons 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*:
  - execu_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 (typicallly 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.

HBM - 8 GB

Region 1: Axon Pointers
- Per Axon:
  - Where axon's synapse is stored in HBM
  - How many rows this axon's synapses take in HBM (how many sets of 8 or less synapses)
- Each row holds 8 pointers with 32 bits = 32 bytes per row
- Base Address: 0x0000
- The Start address in the pointer is *relative* to the synapse base address.
- Each axon gets its own row(s)
- Multiple axons ' synapses cannot share a row

Region 2: Neuron Pointers
- Per Neuron:
  - Where neuron's synapse is stored in HBM
  - How many rows this neuron's synapses take in HBM
- Base Address: 0x4000
- Output neurons also have synapse addresses (explained in region 3)
- The Start address in the pointer is *relative* to the synapse base address.
- Each neuron gets its own row(s) for synapses
- Multiple neurons' synapses cannot share a row

Region 3: Synapses
- Each HBM row stores 8 synapses
- Each synapse is 32 bits [OpCode | Address | Weight]
- Base Address: 0x8000
- Synapse format (32 bits per synapse):

[31:29] OpCode (3 bits):
000 = Regular synapse (send spike to another neuron)
100 = Output spike entry (send spike to host)

[28:16] Target Address in URAM (13 bits):
For OpCode=000: Index of target neuron
For OpCode=100: Index of neuron to report (same as source)

[15:0] Weight (16 bits):
Signed fixed-point value

FPGA

Configurable Logic

2586 CLBs

hbm_processor.v

PHASE 1a:
- reads active axon pointers from HBM Region 1 how does it know where the axon pointers are for the active axons and when to read them / that it's phase 1? State machine variables
- *example*:
  - Axon a0 pointer:
    - Read HBM(Region1+0): Pointer = {length: 1 row, start: 0x8000}
    - Meaning "a0's synapses are at HBM[0x8000]"

PHASE 1b:
- reads neuron pointers of spiked neurons from HBM Region 2

pointer_fifo_controller.v

START HERE

PHASE 1a:
- demuxes the masked pointers collected by hbm_processor
- Input
  - 512-bit HBM data with 16 pointers
  - *example*
    - exec_bram_spiked[15:0] = 0x0007 (group 0 only)
  - mask sent in by external_events_processor
- Output
  - write active axon pointers to ptr# FIFO (others unused)

PHASE 1b:
- demuxes the masked pointers collected by hbm_processor
- Input
  - 512-bit HBM data with 16 pointers
  - *example*
    - exec_bram_spiked[15:0] = 0x0007 (group 0 only)
  - mask sent in by internal_events_processor
- Output
  - write active axon pointers to ptr# FIFO (others unused)

external_events_processor.v

PHASE 1a:
- reads present BRAM
- *example*:
  - Row 0 data: 0x00...00000007
  - Detected: Bits 0,1,2 are set
  - Output:
    - electrical wirte signal to pointer_fifo_controller = exec_bram_spiked[15:0] = 0x0007
    - Neuron group 0 has 3 active axons)

BRAM - 1 MB

USER INPUTS

FUTURE BUFFER
- 8192, 16-bit rows
  - 131072 axons
- which axons will fire in the next timestep
- Inputs initially sent here

PRESENT BUFFER
- 8192, 16-bit rows
  - 131072 axons
- which axons firing in this timestep

FIFO

ptr#
- stores 32-bit HBM Region 3 addresses (pointers to synapse data)
- used for Phase 1 spikes

ENDS HERE

When execute() is called, Future Buffer and Present Buffer get swapped: meaning the incoming input spikes becoming the Present Buffer spikes for the next timestep.

internal_events_processor

PHASE 1b:
- Reads URAM states & applies threshold
  - if V >= threshold --> mark as spiked
- Output: exec_uram_spiked[15:0] mask
- *Example*:
  - If h0, h1, h2, h3, h4 all spiked:
    - exec_uram_spiked[0] = 0x001F (bits 0-4 set)
    - exec_uram_spiked[1-15] = 0x0000 (no neurons in groups)

URAM - 4.5 GB
- Stores current timestep's **Membrane Potentials**
- Initially zeroed during clear(), then updated at runtime as neuron integrates inputs
- Organization:
  - 16 banks, each storing handling 4096, 72-bit words, each word handling the 36 bit membrane potentials of 2 neurons = 8,192 neurons per bank
- *Example*:
  - Word 0 (address 0x000):
    - Bits [35:0]: Neuron 0 (h0) membrane potential = 0x0_0000_0000 (V=0)
    - Bits [71:36]: Neuron 1 (h1) membrane potential = 0x0_0000_0000 (V=0)
- *Addresses*:
  - Full 17-bit Neuron Address Structure:
    - [16:13] = neuron group (0-15) → selects which of 16 URAM banks
    - [12:1] = word address (0-4095) → selects which 72-bit word in that bank
    - [0] = upper/lower (0=lower, 1=upper) → selects which neuron in the word
  - 13-bit HBM Synapse Address (stored in Region 3):
    - [12:1] = word address (0-4095) → bits [12:1] of full address
    - [0] = upper/lower neuron → bit [0] of full address
    - The group field [16:13] is NOT stored because the hardware uses parallel processing: Each of the 16 neuron groups gets its own dedicated 13-bit address from the 512-bit HBM row simultaneously.