Introduction to Communicating Events in Neuromorphic Chips

Melika Payvand

Dec 16th. 2019

The material for the lecture is from two sources:

1) Liu, S. C., Delbruck, T., Indiveri, G., Whatley, A., & Douglas, R. (Eds.). (2014). *Event-based neuromorphic systems*. John Wiley & Sons.

2) Sparsø, Jens. "Asynchronous circuit design-a tutorial." Chapters 1-8 in" Principles of asynchronous circuit design-A systems Perspective". Kluwer Academic Publishers, 2001

The challenge

- Neurons generate action potentials which can be treated as all-or-none events called action potentials (spikes)

- In the cortex, a neuron can deliver a spike to thousands of destinations

- Neuromorphic electronics systems must embed complex networks of neurons, axons and synapses built in 3D into a 2D silicon substrate.

→ **What can we do?**

Solution: Leveraging the fast CMOS digital gates

- Modern Digital gates have picoseconds delays >> time constants of the neurons
- The mismatch between the logic-optimized and the required interconnectivity can be overcome by using time-multiplexed communication.
- Similar to communication networks?

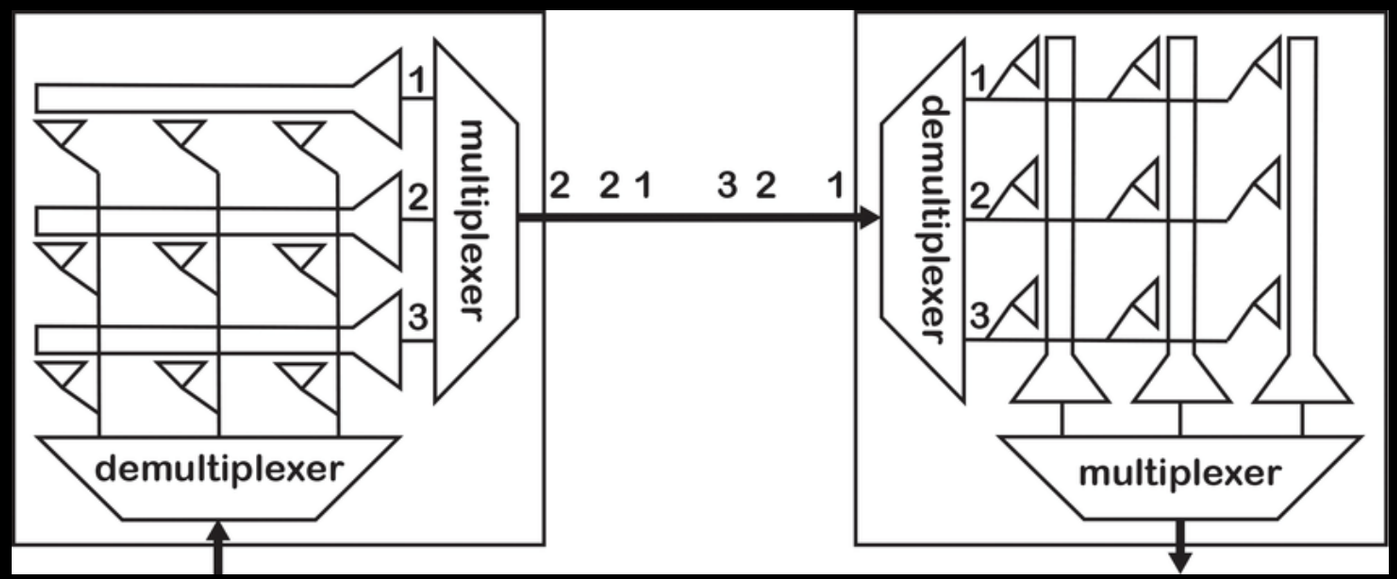
Communication networks

- Circuits switched:
  - Two end-points communicate by setting up a virtual circuit
  - This virtual path is dedicated for the communication between the two end points.
  - Once the communication ends the virtual circuit is destroyed and the hardware resources used by the circuit are released for use by another visual circuit
    - —> set-up ,tear down cost
- Packet switched:
  - Operate by time-multiplexing segments of the network
  - Instead of creating an end-to-end path, each item being communicated (packet) request access to shared resources on the fly.
  - Efficient when communication between two end points is in bursts of small amounts of information —> ideal for delivering spikes between neurons: small information

Time: information to be transferred

- The time at which a spike occurred is the information to be transferred
- Representation of time is not a trivial task:
  - Discrete time?
    - Time is discretized onto global ticks: communication network delivers time-stamped spikes at the global time stamp
  - Continuous time: Time represents itself: if we have these assumptions:
    - Neurons' spiking rate is very low (tens of Hz) compared to the speed of digital electronics (GHz)
    - Axonal delays are in the order of ms compared to the switching delay of gates (tens of picoseconds)
    - A very small variation (<0.1%) in spike arrival time should not have a significant impact on the overall system behavior

Address-Event Representation



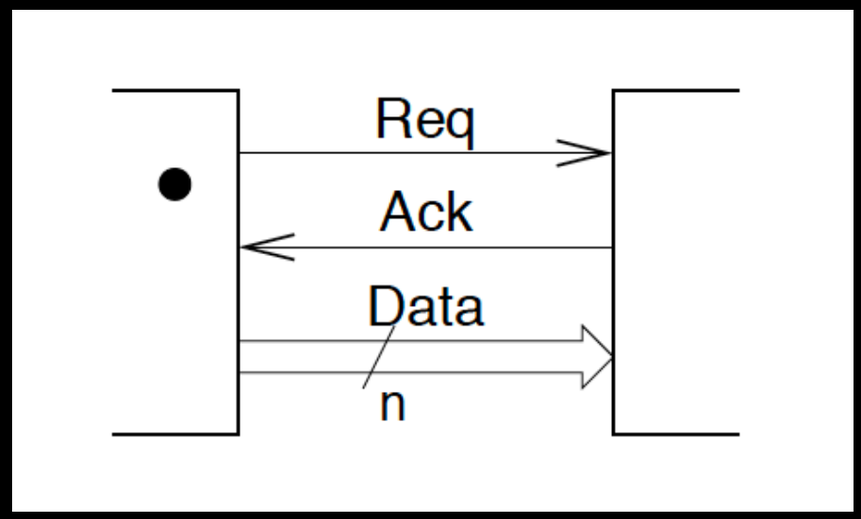
- First proposed in 1991 in Carver Mead's lab (Lazzaro, Mahowald)
- Function: Provide multiplexing/demultiplexing functionality for spikes
- Spikes are generated asynchronously when the neuron fires
- The communication bus accepts the spikes and places the encoded address of the spiking neuron on the bus at the time of the spike.
- The decoder at the receiver decodes the address and delivers it to the destination
- Since multiple neurons can fire at the same time, an arbiter is required to give permission to the neurons to use the shared bus

Asynchronous Communication Fundamentals

- Shared bus: the bus should grant permission for access: How?
- Whenever the neuron spikes, it asks for the access (Request) and if the bus is free, the request gets acknowledged and the spikes is placed on the bus
- This is done through a hand-shaking mechanism

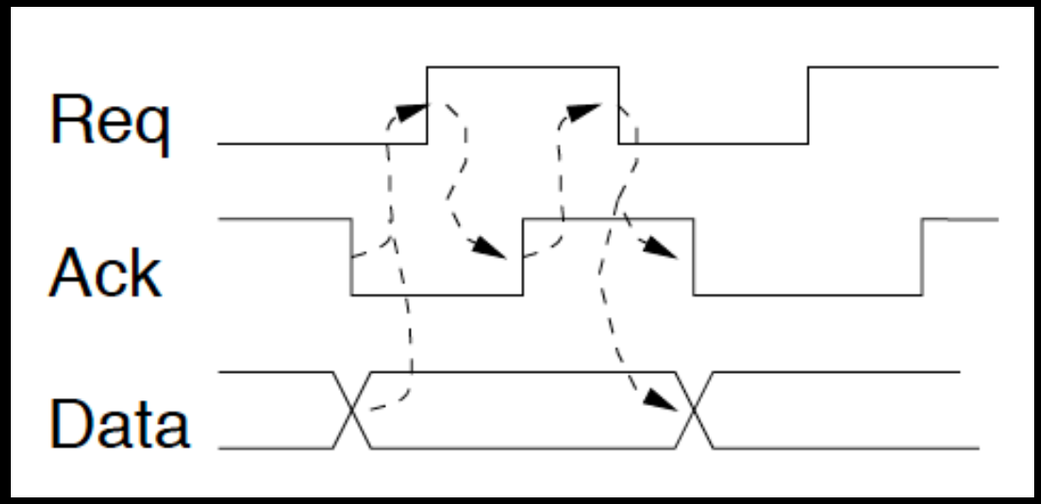
Handshaking protocols

- Bundled-data protocol:
  - data signals use Boolean levels to encode information
  - separate request and acknowledge wires are bundled with the data signals
  - One example is the 4 phase handshaking protocol



4-phase handshaking

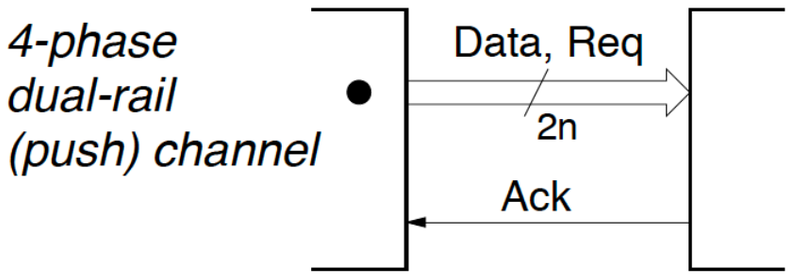
- The sender issues data and sets the request to high
- The receiver absorbs the data and sets acknowledge to high
- The sender responds by pulling the request to low (data is no longer valid)
- The receiver acknowledges by setting the acknowledge low



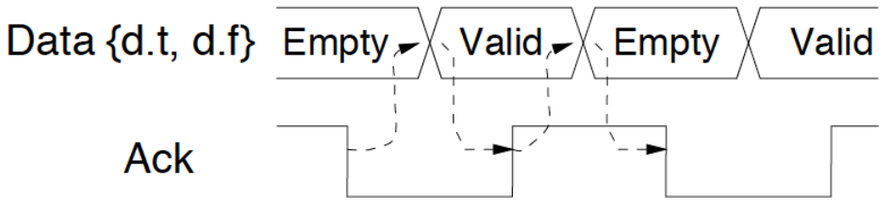
Delay sensitivity

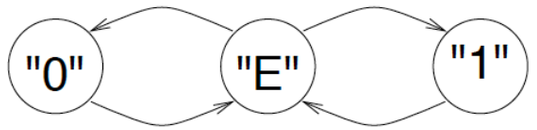
- All the bundled-data protocols rely on delay matching, such that the order of signal events at the sender's end is preserved at the receiver's end —> delay sensitive
- Have to be very careful with routing, inserting delays
- Or take alternative designs that are delay insensitive

Delay-insensitive design: 4-phase dual rail



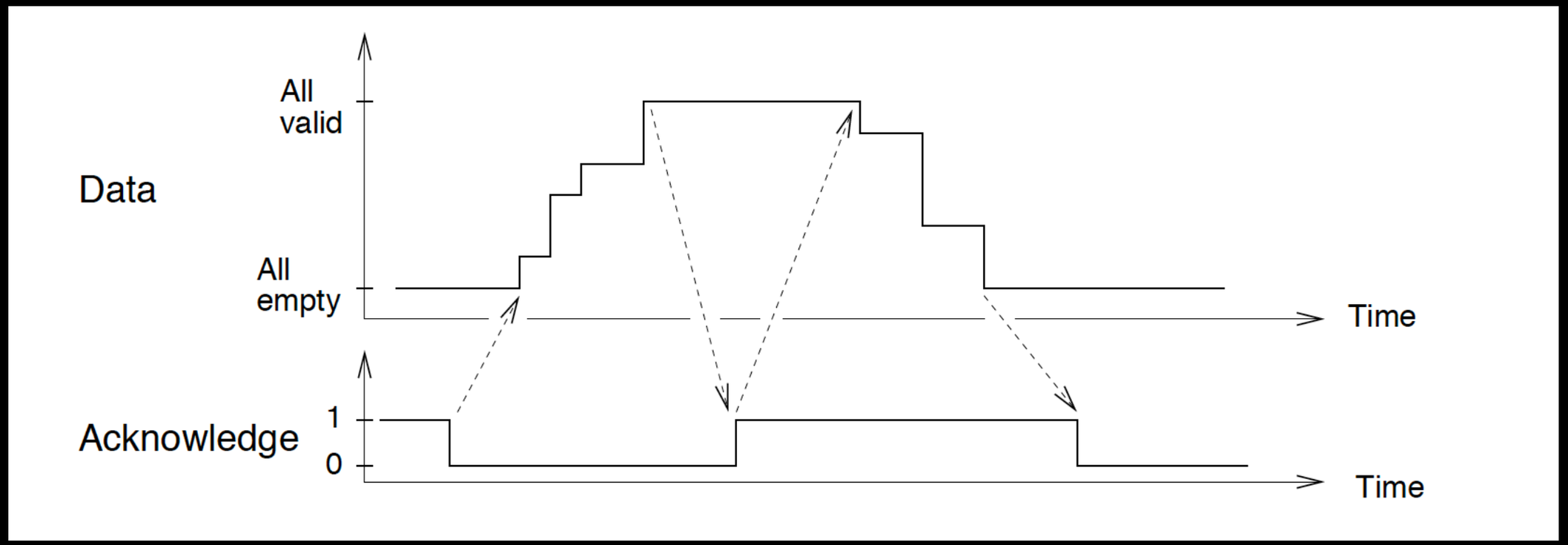
	d.t	d.f
Empty ("E")	0	0
Valid "0"	0	1
Valid "1"	1	0
Not used	1	1





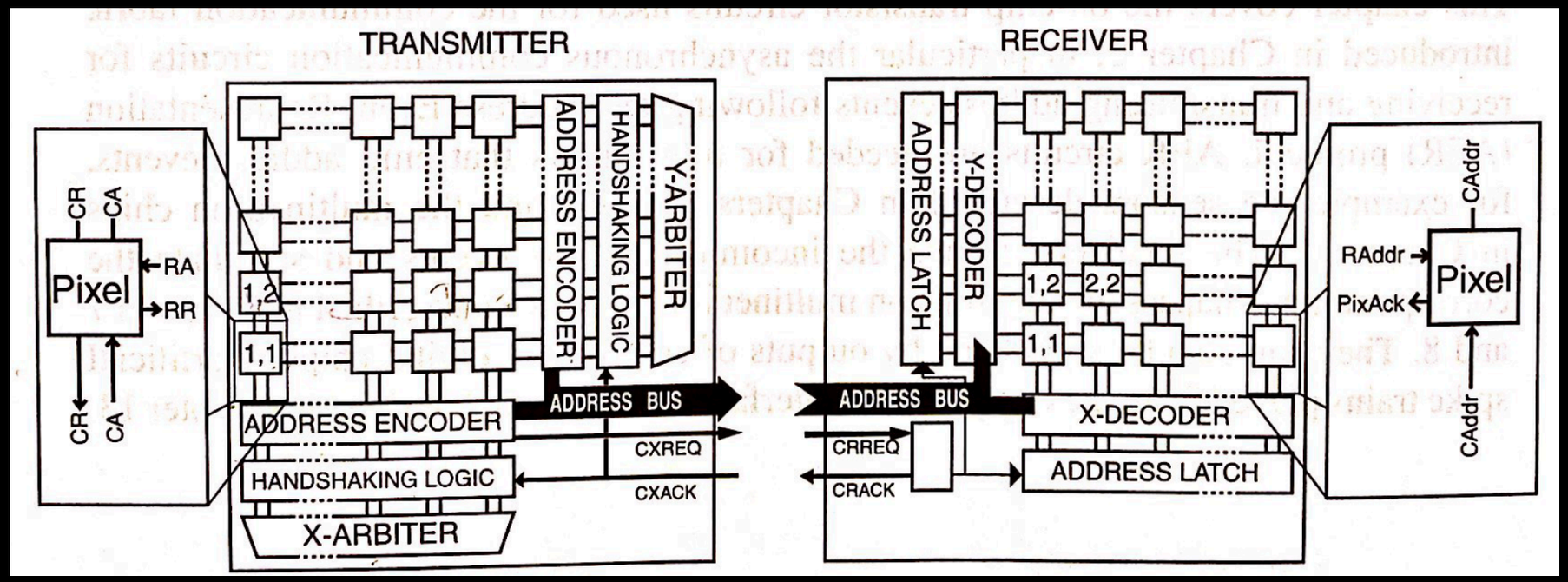
- Encodes the request signal into the data signals using two wires per bit of information: d.t for encoding logic 1 and d.f for encoding logic zero
- {x.f, x.t}= {1,0} (logic 0) and {x.f, x.t}= {0,1} (logic 1) represents “valid data”
- {x.f, x.t}= {0,0} represents no data
- Transition from one valid code-word to another one is not allowed (look at the FSM)
- 4 phase handshaking:
  - Sender issues a valid codeword
  - Receiver absorbs the valid codeword and sets acknowledge high
  - Sender replies by an empty codeword
  - The receiver takes the acknowledge low

Extending to bit-parallel channels



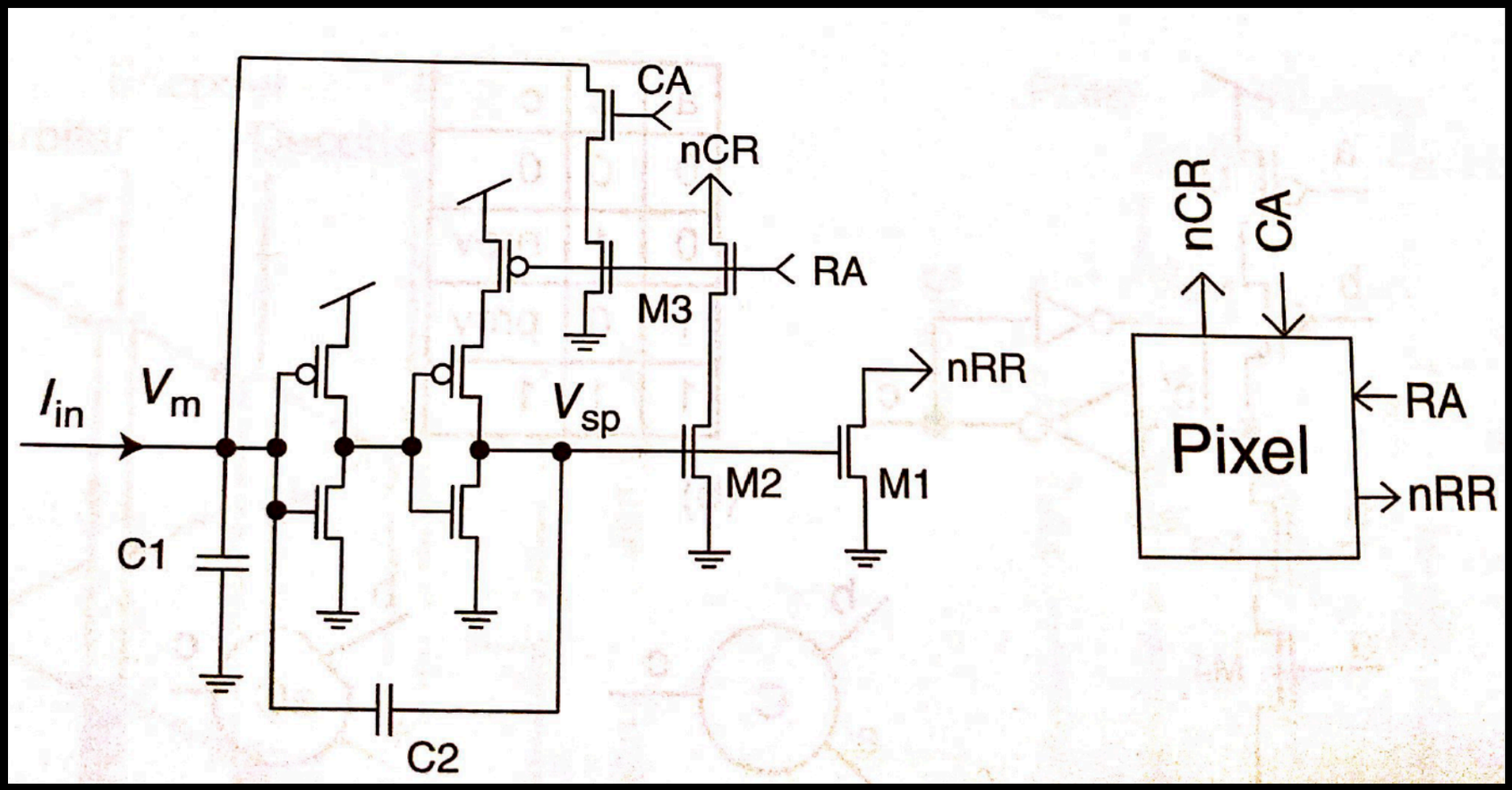
- N-bit data channel is formed by concatenating N wire pairs
- A receiver detects when all bits are valid (sets ack high)
- And detects when all bits are empty (sets ack low)

On-Chip AER Circuits



- Pixel fires: Activates its row request RR to the Y-arbiter
- Any active pixel in the same row will also activate the same RR
- Y-arbiter only acknowledges only one of these request by setting one of RAs to high
- The pixels with an active RA then sends a request in the column direction by activating the corresponding CR line to the X-arbiter.
- X,Y arbiters communicate with X,Y encoders which code the selected row and column and start a communication cycle with the handshaking block handling the off-chip communication through CXREQ and CXACK
- On receiving both RA and CA lines, the pixel withdraws its request from both arbiters

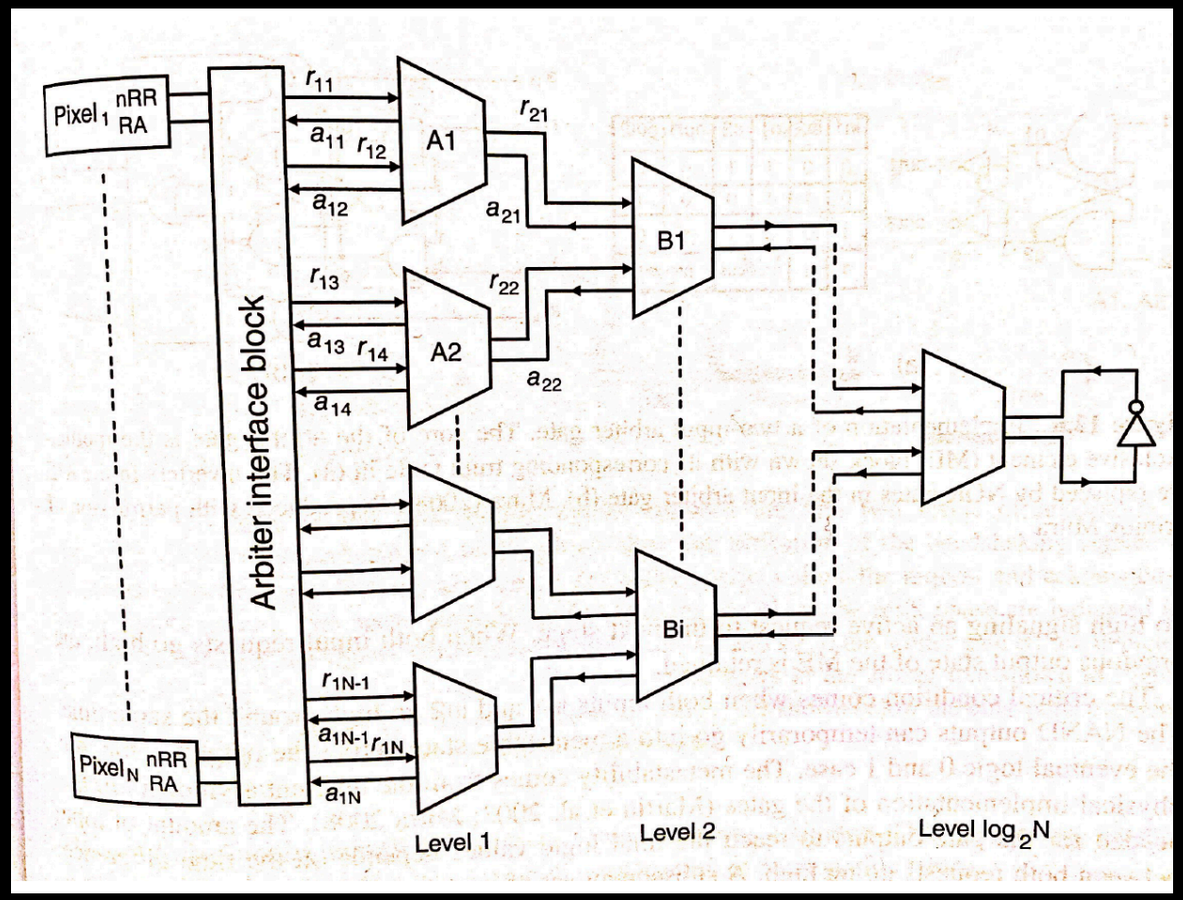
Example AER handshaking neuron circuit

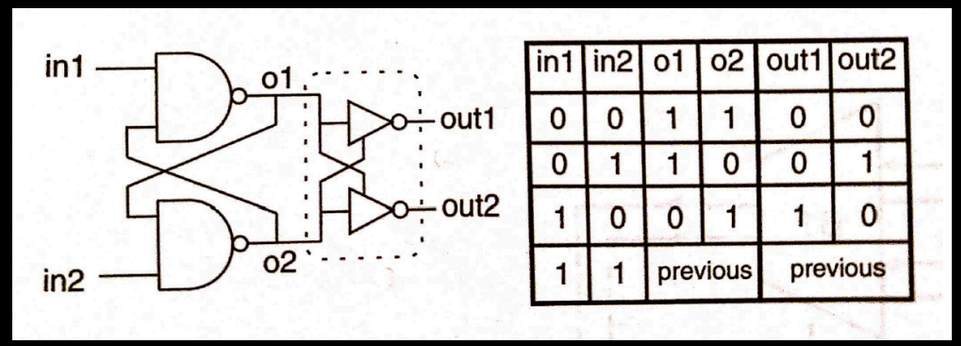


- When Vm passes the threshold, Vsp drives M1 to pull down on nRR*.
- When RA is activated from the row arbiter, M2 and M3 activate nCR
- When CA line is activated by the column arbiter, Vm is pulled low, resetting neuron output Vsp
- This stops pulling down nRR

* n means negate, nRR means Row Request is active low

Arbitration

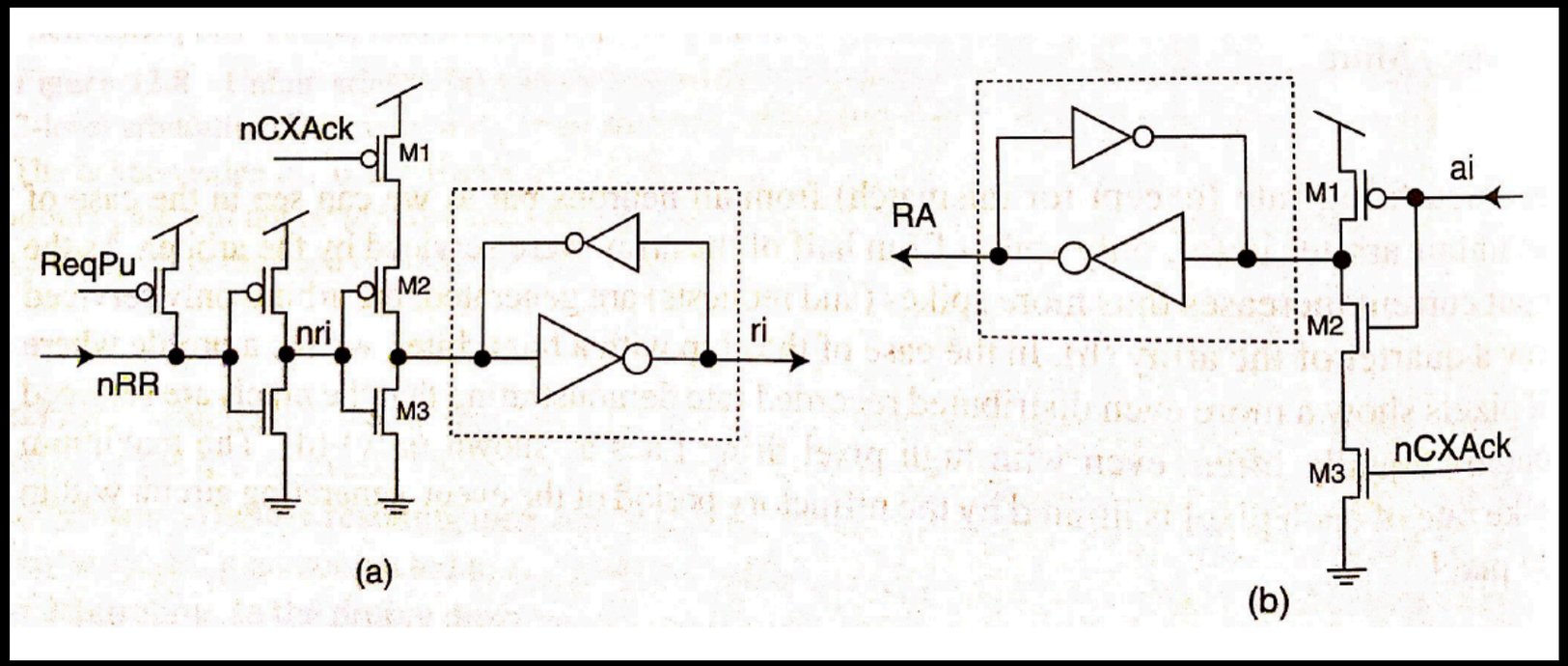




in1	in2	o1	o2	out1	out2
0	0	1	1	0	0
0	1	1	0	0	1
1	0	0	1	1	0
1	1	previous		previous	

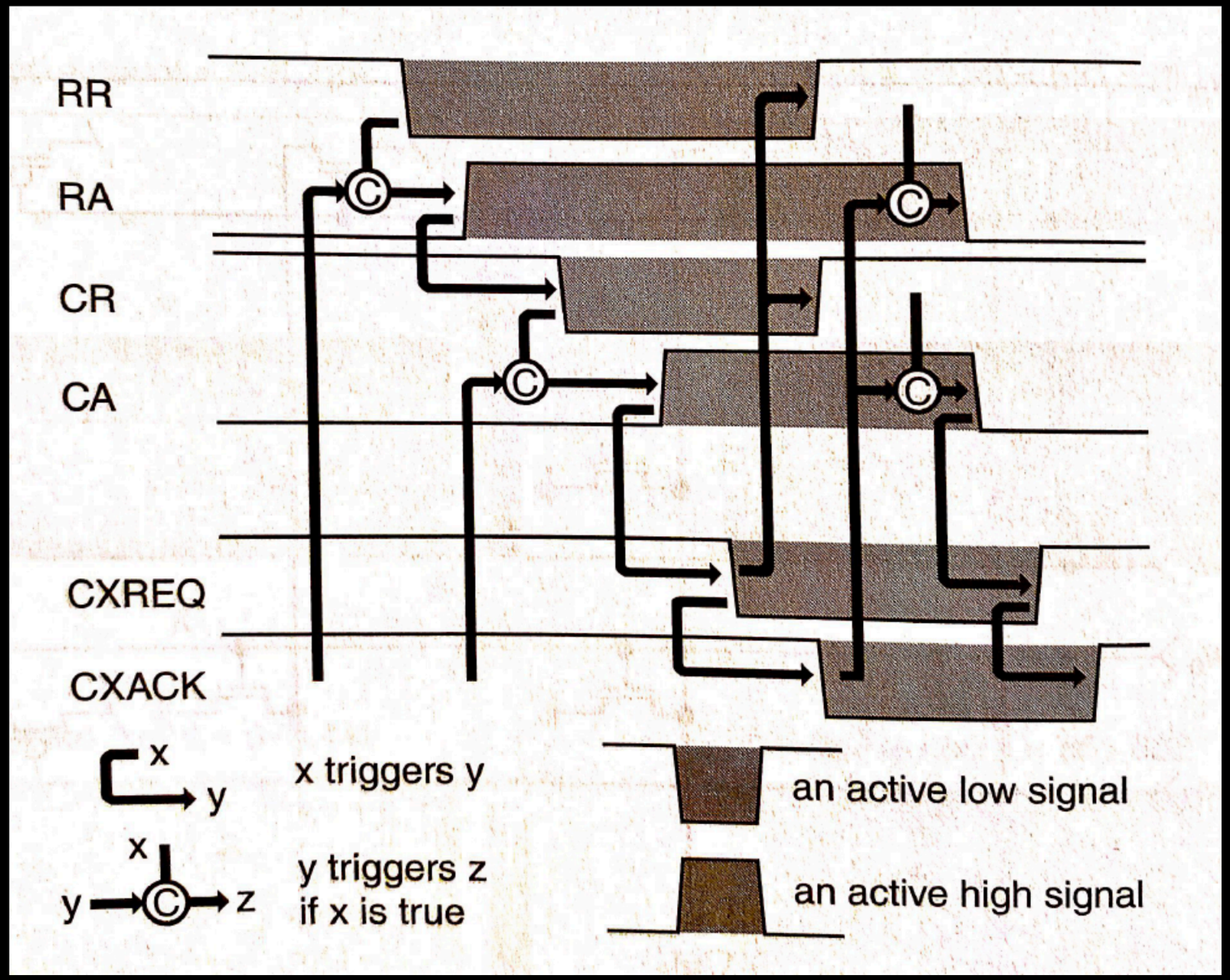
- The arbiter consists of a tree-like structure of N-1 two-input arbiter cells arranged in log2(N) levels
- Each two-input arbiter gate outputs a Req to the next level if any of its inputs are active
- The final selected output Req at the top of the tree goes to an inverter and becomes Ack signal propagating downward through various gates until corresponding RA is activated

Arbiter Interface Circuit

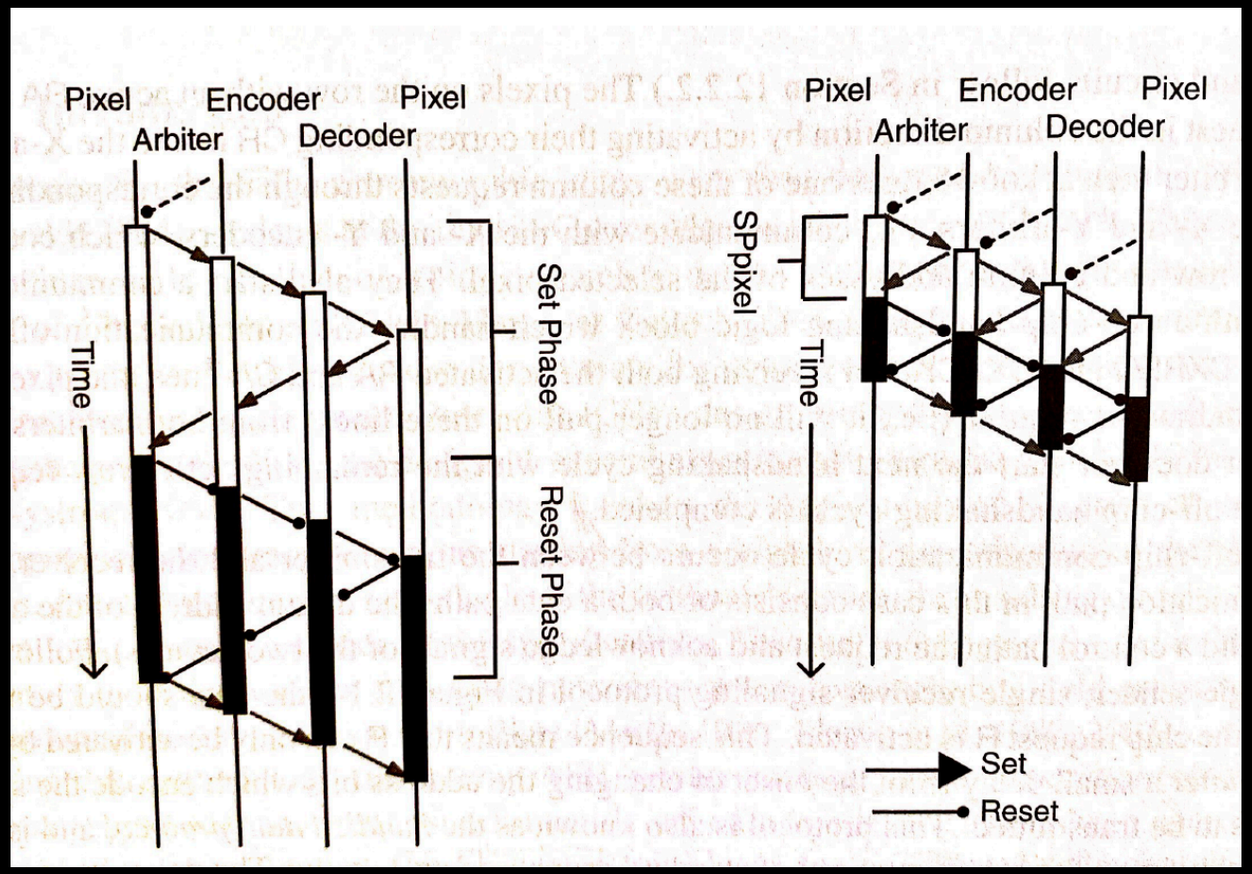


- An active low nRR signal generates a request to the arbiter (ri)
- pFET driven by ReqPu returns nRR high when no request
- The req is not taken away until nCXAck become inactive
- The circuit in b activates RA only if the arbiter returns an active acknowledge and nCXAck is inactive
- This circuit ensures that a new row can be selected by the arbiter only after the address of the sending pixel has been transmitted and recorded off-chip

Combined operation



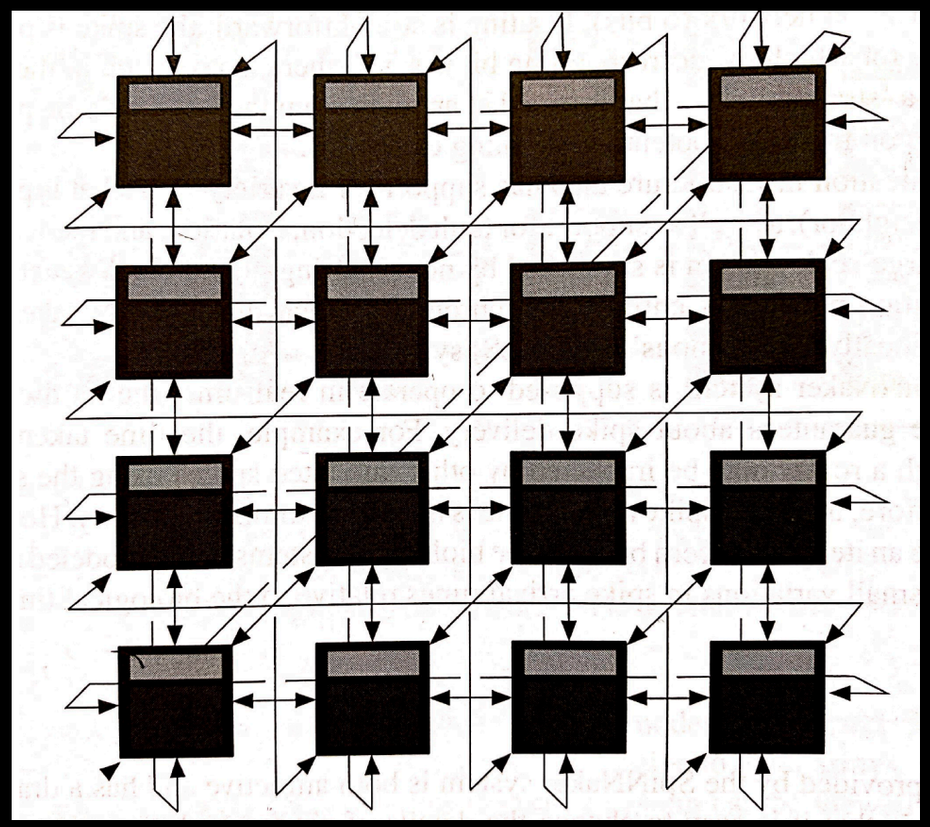
Control signal flow



- a: waiting for the set signal to propagate to the final destination before starting the reset phase.
- b: pipelining reduces the handshaking time by allowing the signals to propagate forward in the set phase without waiting for the reset phase of the previous stage

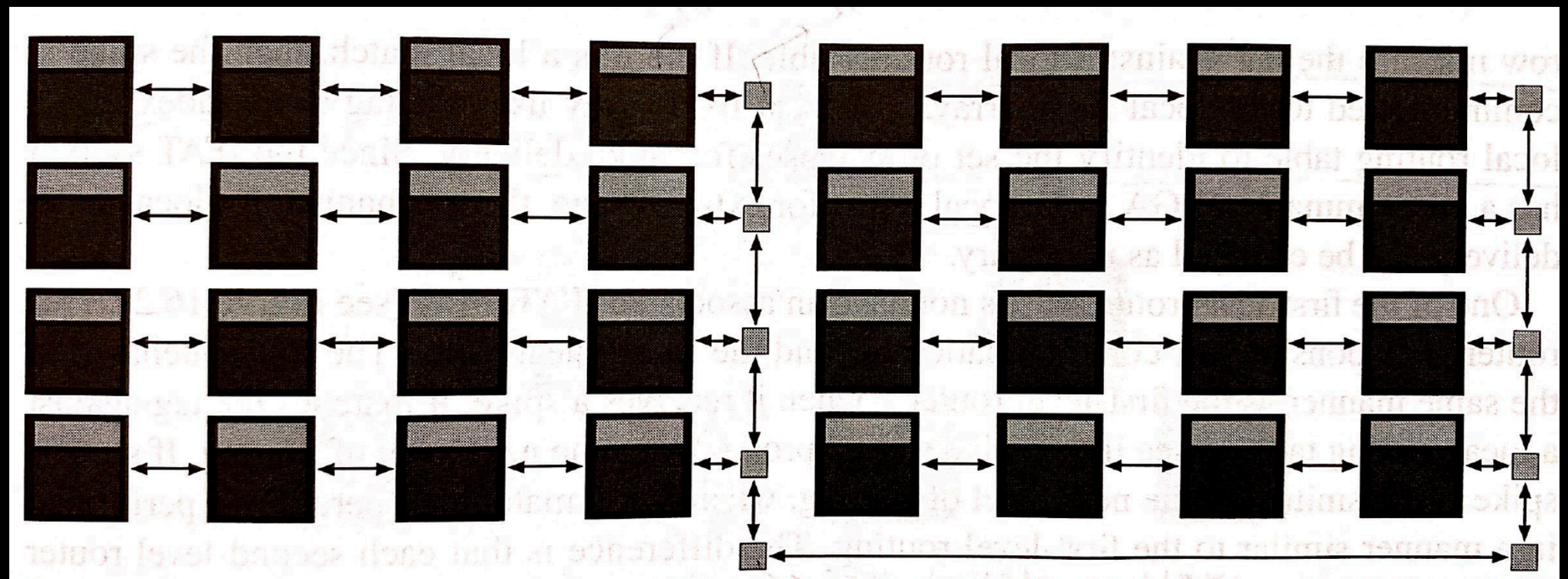
Towards Large-Scale Neuromorphic Systems

SpiNNaker (mesh-based)



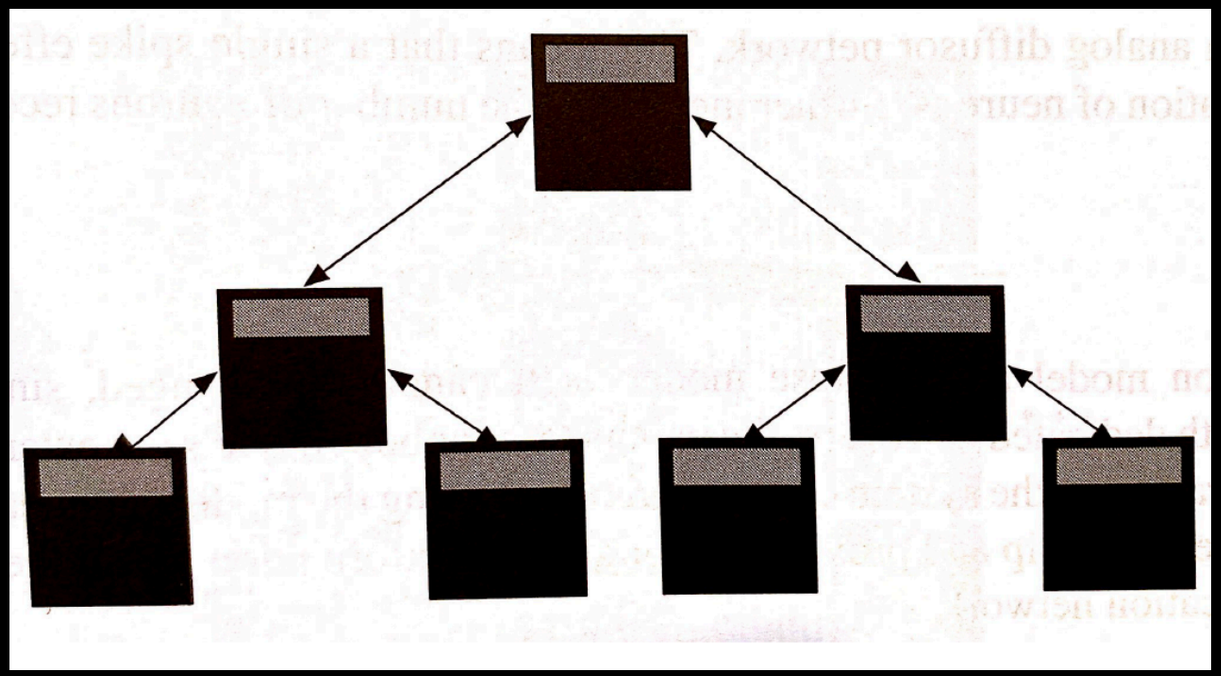
- 18 Arm cores which can be programmed to neuronal and synaptic computation
- Each chip can communicate with six nearest neighbors.
- Each chip participates in a horizontal, vertical and diagonal ring
- When a neuron spikes, it is given a source routing key and is delivered to the communication fabric for delivery to destination neurons
- No central routing table, each chip has its own routers to make local decisions
- When a spike arrives on one of the 6 incoming ports, the source key is matched against the entries in a TCAM
- If there is match, the destination address is retrieved.

HiAER (multi-level router)



- Integrate and Fire Array Transceiver (IFAT) nodes and routing chips
- Each node has a communication local routing resource
- If a spike is local, it is looped back internally
- If non-local,
  - it is transmitted to the router immediately adjacent to the IFAT node (first-level router)
  - It broadcasts (via nearest neighbor communication) the spike to the local row
  - Each individual row matches the tag against a local routing table

NeuroGrid (tree-based)



- The generated spike travels up the tree to an intermediate node and then traverse down to the destination core
- Spikes are source-routed. Each spike contains the path taken by the network