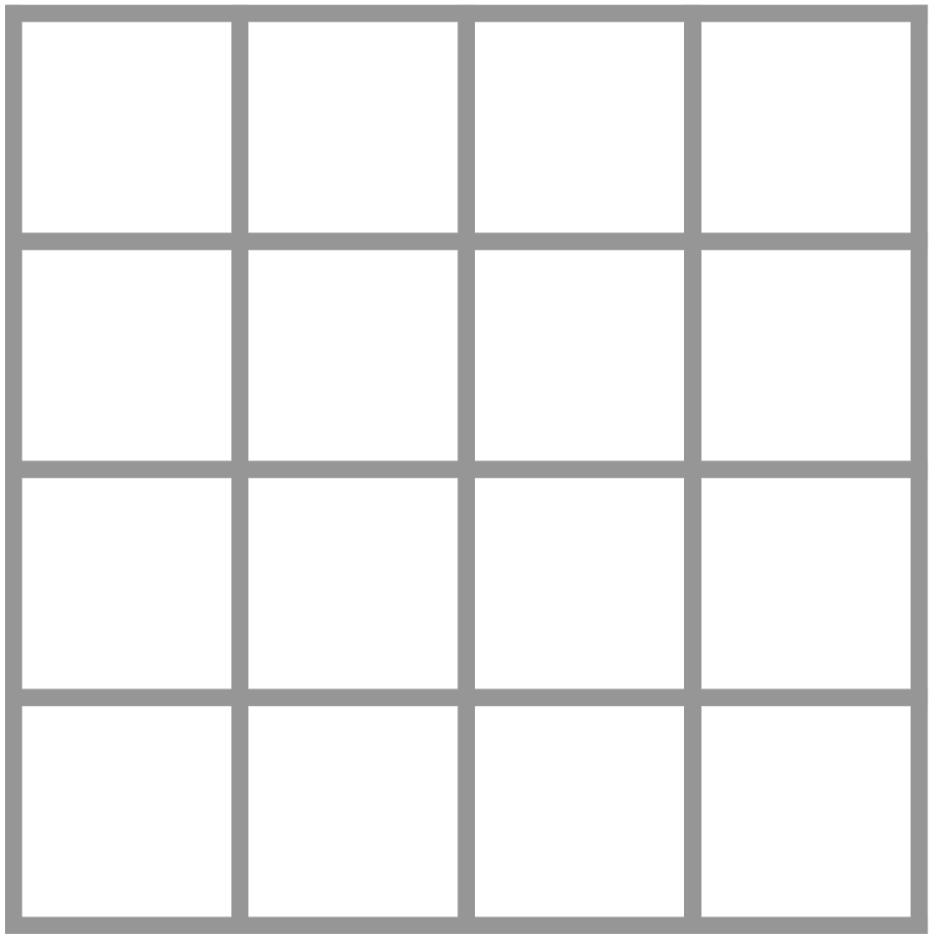
Routing Events in Two-Dimensional Arrays with a Tree

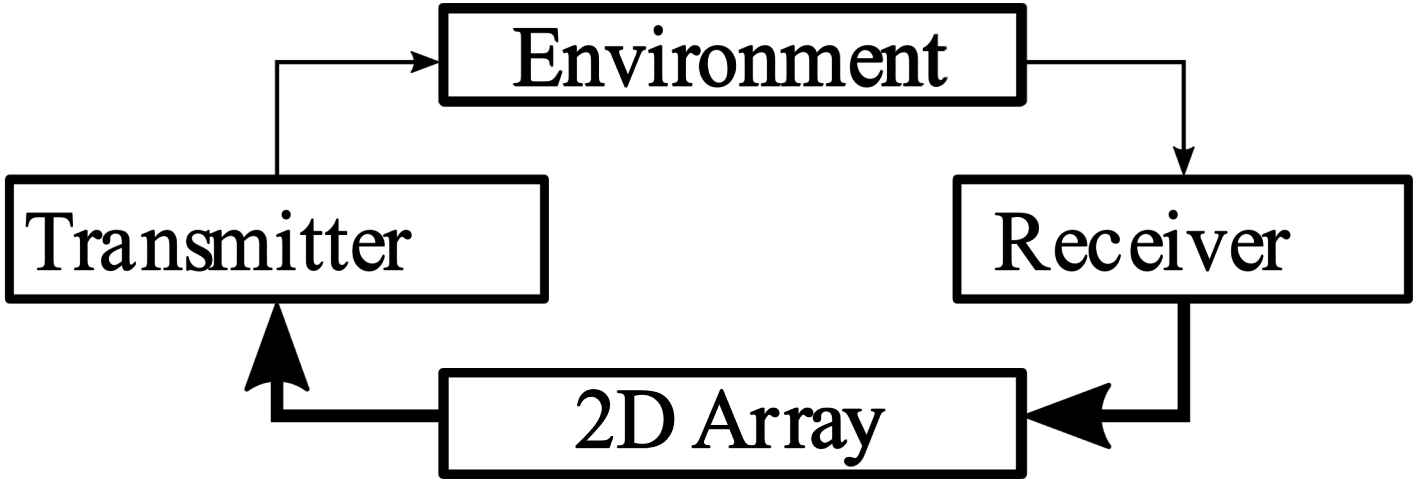
Sam Fok and Kwabena Boahen*
Femtosense Inc. and Stanford University

CVPR 2021 Workshop on Event-Based Vision

*presenter

Problem Statement

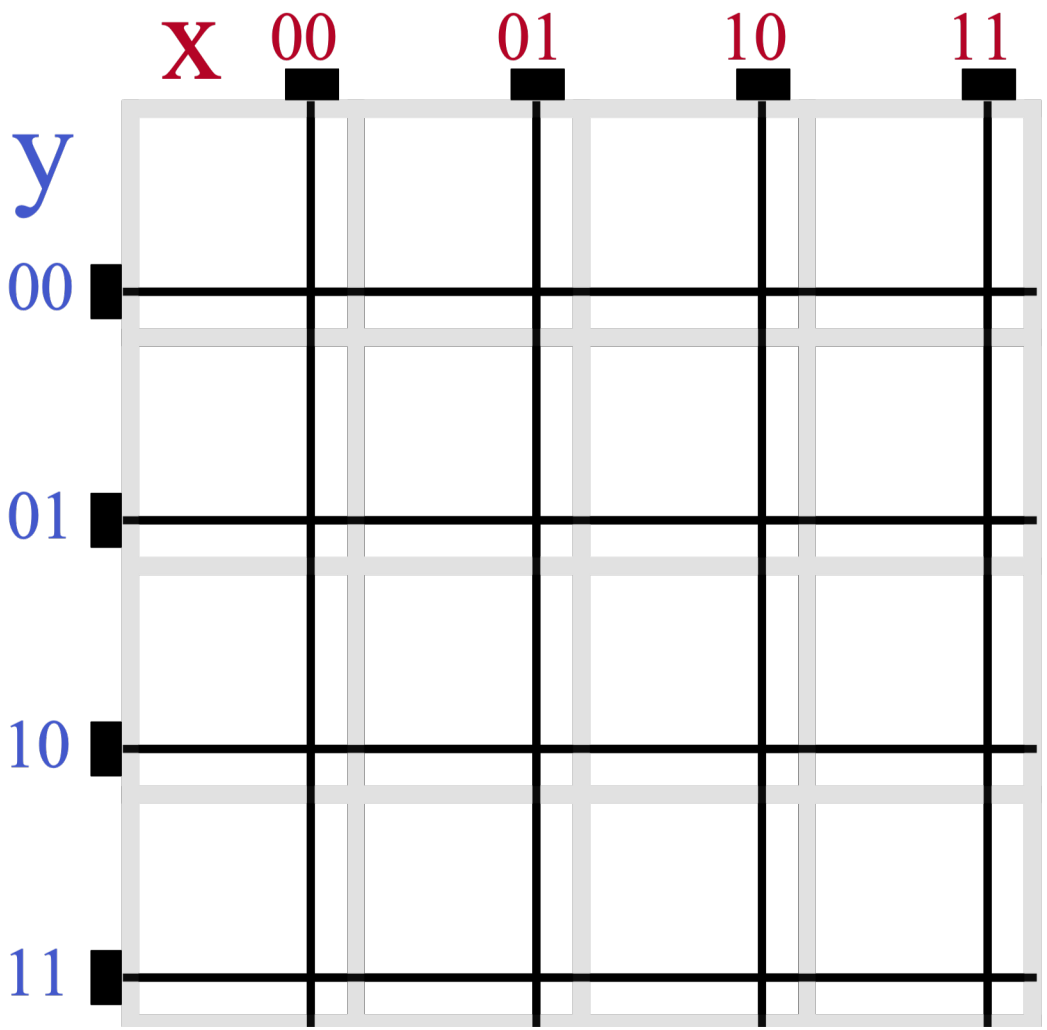




Transmit and receive up to 1,000 eps/pixel

- For sparsely active pixels
  - Minimize latency
  - Maximize throughput

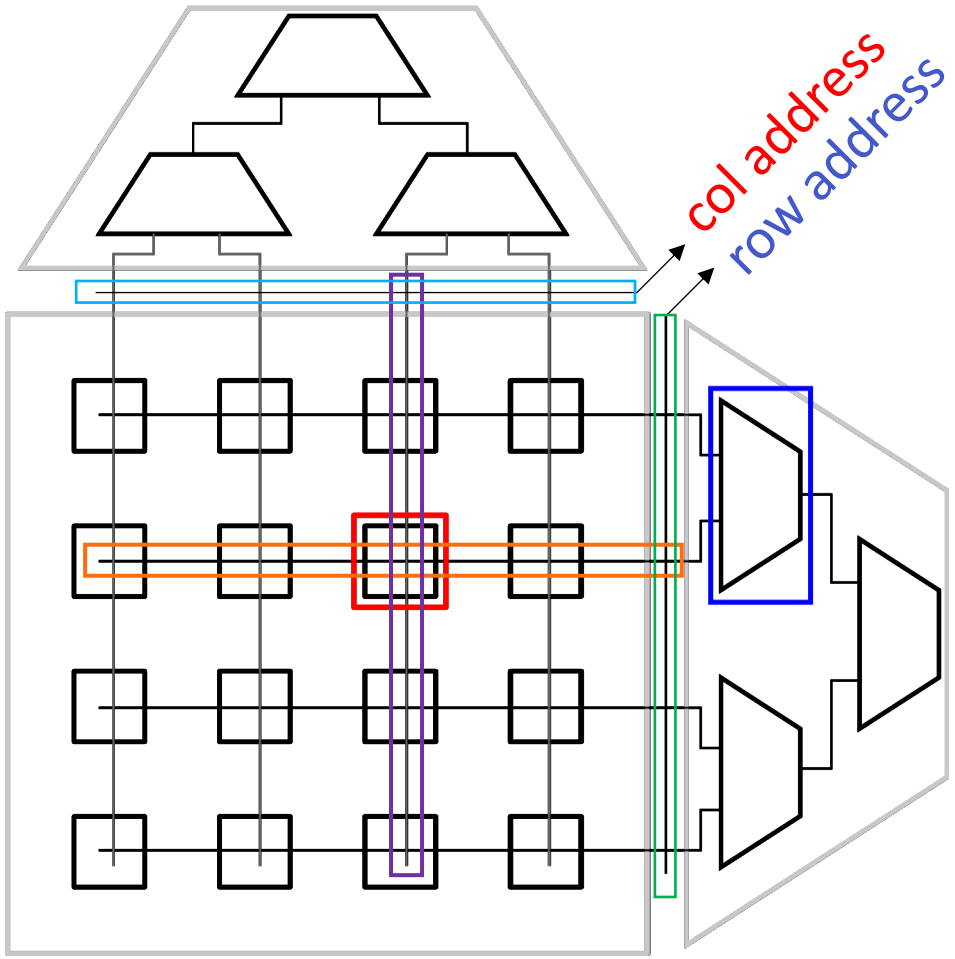
Existing Solution



Good: O(√N) circuitry
- for N clients

Not as good:
- race conditions
- timing assumptions
- poor scaling

Event-Driven Transmitter



Pixel

Arbiter Cell

Wired-Or (row)

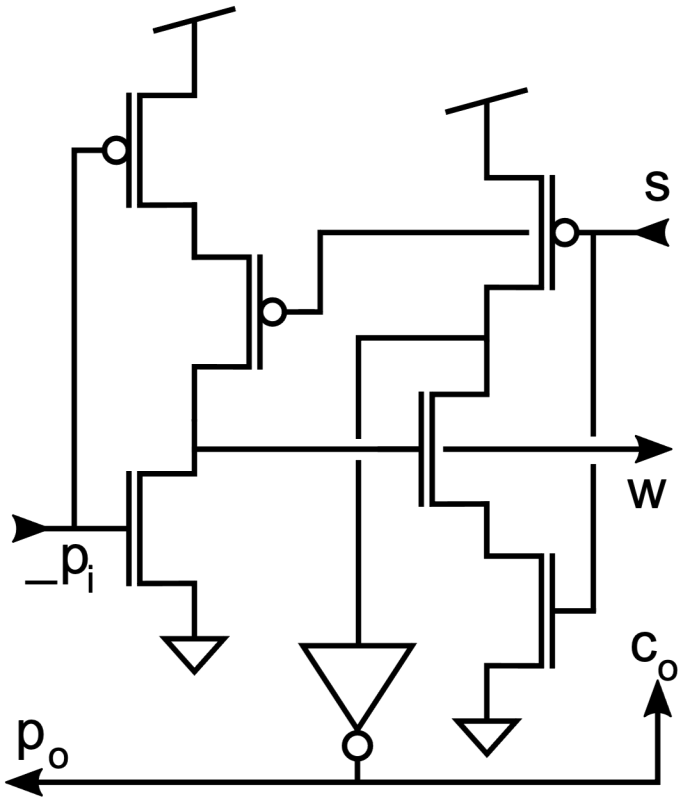
Encoder (row)

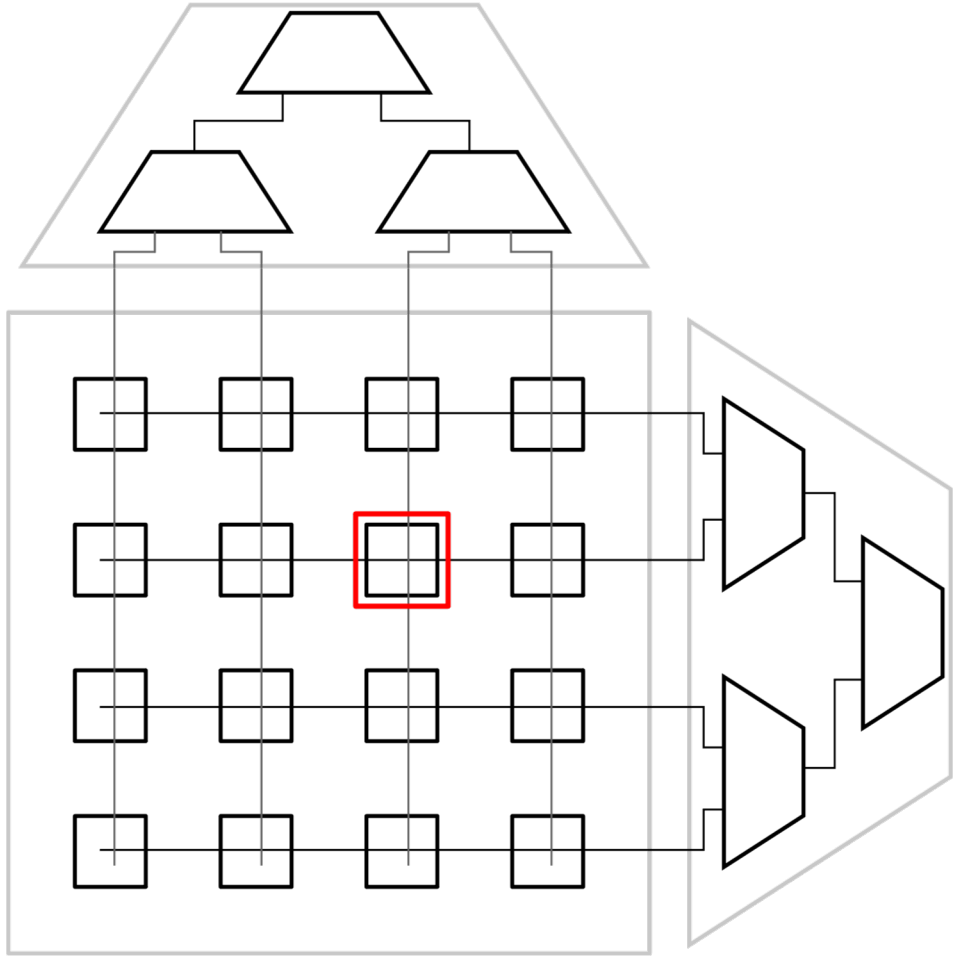
Wired-Or (col)

Encoder (col)

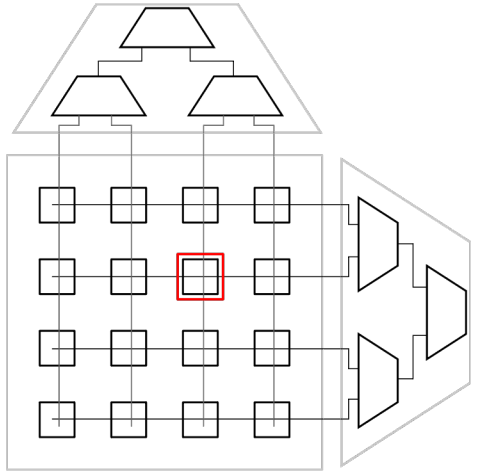
Sivilotti 1991 & Mahowald 1992

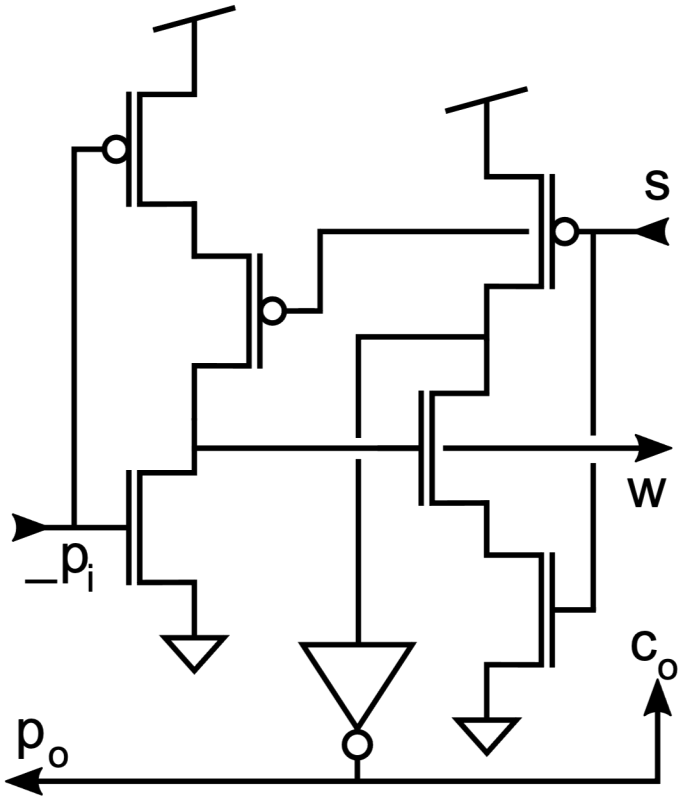
Pixel Circuitry





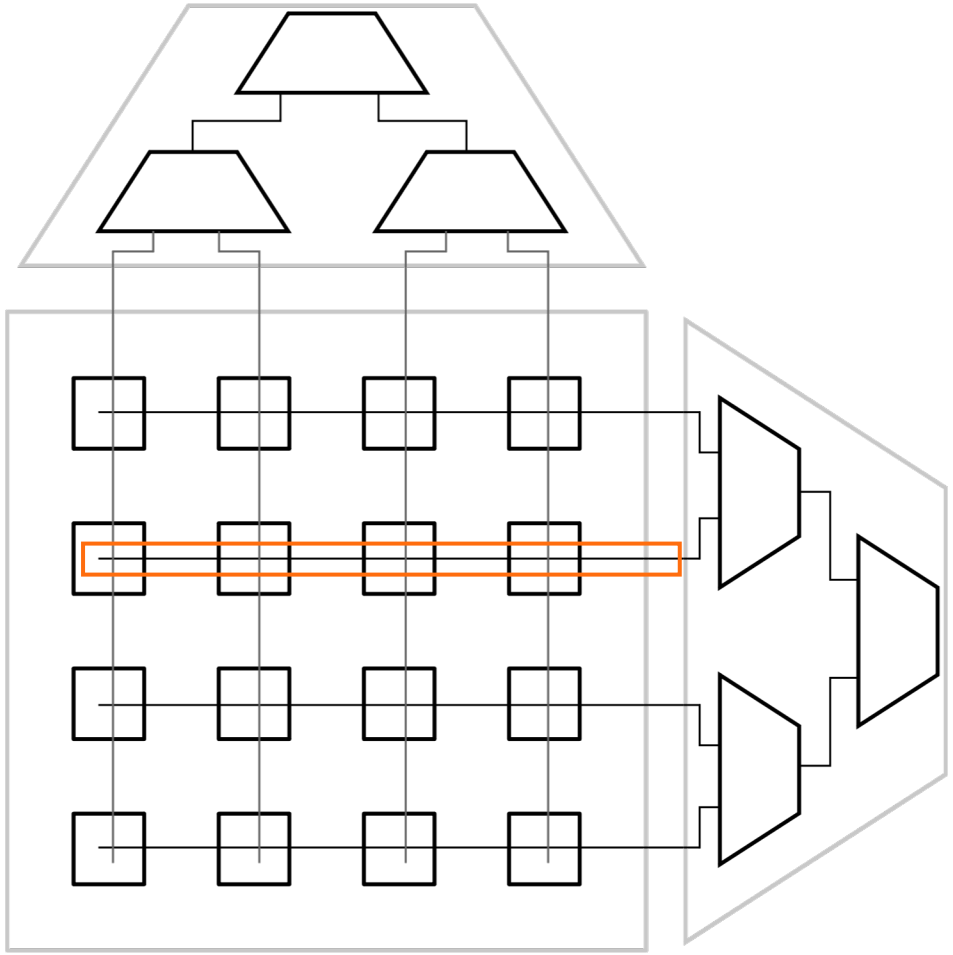
Pixel Interface





$\_p_i$: pixel request
w: row request
s: row select
p_o: pixel reset
c_o: column data

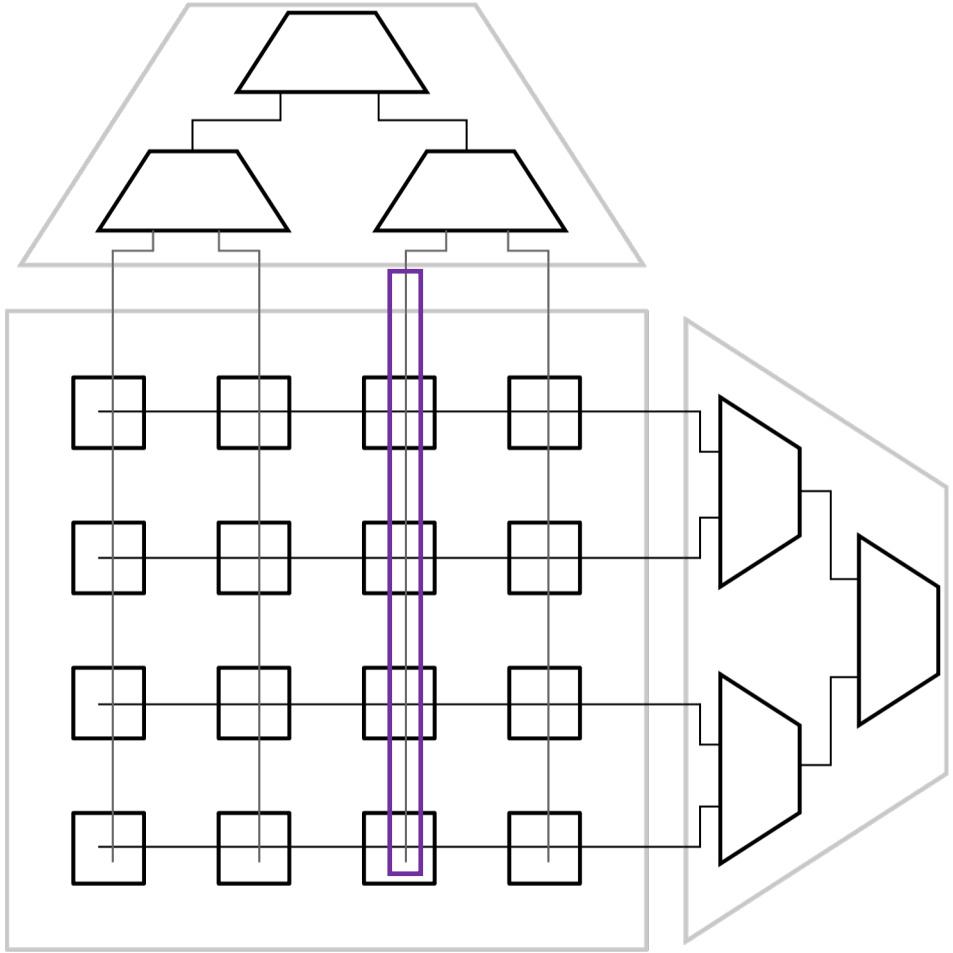
Row Wired-Or



Connects many to one

- row-select latches pixels' request
- to halt new requests in that row
- but these requests race select

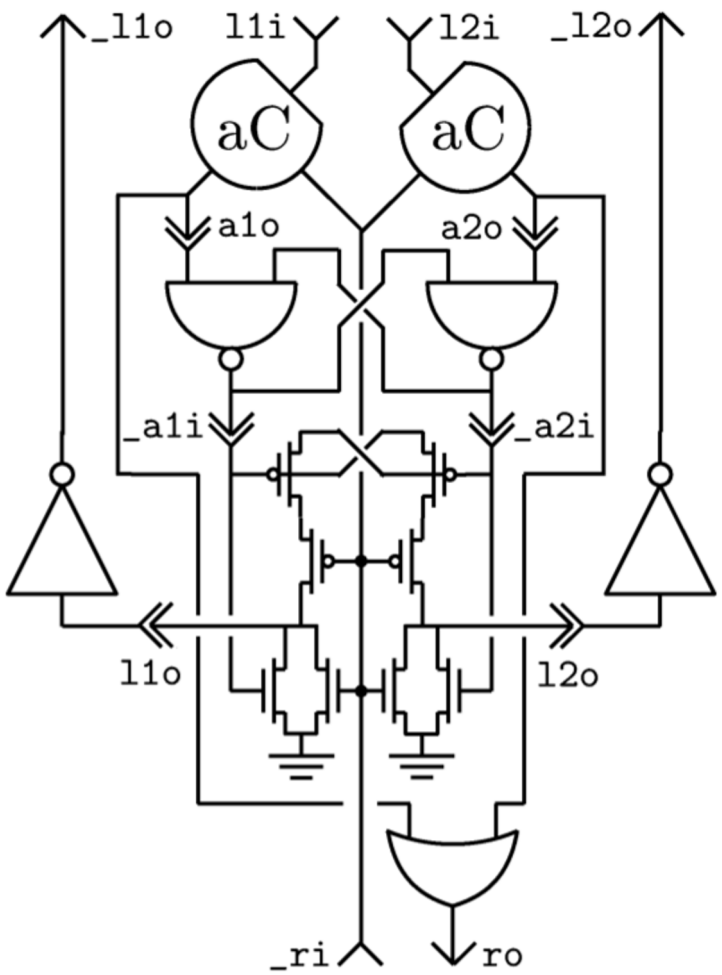
Column Wired-Or

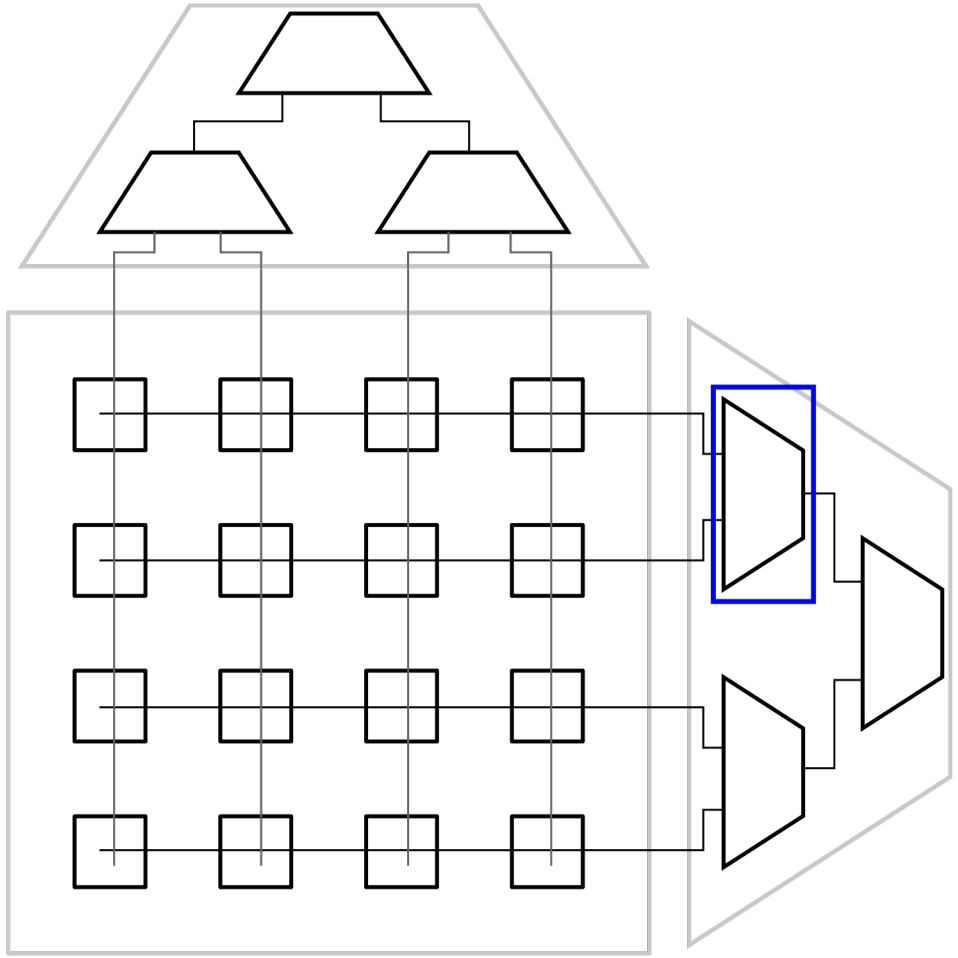


Multiple columns are active

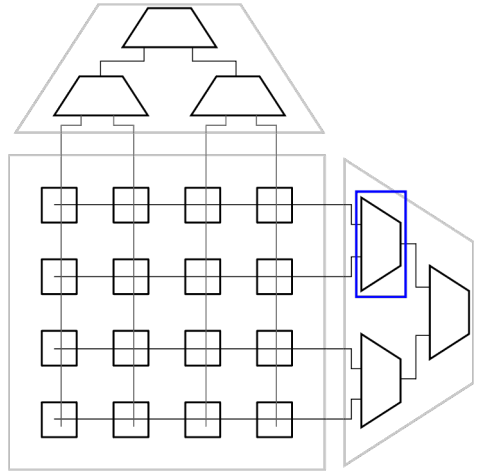
- their signals are latched
- slow signals don’t register
- so delays must match

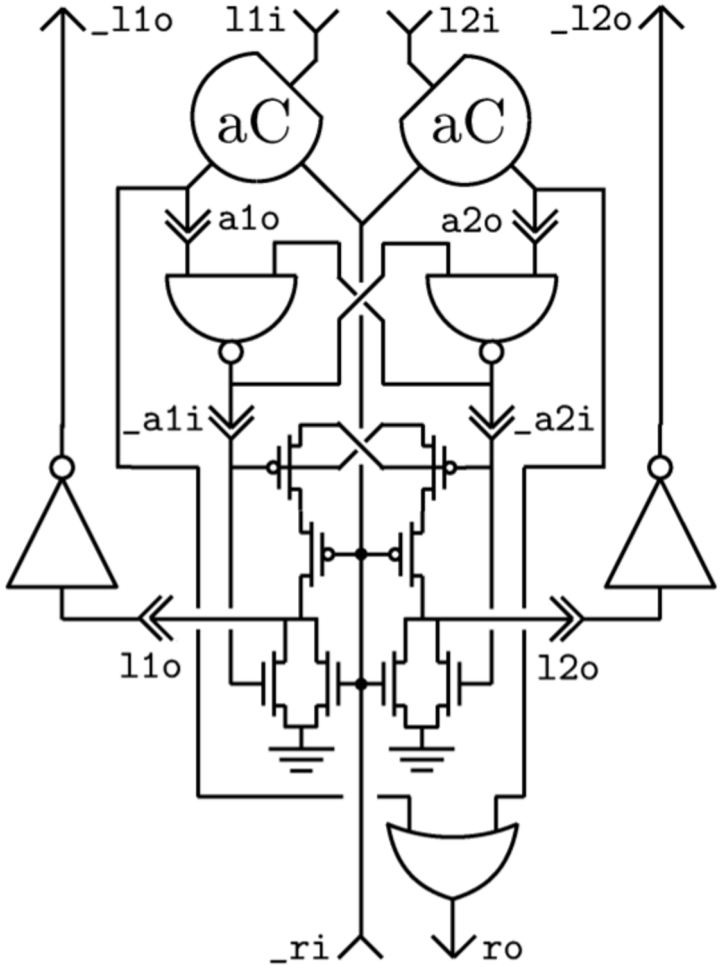
Fair Greedy Arbiter





Fair Greedy Arbiter

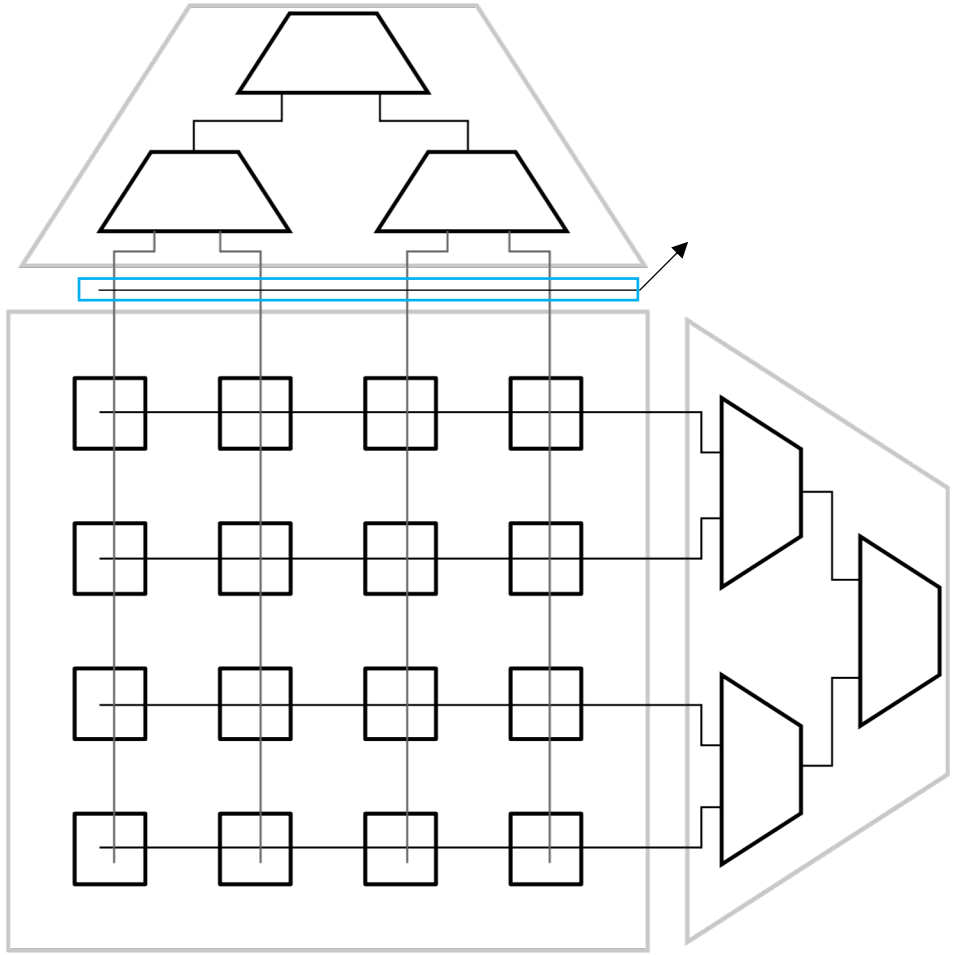




Services requests *breath-first*

- parent's grant (_ri) latches child's request (l1i,l2i) in aC-element
- this prevents child from re-requesting until all tree's leaves are visited
- but the request races the grant

Column Encoder

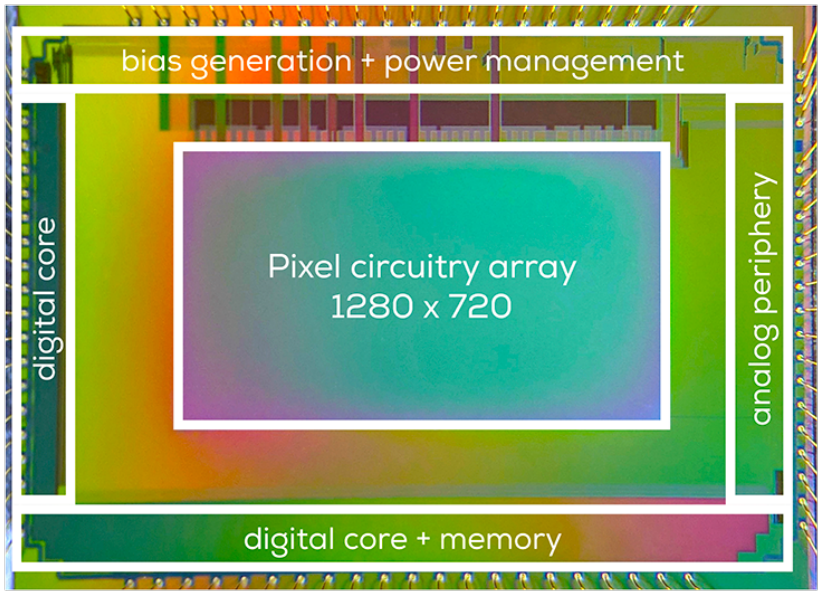


In critical path

- address encoding happens serially
- time ∝ capacitance ∝ columns
- as columns increase, throughput drops (< 50M eps)

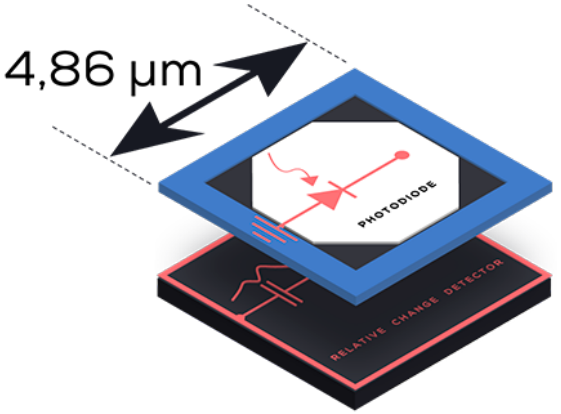
SoTA Event-Based Vision Sensor

Top micrograph



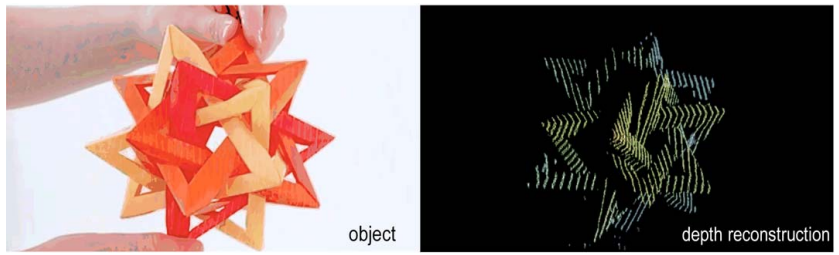
Bottom micrograph

90nm BSI CIS



40nm CMOS

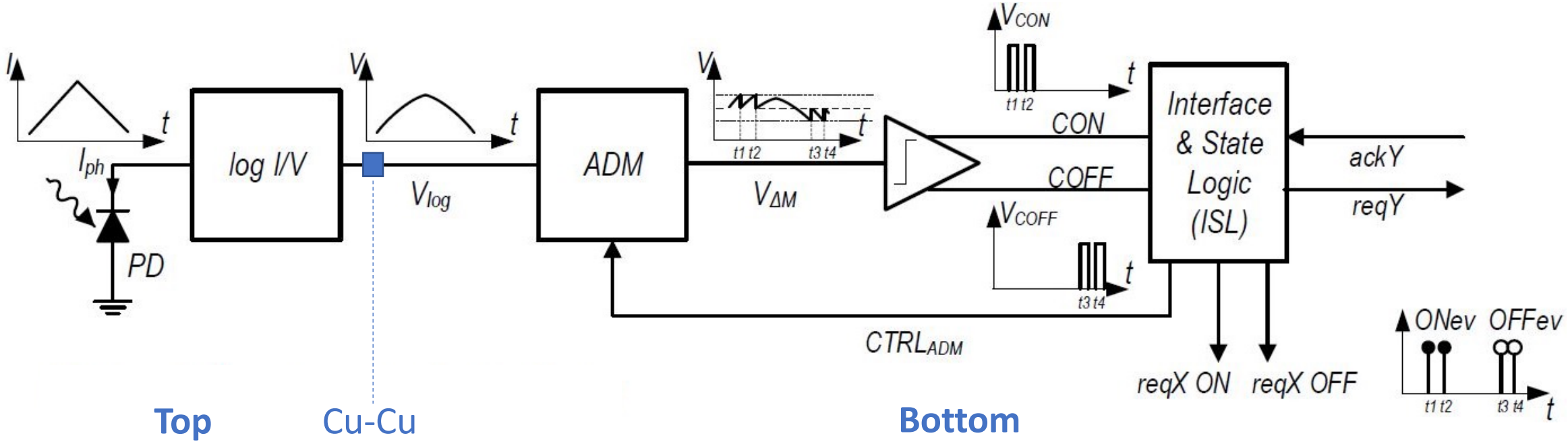
Senses depth with structured light



1 Geps at 1.6 bit/event

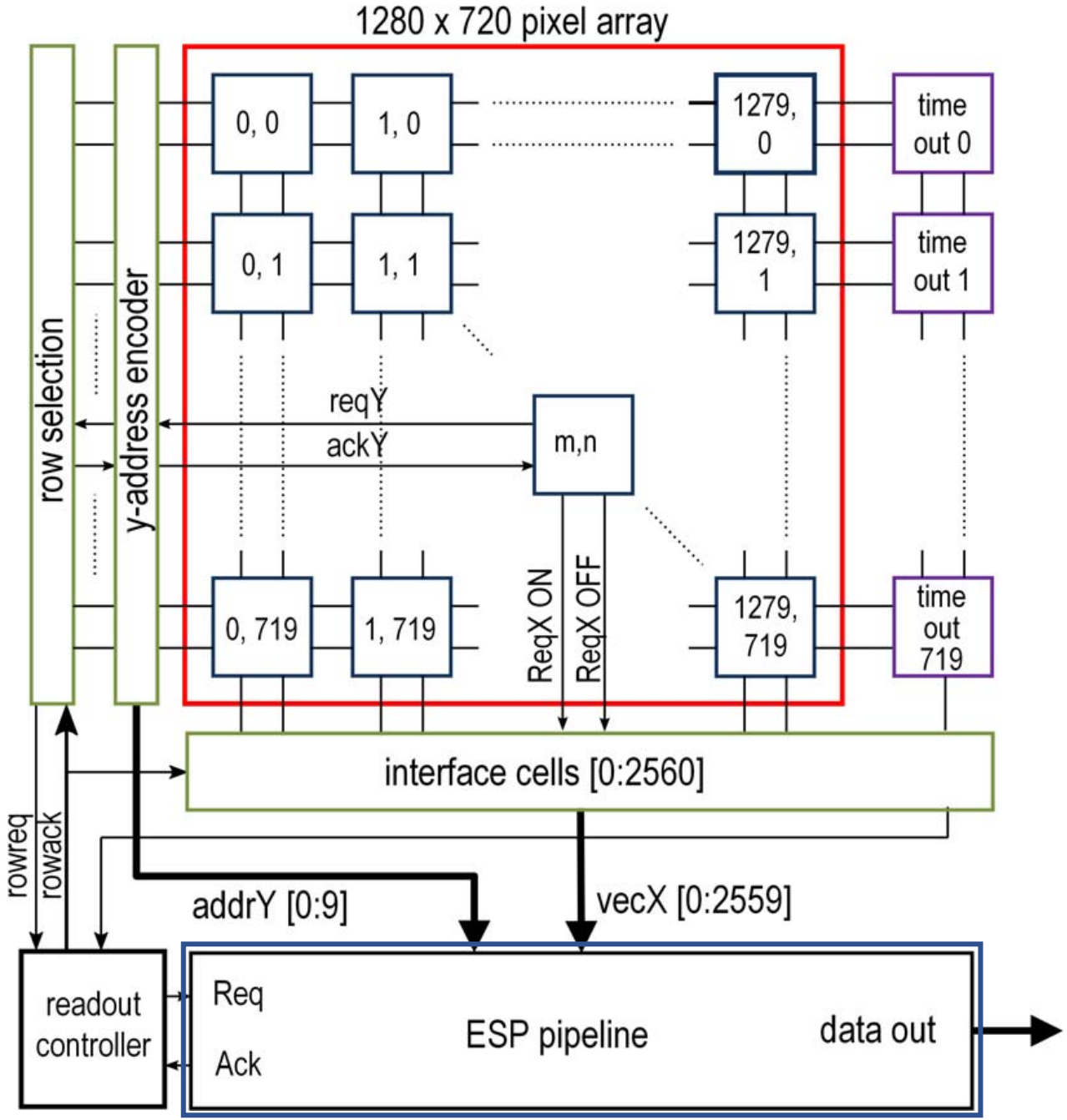
Finateu et al. 2020

Pixel Circuit



Light Sensor: Logarithmic photoreceptor

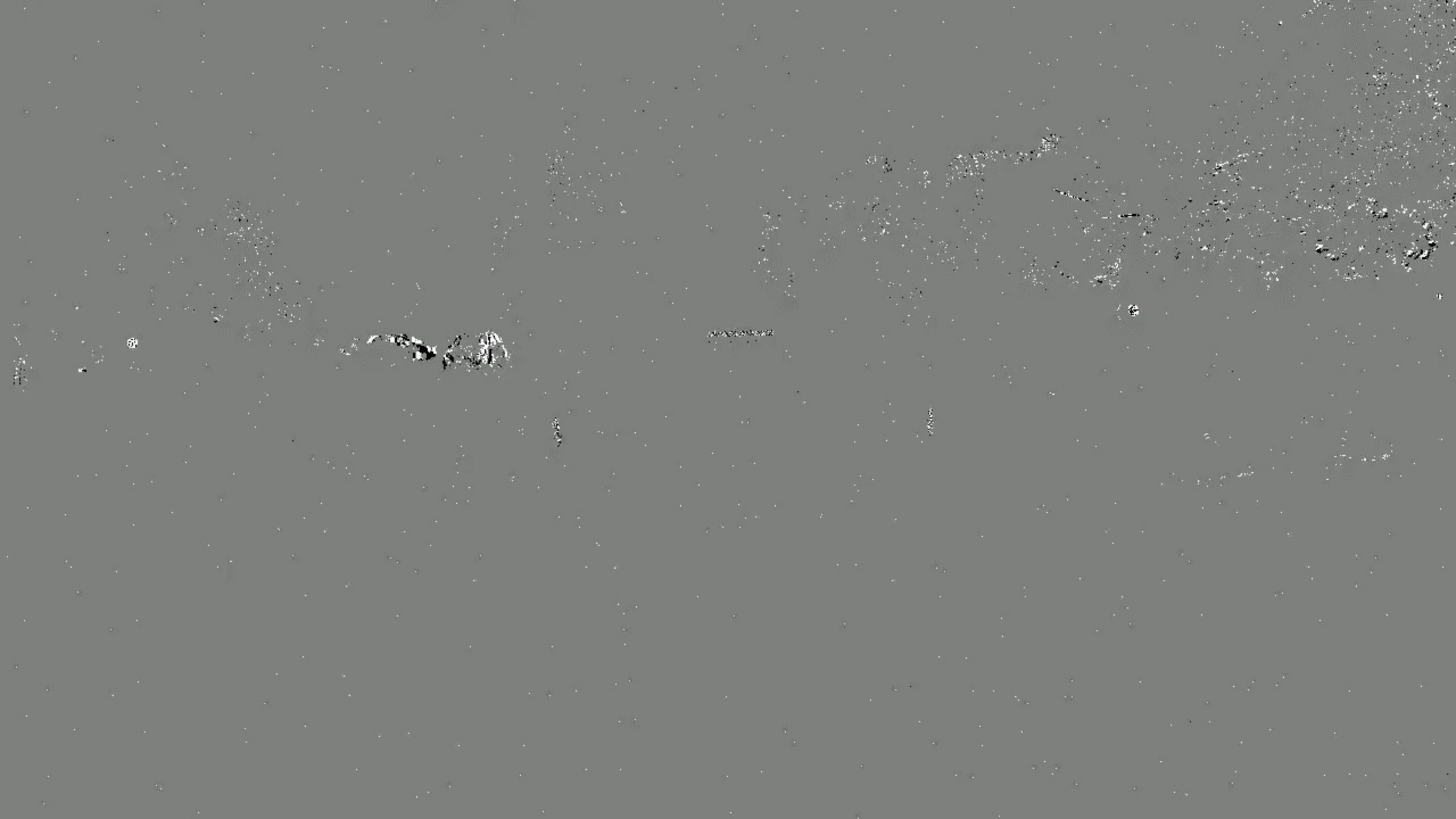
Change Detector: Adaptive delta modulation



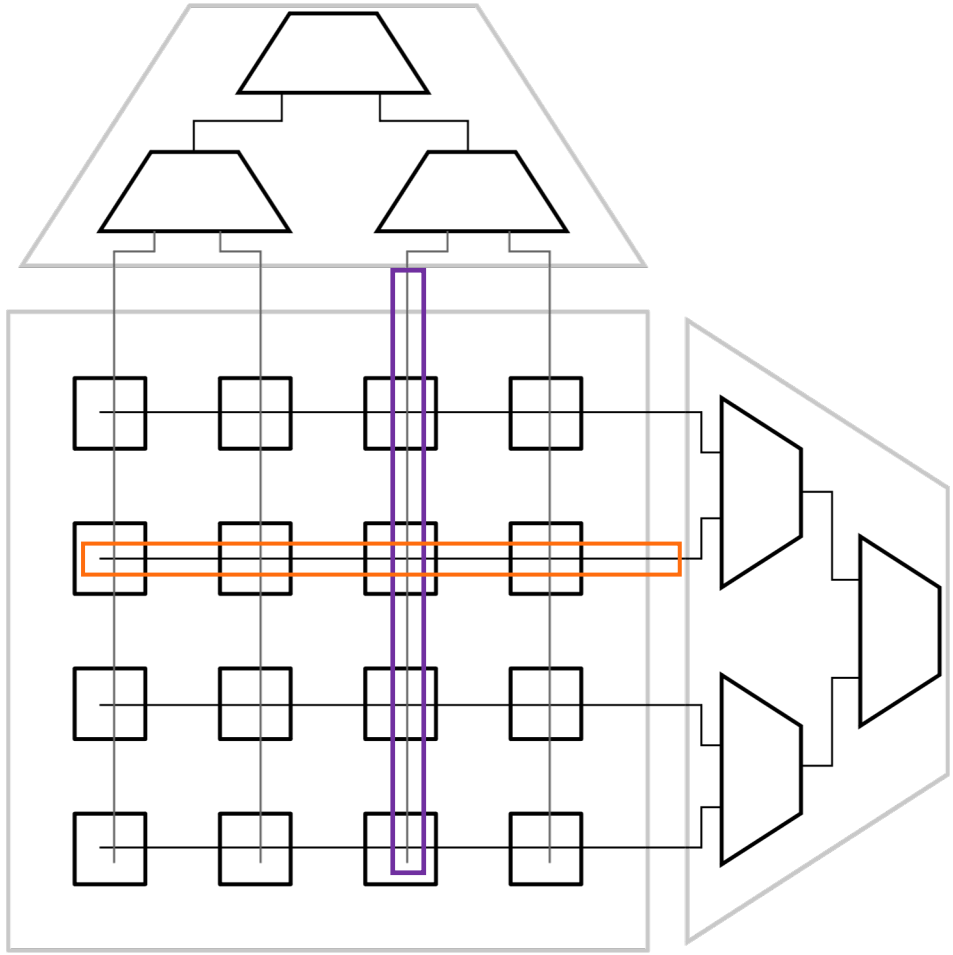
Event Signal Processor (ESP)

- replaces column-encoder
- timestamps row data immediately (1µs resolution)
- compresses this data spatio-temporally (vector format)
- to as little as 1.6b/event (on average, for heavy load)

Finateu et al. 2020



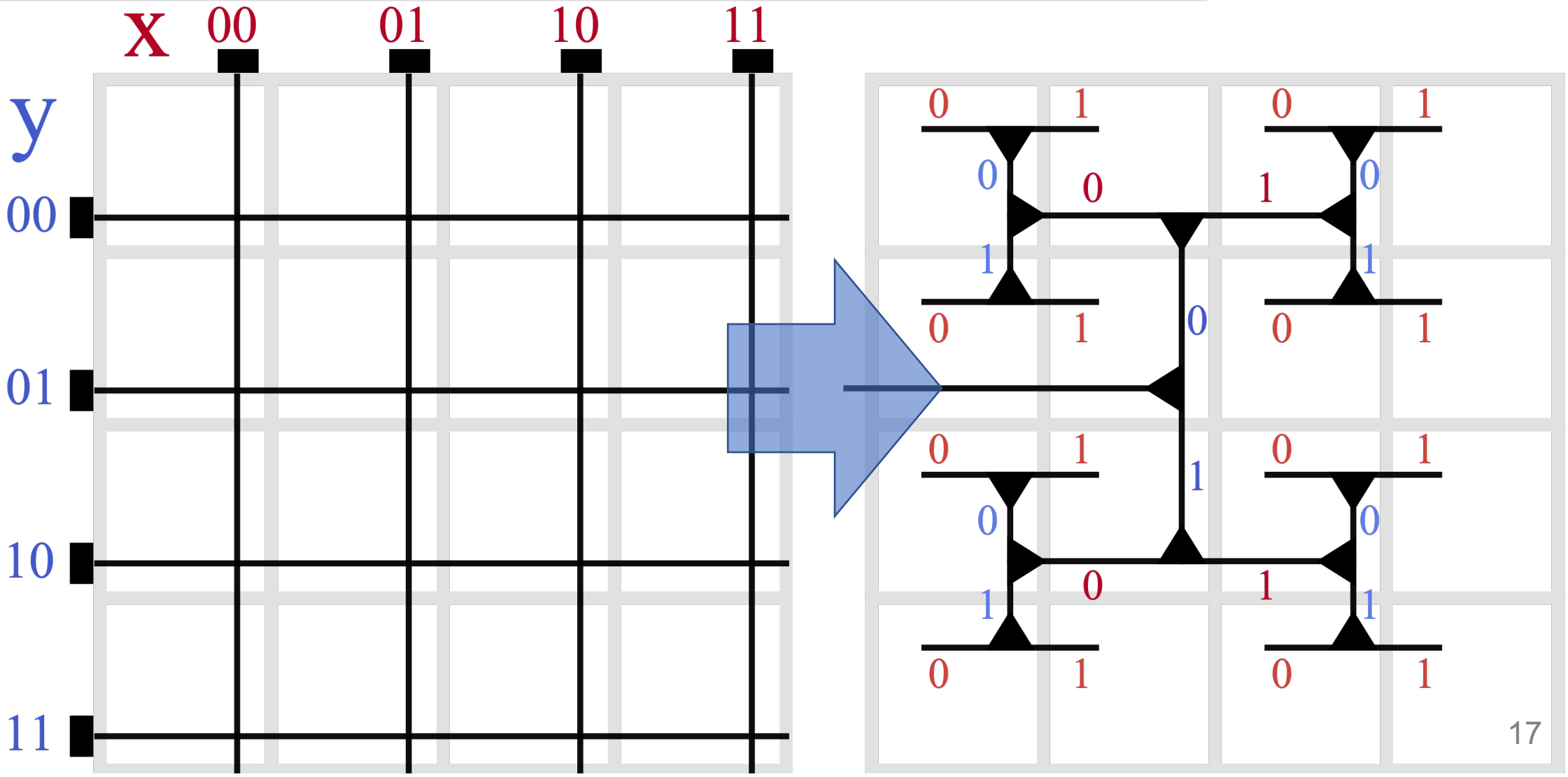
Wired-Or Scales Poorly



Row and column lines lengthen

- distributed capacitance increases
- voltage takes longer to equilibrate
- signal integrity degrades

Proposed Solution

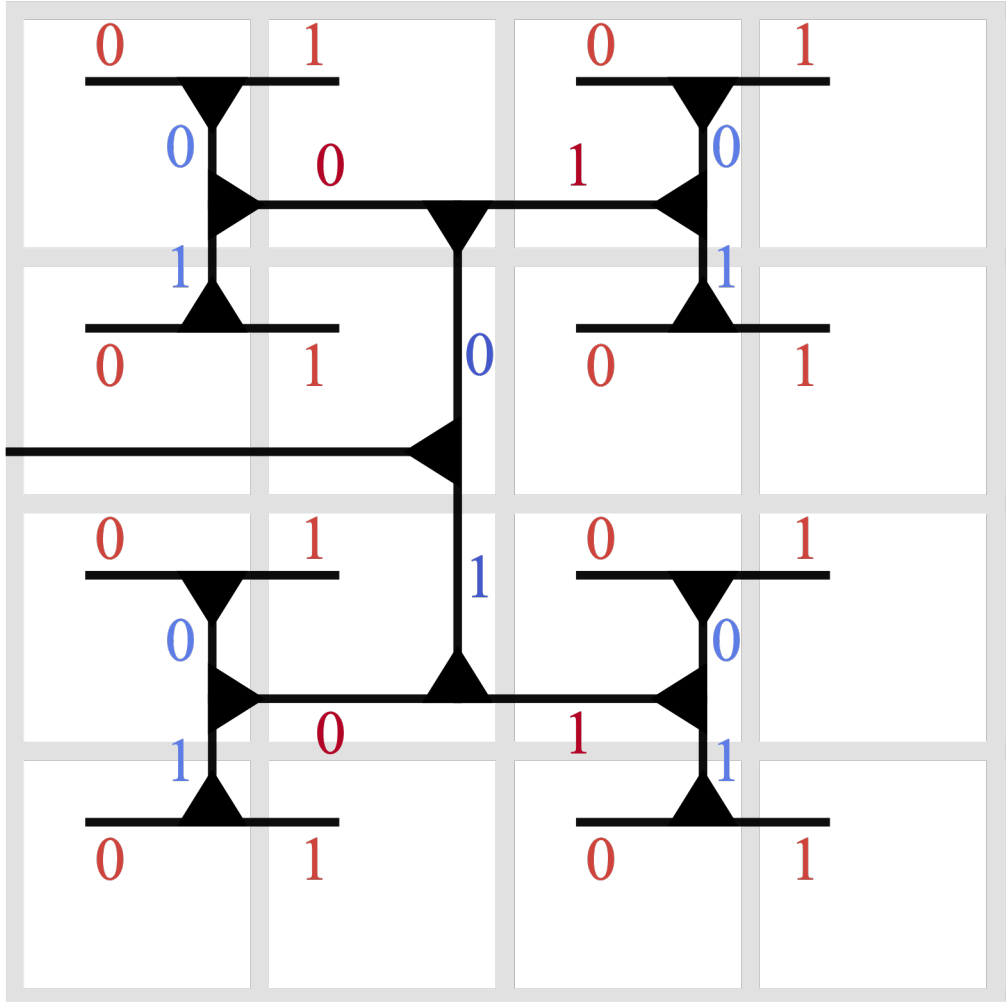


Places Transceiver Beneath Array

Pro: No wired-ors
- ~~race in row of pixels~~
- ~~matched delays in columns~~
- ~~poor scaling~~

Pro: Encodes address while arbitrating
- ~~encoder in critical path~~
- ~~child-request/parent-grant race~~

Con: O(N) cost
- Communicating address bits serially helps

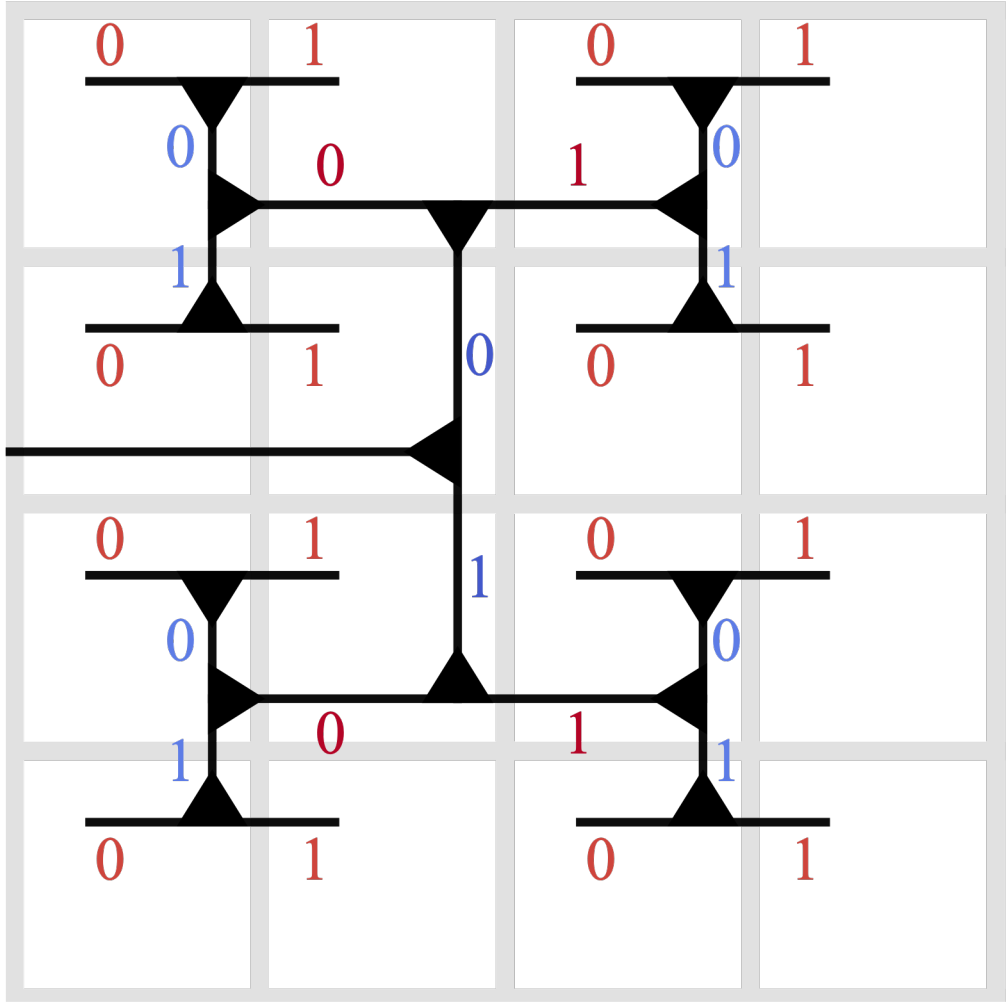


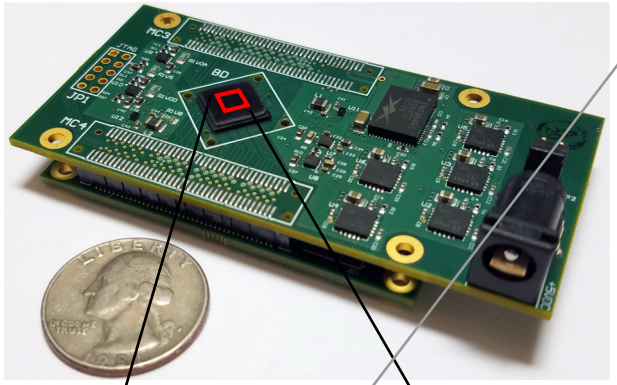
Bit-Serial Communication

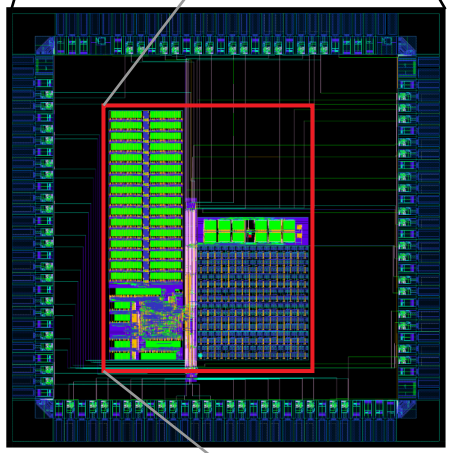
Pro: Arbitrarily large address-space
- no additional wires needed

Pro: Accommodates data as well
- just append additional bits

Con: Less throughput than bit-parallel
- Saving wiring cuts bandwidth



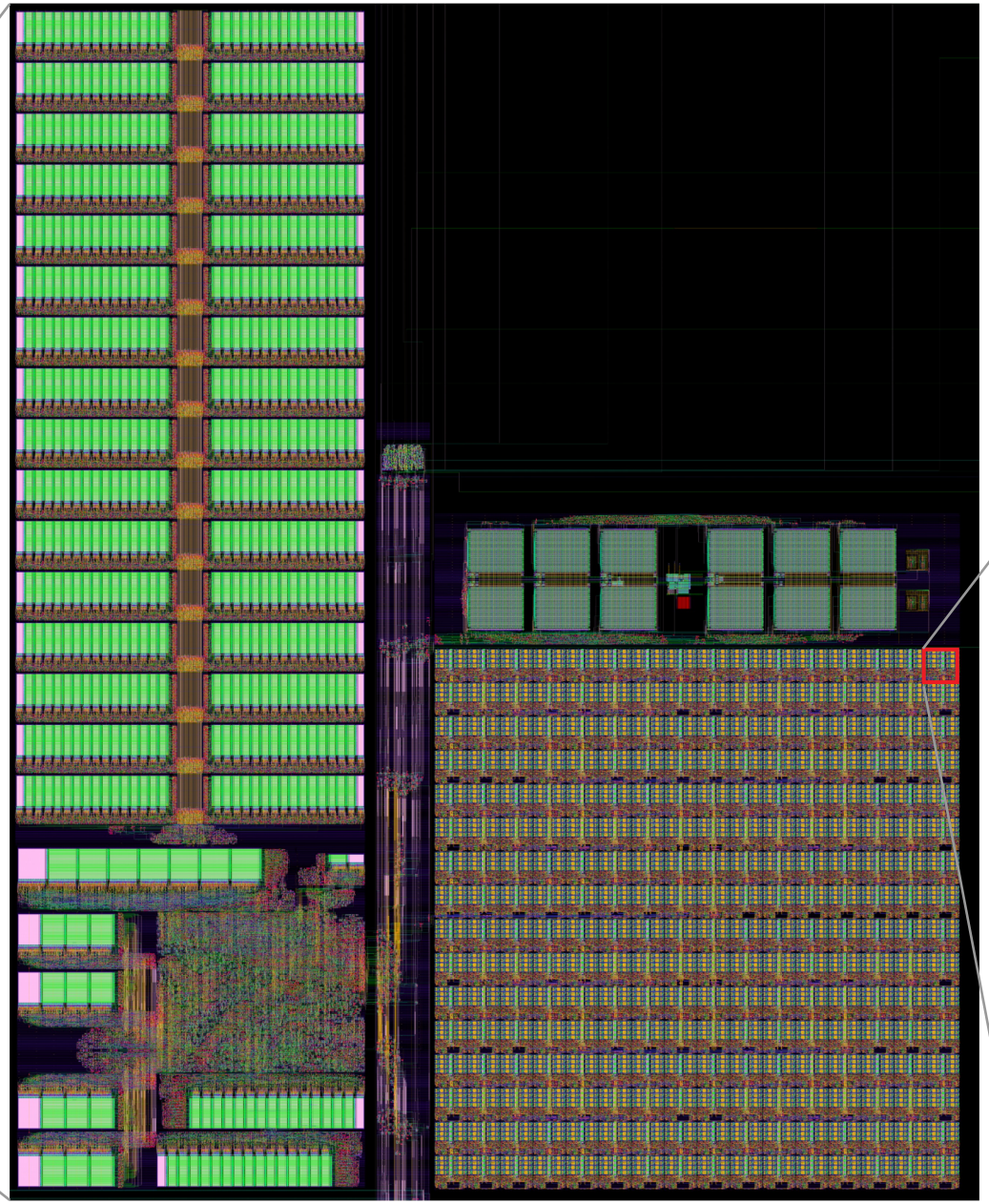




2mm

2mm

Braindrop Chip



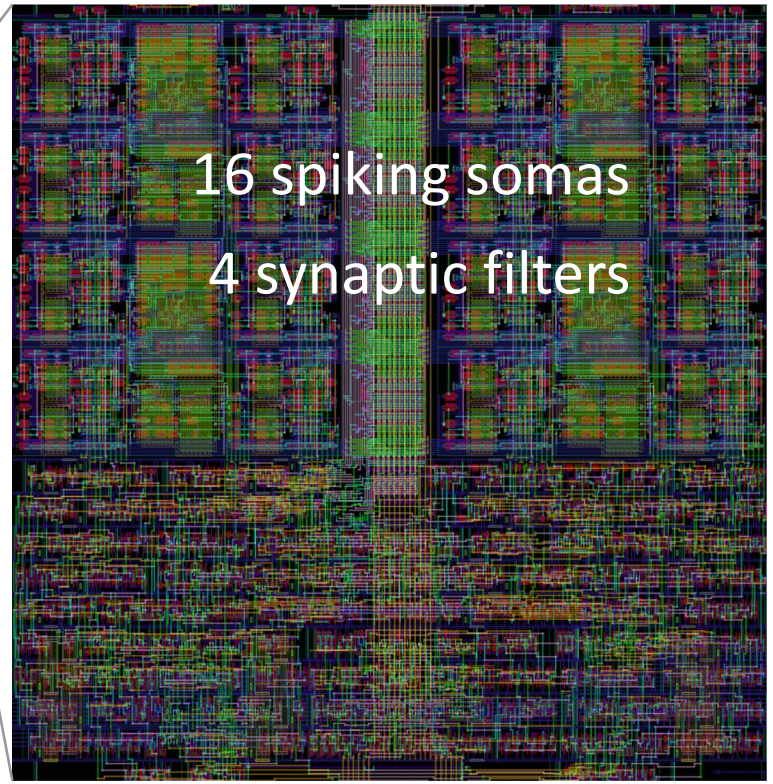
28nm 1P8M FDSOI

4096 Spiking Somas

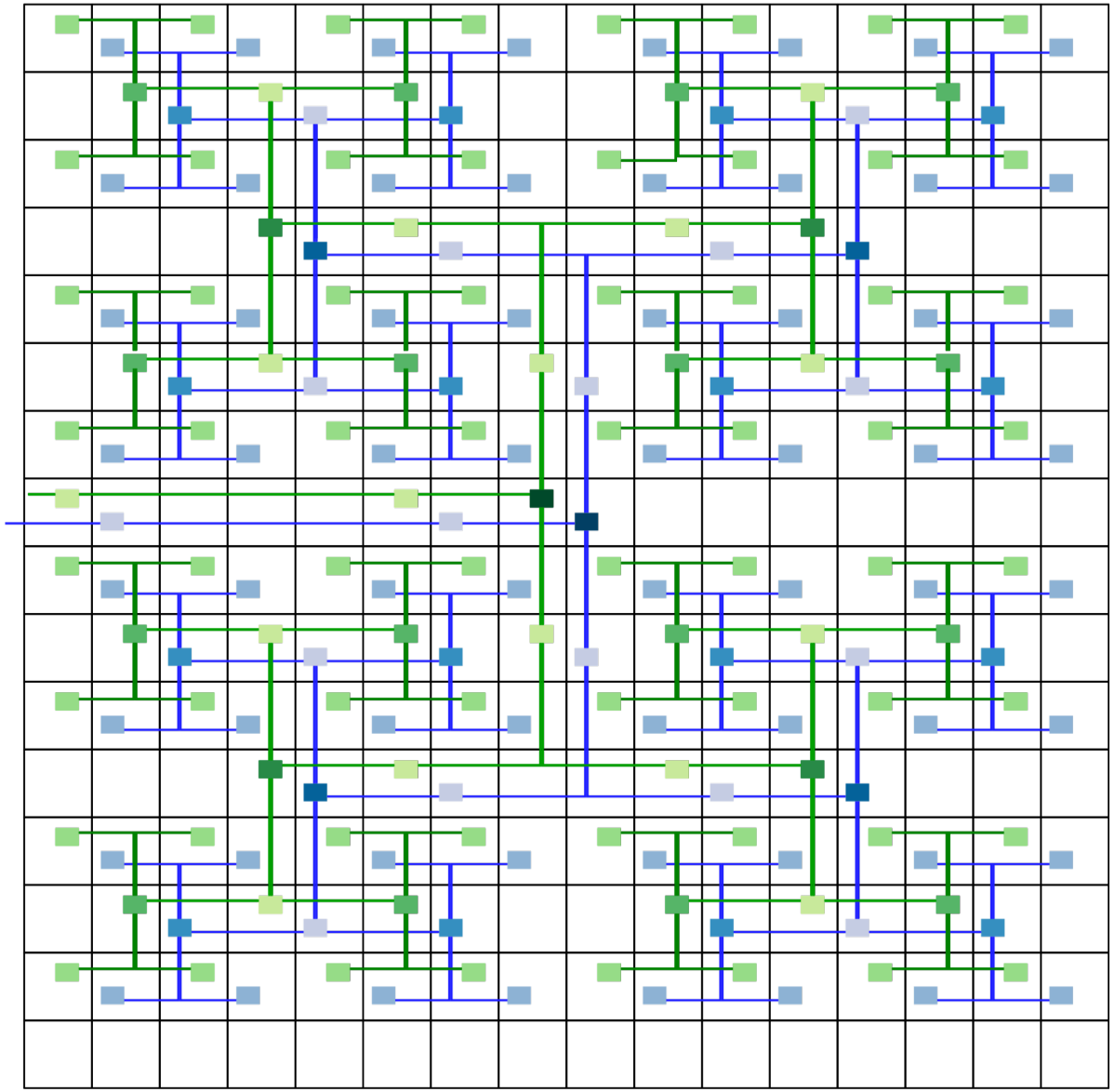
1024 Synaptic Filters

64kB Weight Memory

100kB Total Memory

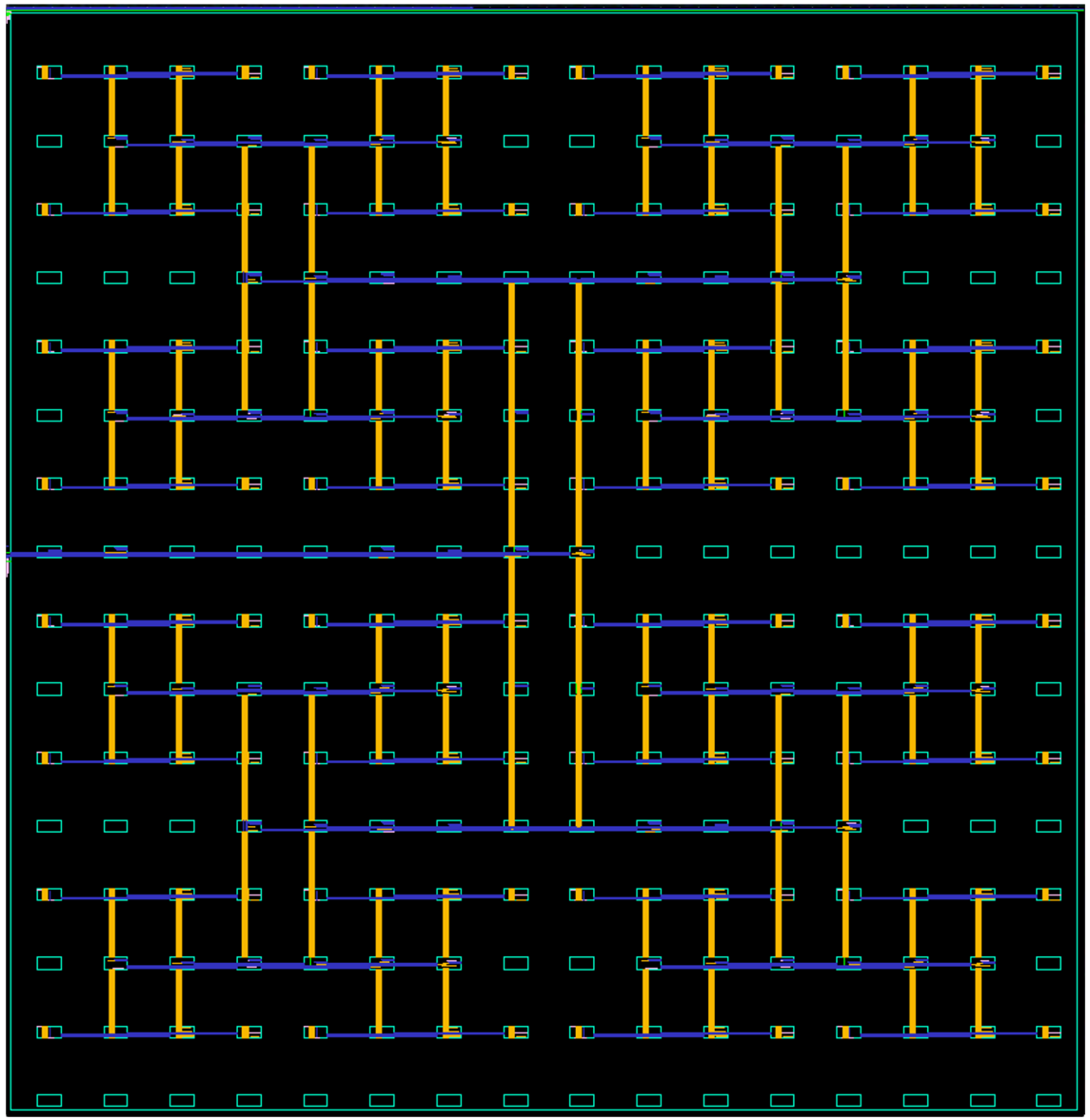


Necker et al. 2019

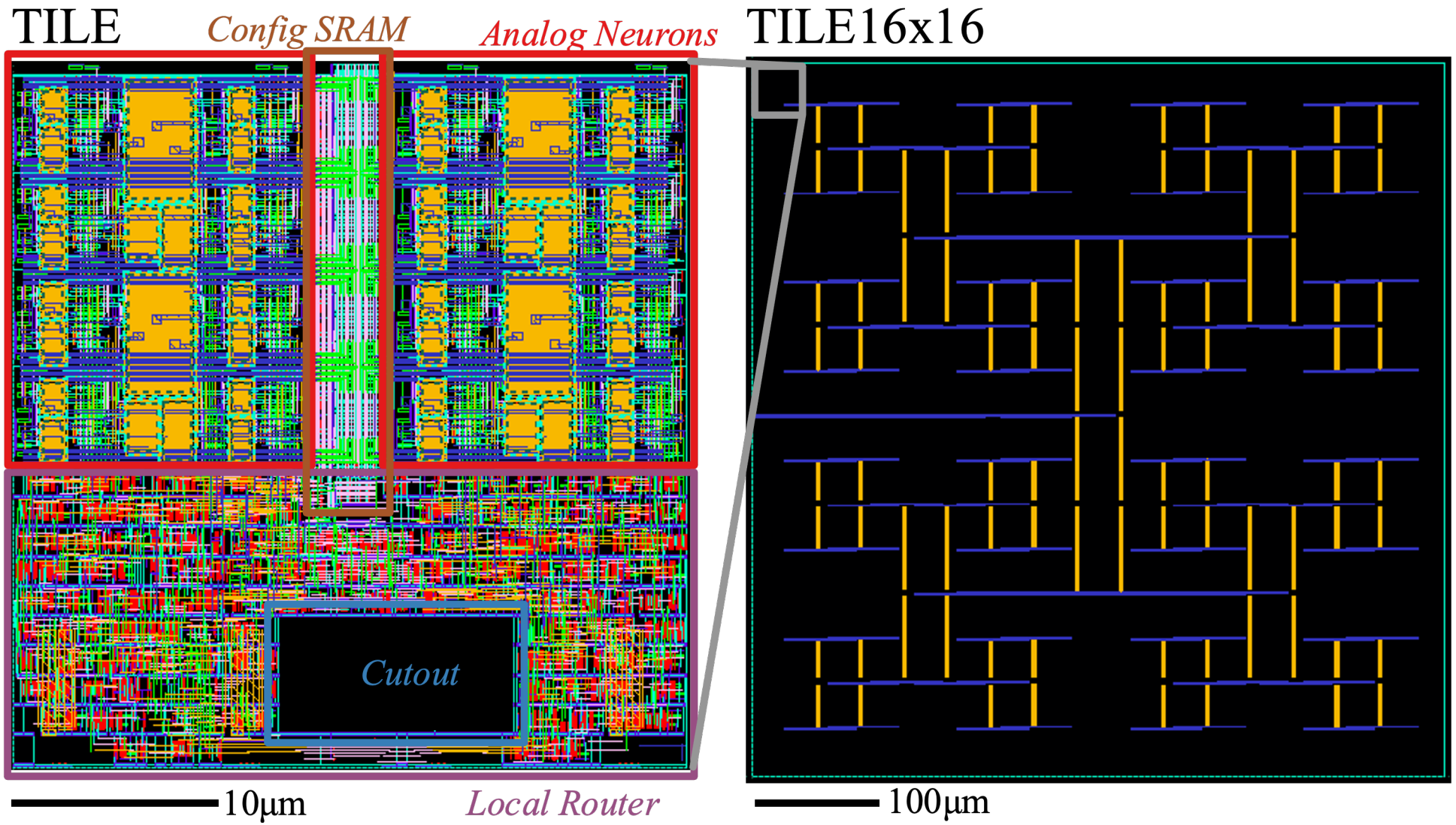


RV Level 5 4 3 2 REP

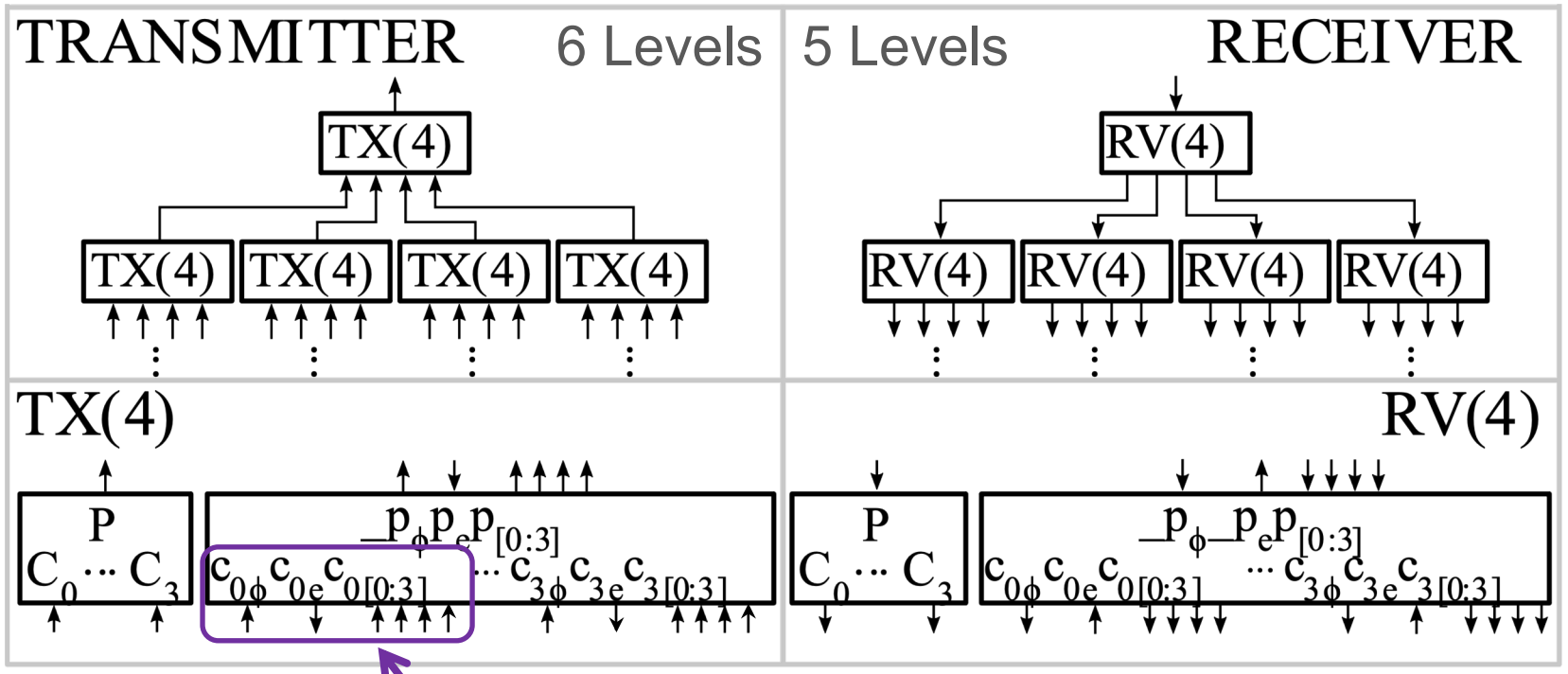
TX Level 6 5 4 3 REP



Layout of H-Trees



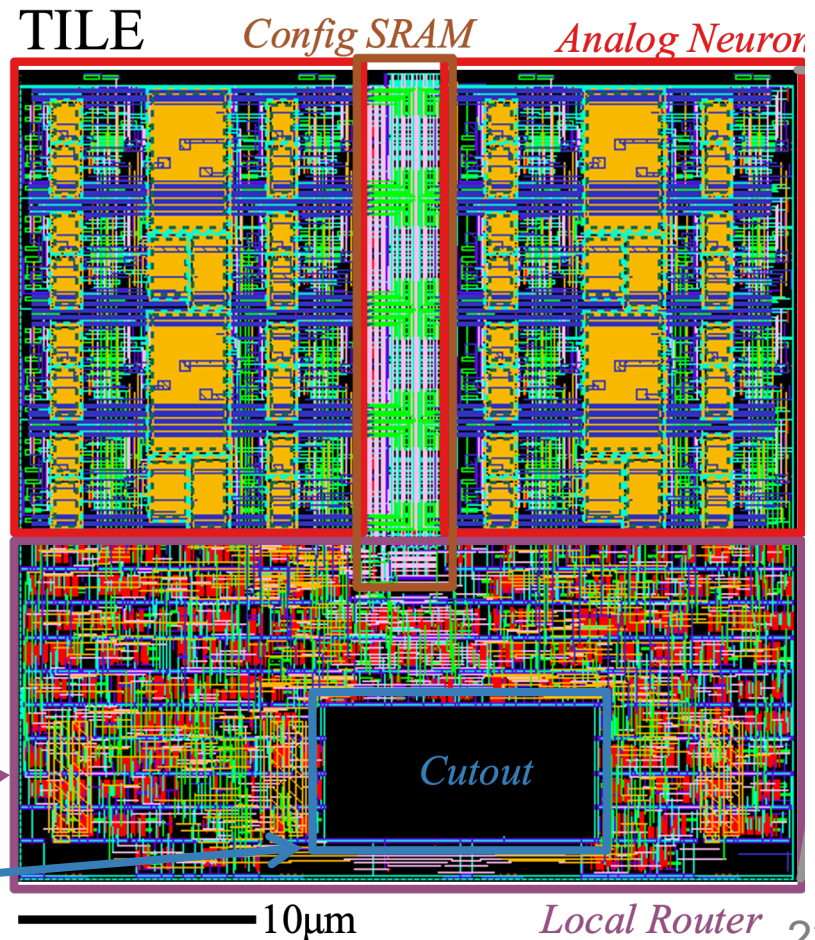
4-ary Trees with 6 and 5 Levels



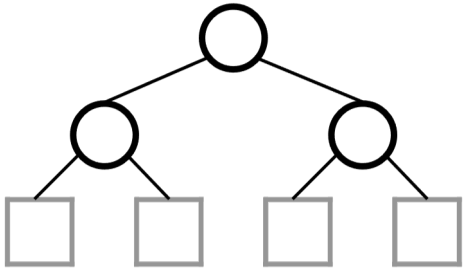
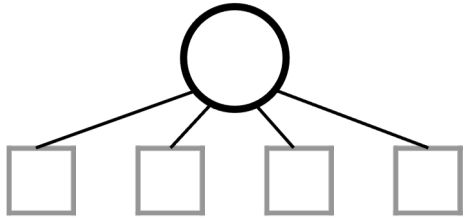
Serial link with six signals

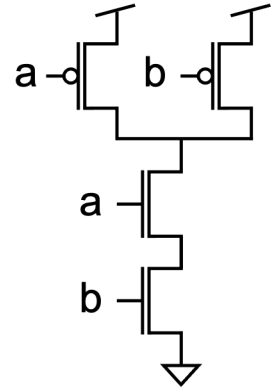
Transmitter L1-2 + Receiver L1

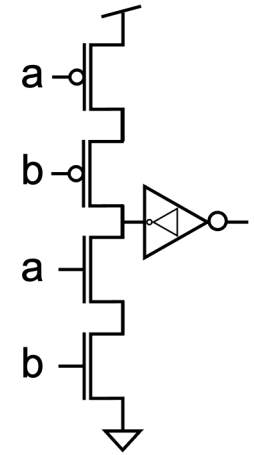
Transmitter L3-6 + Receiver L2-5

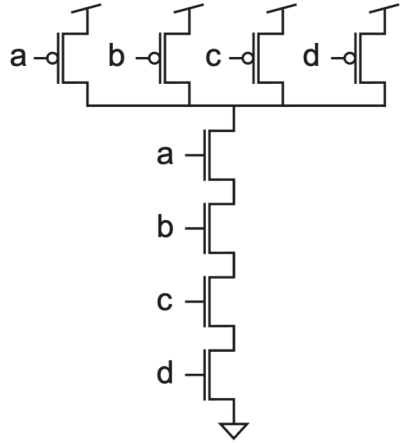


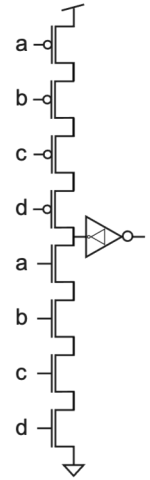
4-ary versus Binary Tree

	Leaves	Nodes	Transistors	
	N	N − 1	4	8
	N	(N − 1)/3	8	12
2-ary:4-ary	1:1	3:1	1:2	2:3







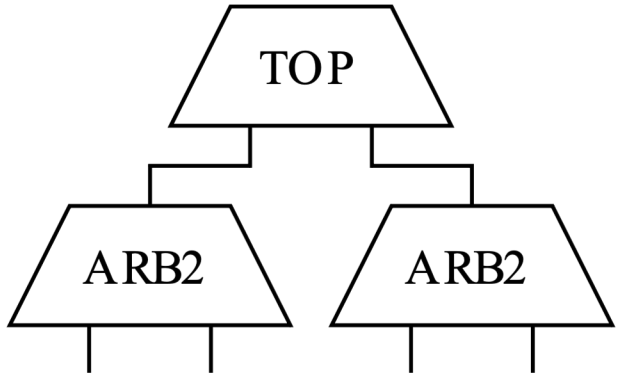


4-ary Cuts Transistors 13% to 45%

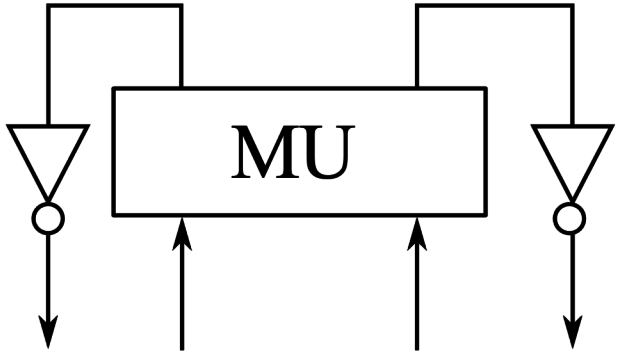
	Transmitter		**Receiver**	
Degree	2-ary	4-ary	2-ary	4-ary
Leaf Node	78	208	30	54
Intermediate Node	91	255	64	148
4-ary / 2-ary	0.867		0.550	

Four-Input Arbiter Cell

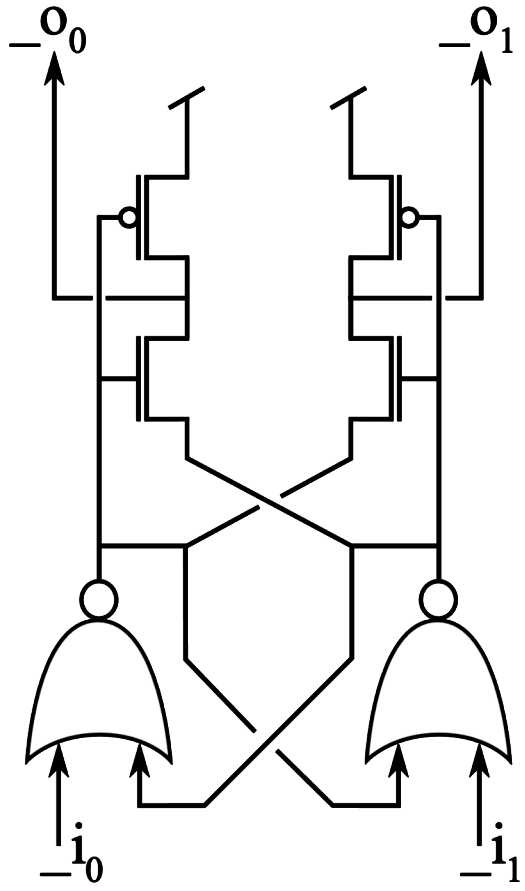
ARB(4)

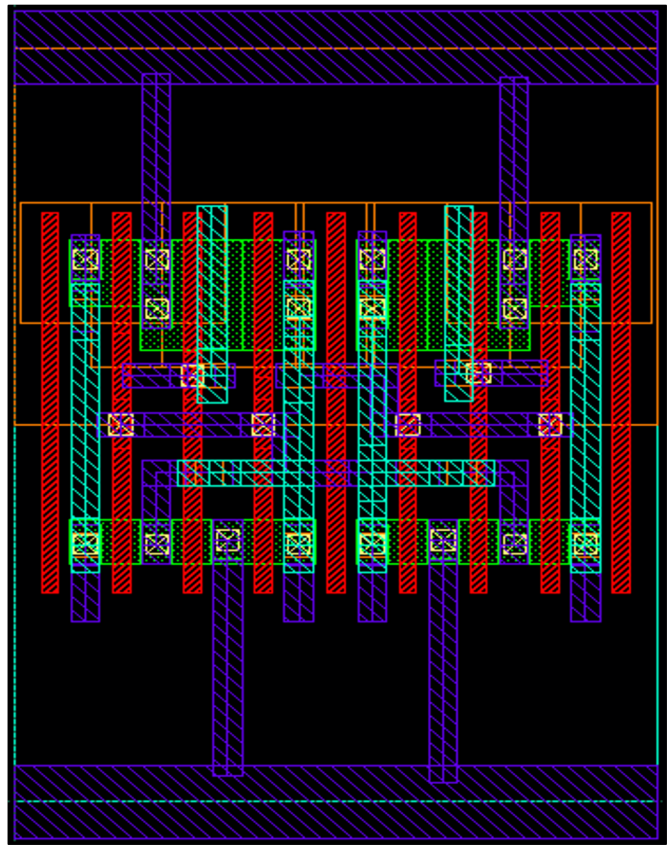


TOP

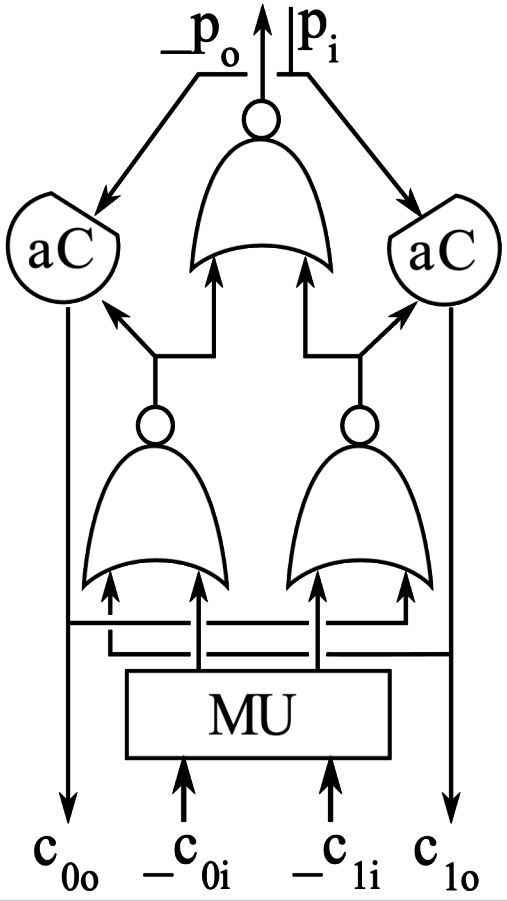


MU

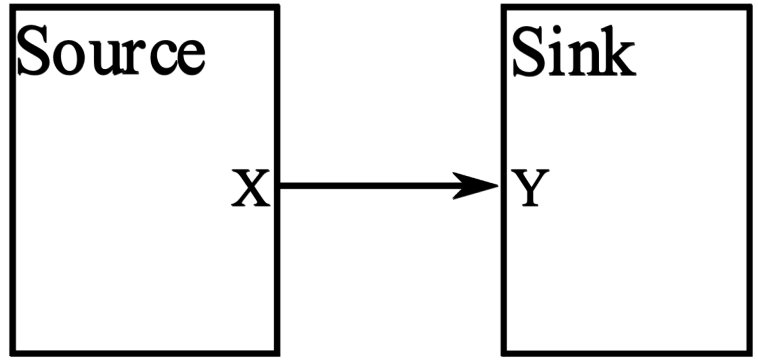


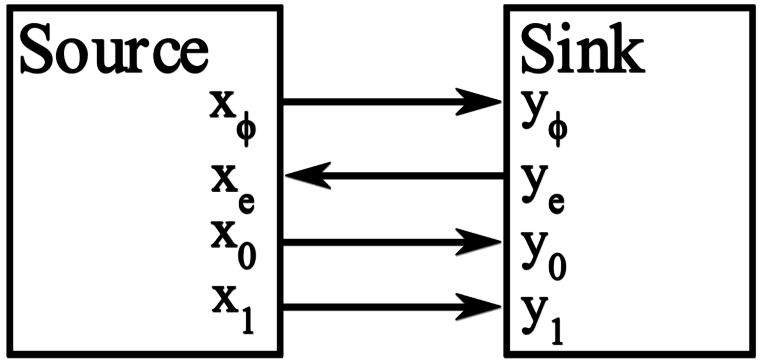


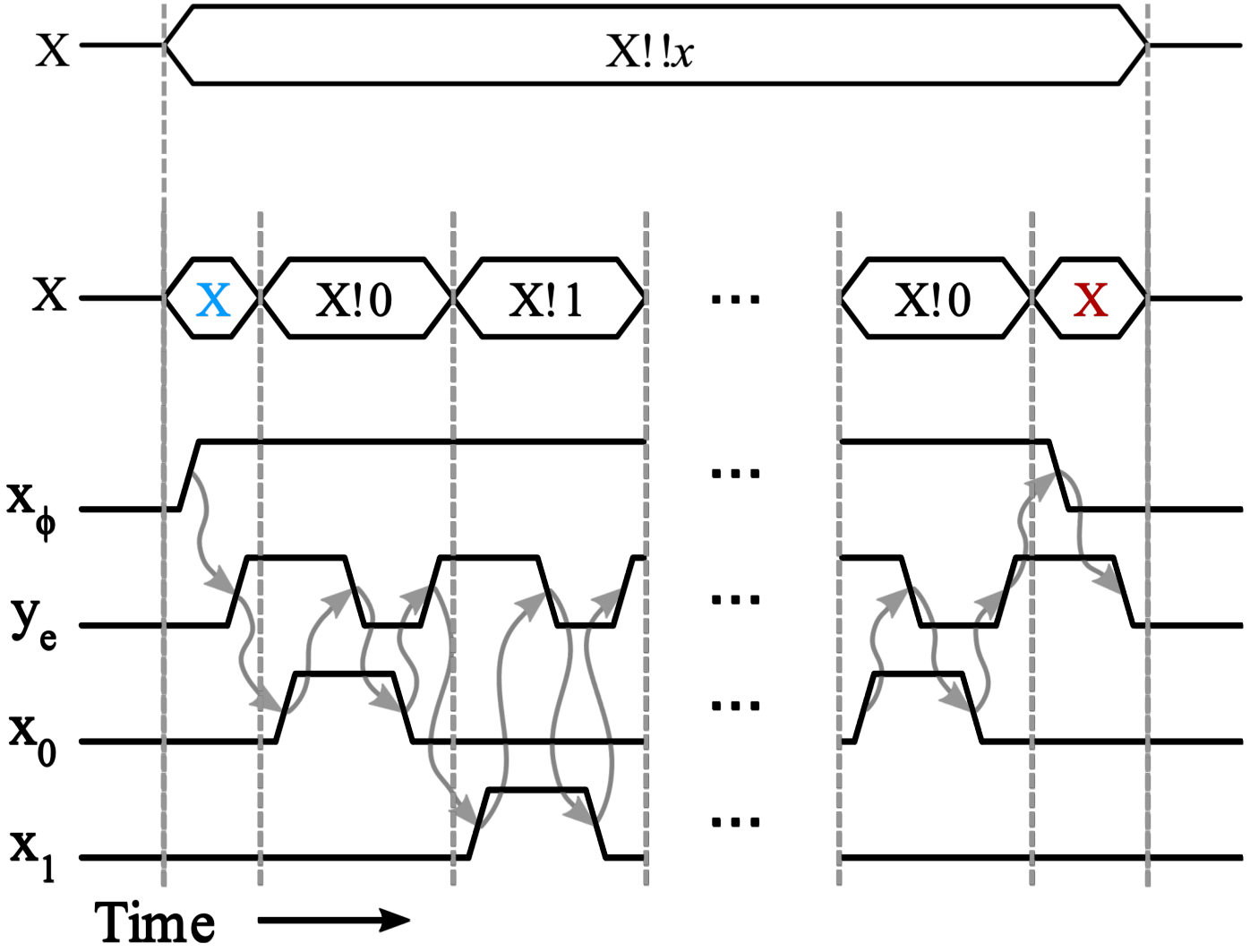
ARB2



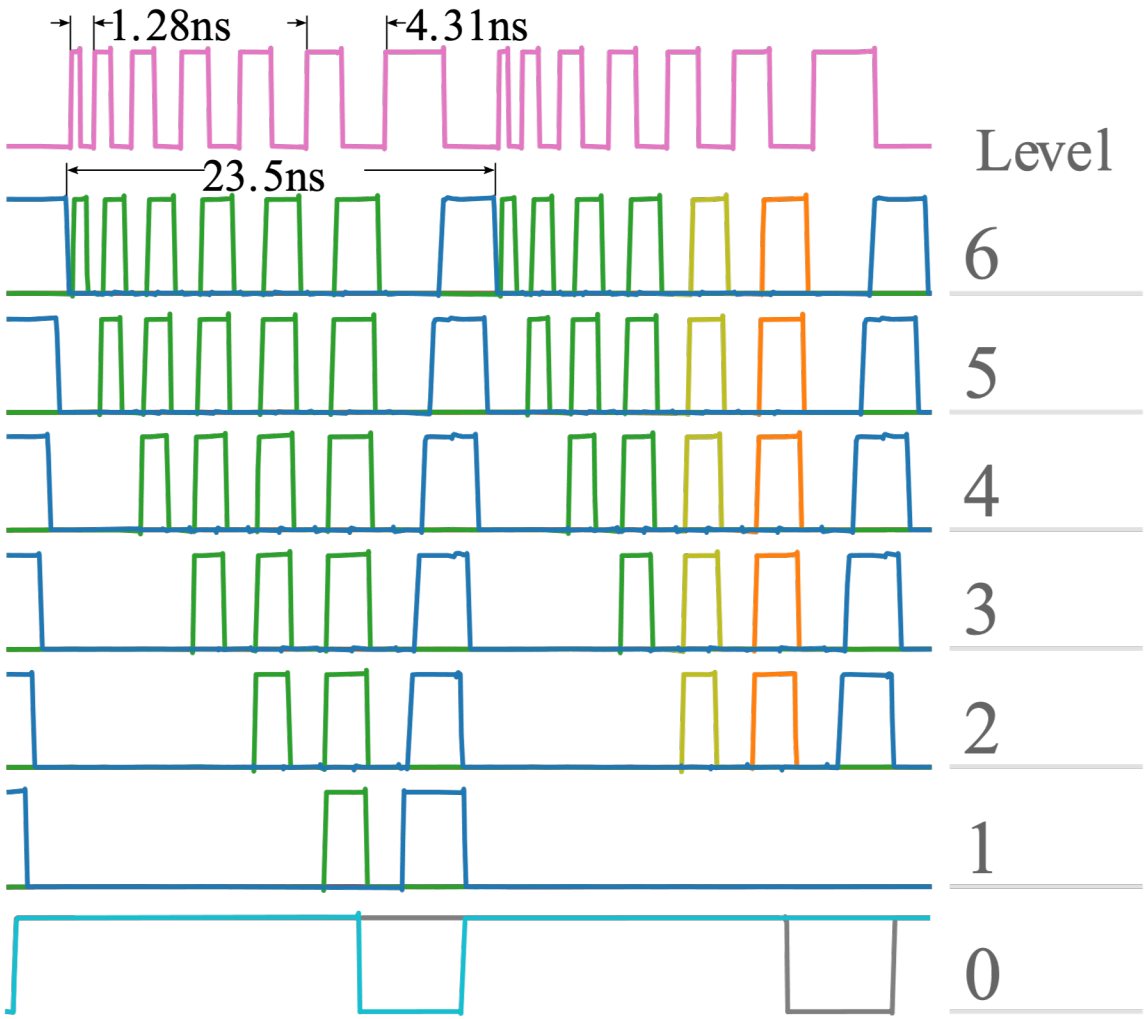
Delay-Insensitive Serial Protocol

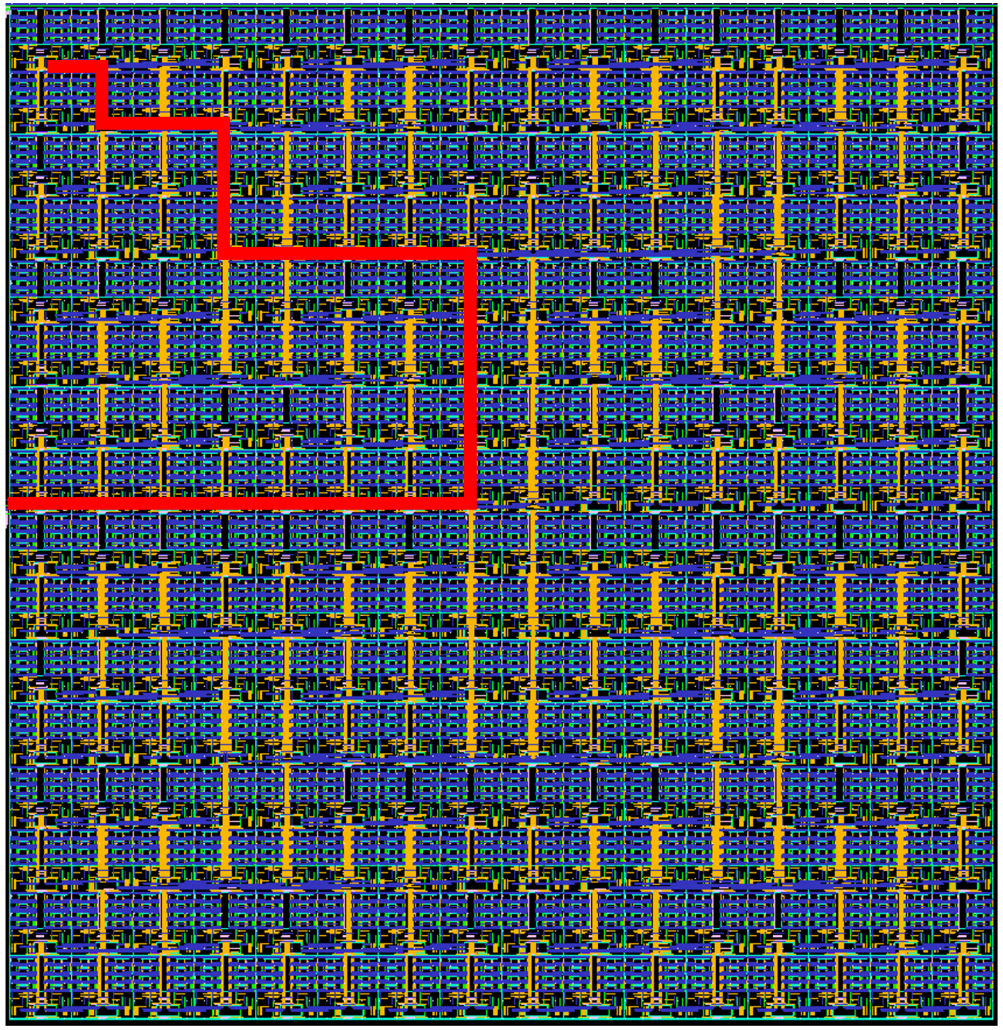




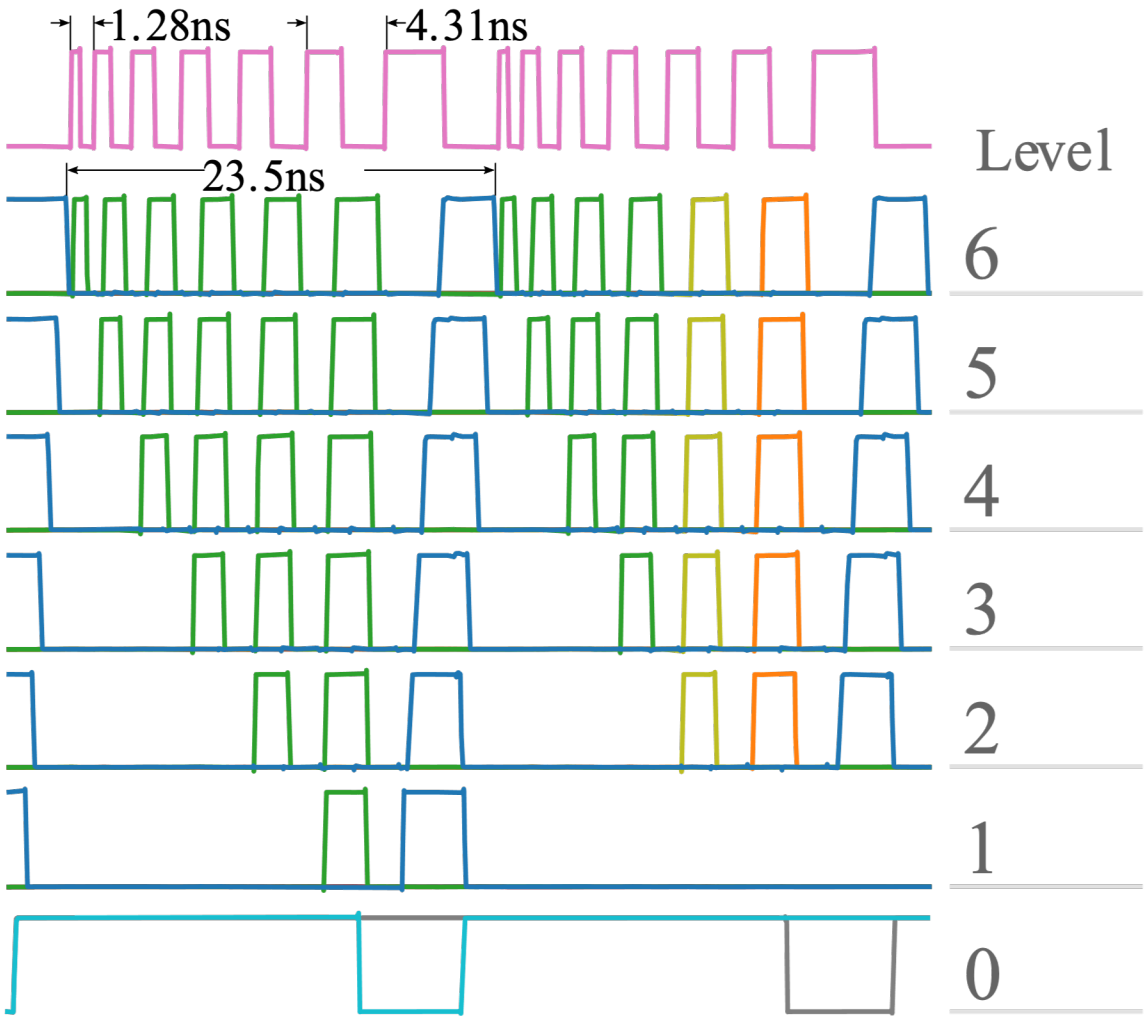


Simulation of Transmitter Layout

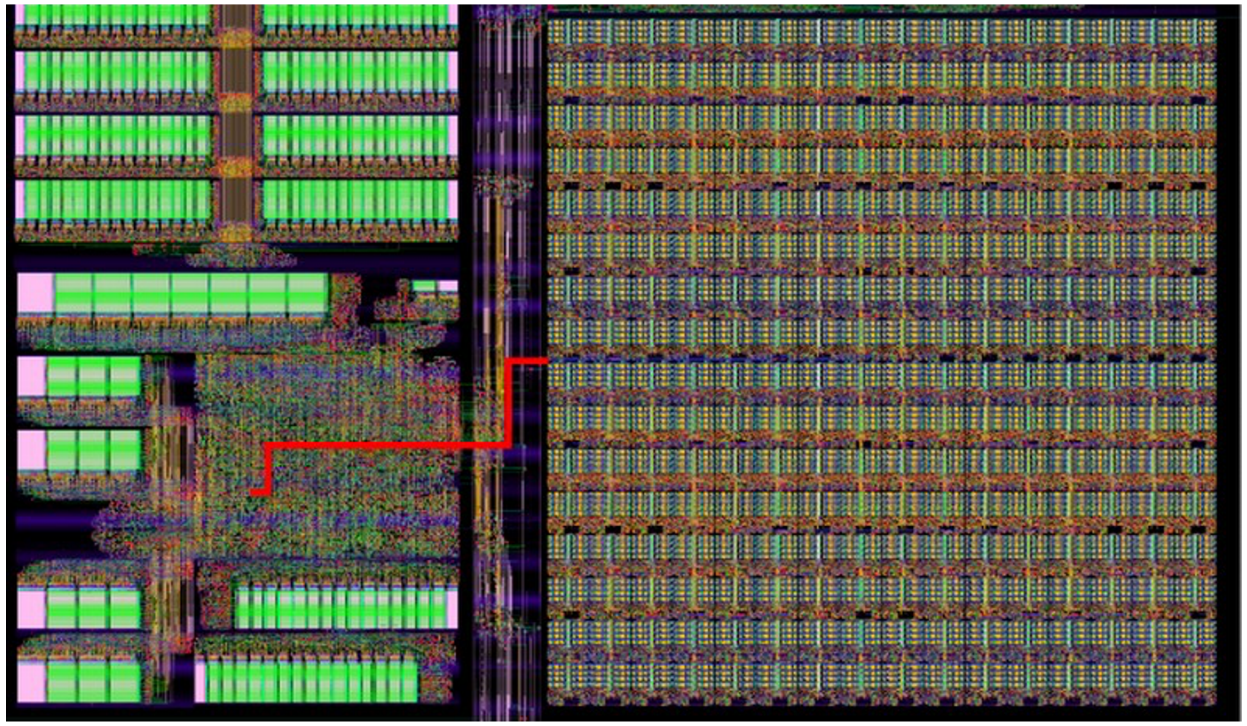




Simulation versus Measurement



Simulation: 42.5 Meps
Measurement: 18.1 Meps

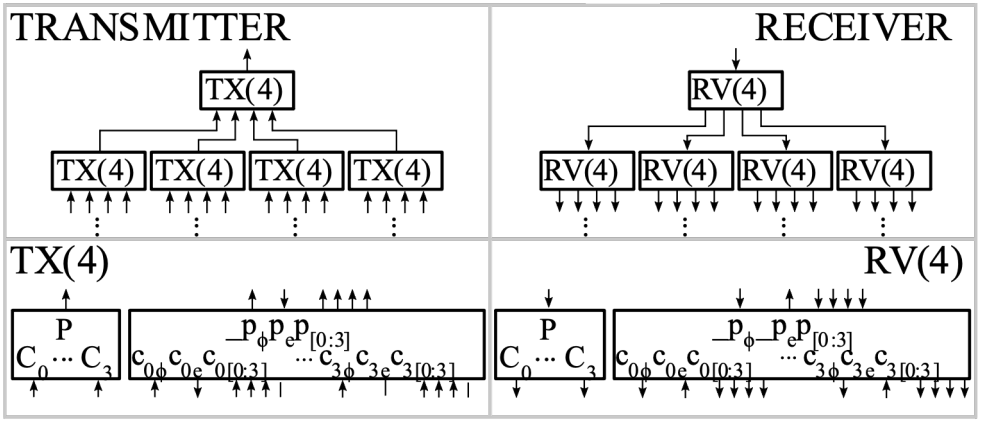


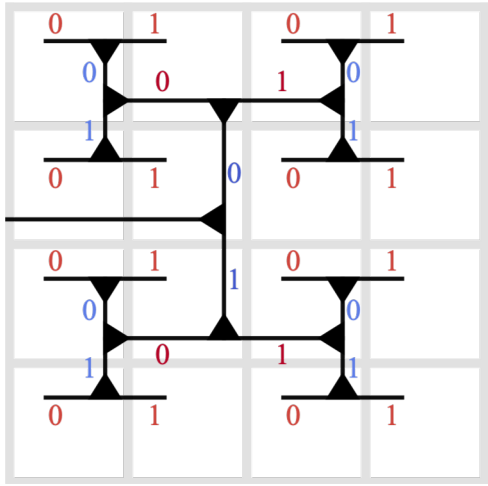
Fok & Boahen 2018

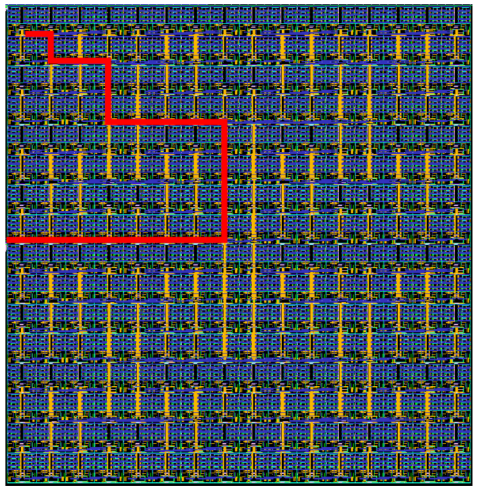
Summary

Proved the concept

- races, matched delays, and wired-ors eliminated
- fully delay-insensitive asynchro-nous implementation
- $t_{\mathrm{cyc}} = 23.5\mathrm{ns}$ supports 256×256-array at 650 eps/pixel







Prospects

Bit-serial (L levels)

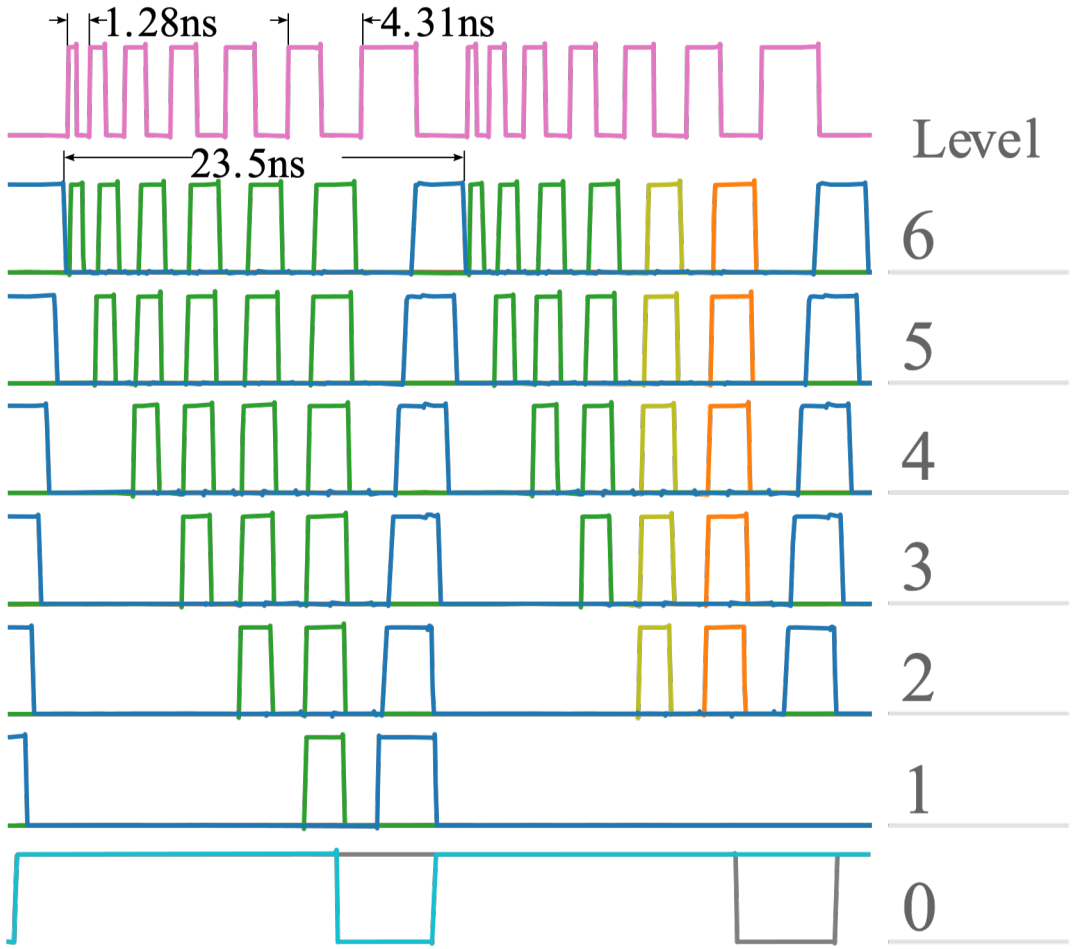
$$t_{\mathrm{cyc}} = t_0 + L(t_1 + (L-1)\,\Delta t/2)$$

$$t_0 = 6.7\mathrm{ns}, t_1 = 1.3\mathrm{ns}, \Delta t = 0.6\mathrm{ns}$$

Pipeline: $t_{\mathrm{cyc}} = (L+1)t_1 = 14.3\mathrm{ns}$

Bit-parallel: $t_{\mathrm{cyc}} = t_1 = 1.3\mathrm{ns}$

supports 1,024×1,024-array at 730 eps/pixel



Thank you for your attention

Funding

GrAI Matter Labs

Acknowledgement

Christoph Posch for video from Prophesee camera.

More info

S Fok and K Boahen, **A Serial H-Tree Router for Two-Dimensional Arrays**, *24th IEEE Symposium on Asynchronous Circuits and Systems,* IEEE Press, 2018.
www.github.com/samfok/AER_serial_tree_router