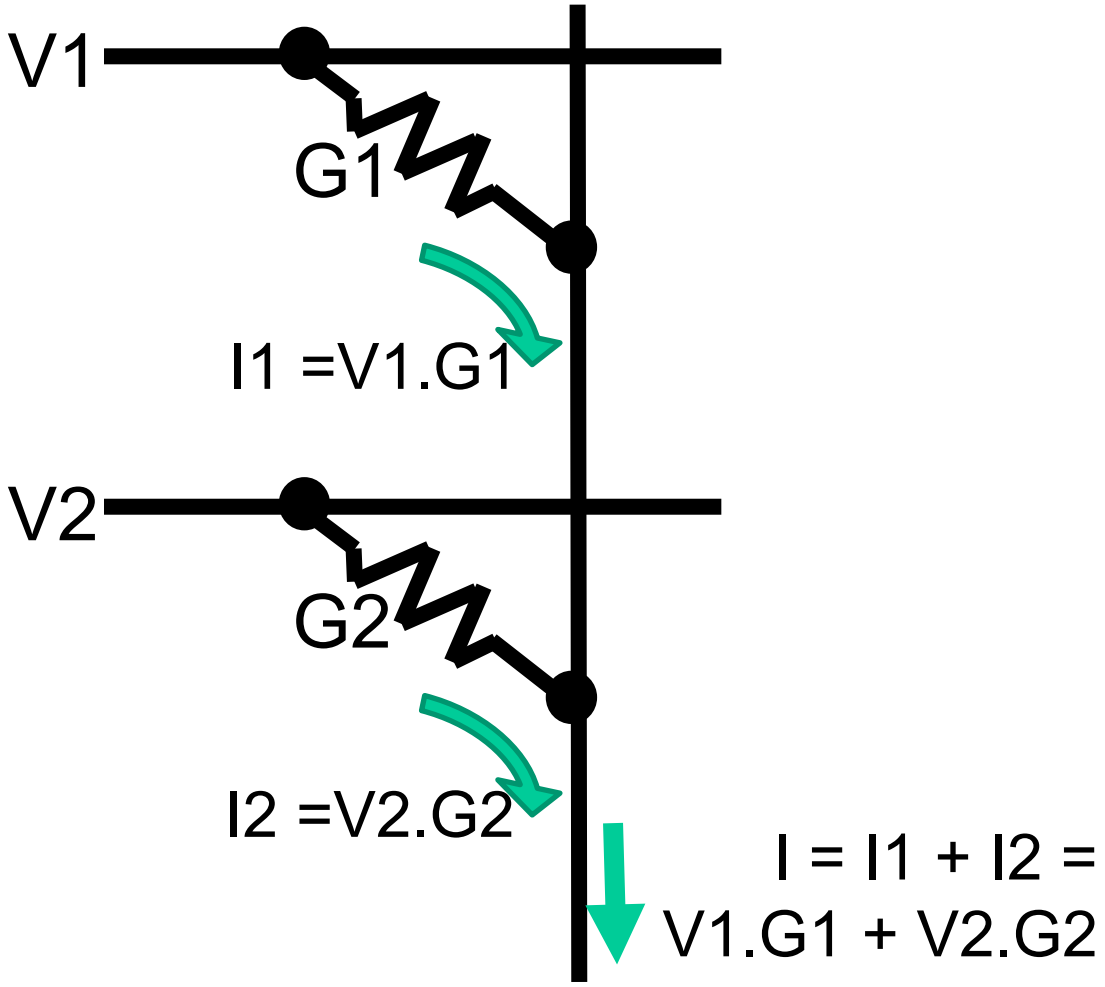
Lecture: Analog Accelerator

- Topics: memristor basics, ISAAC accelerator

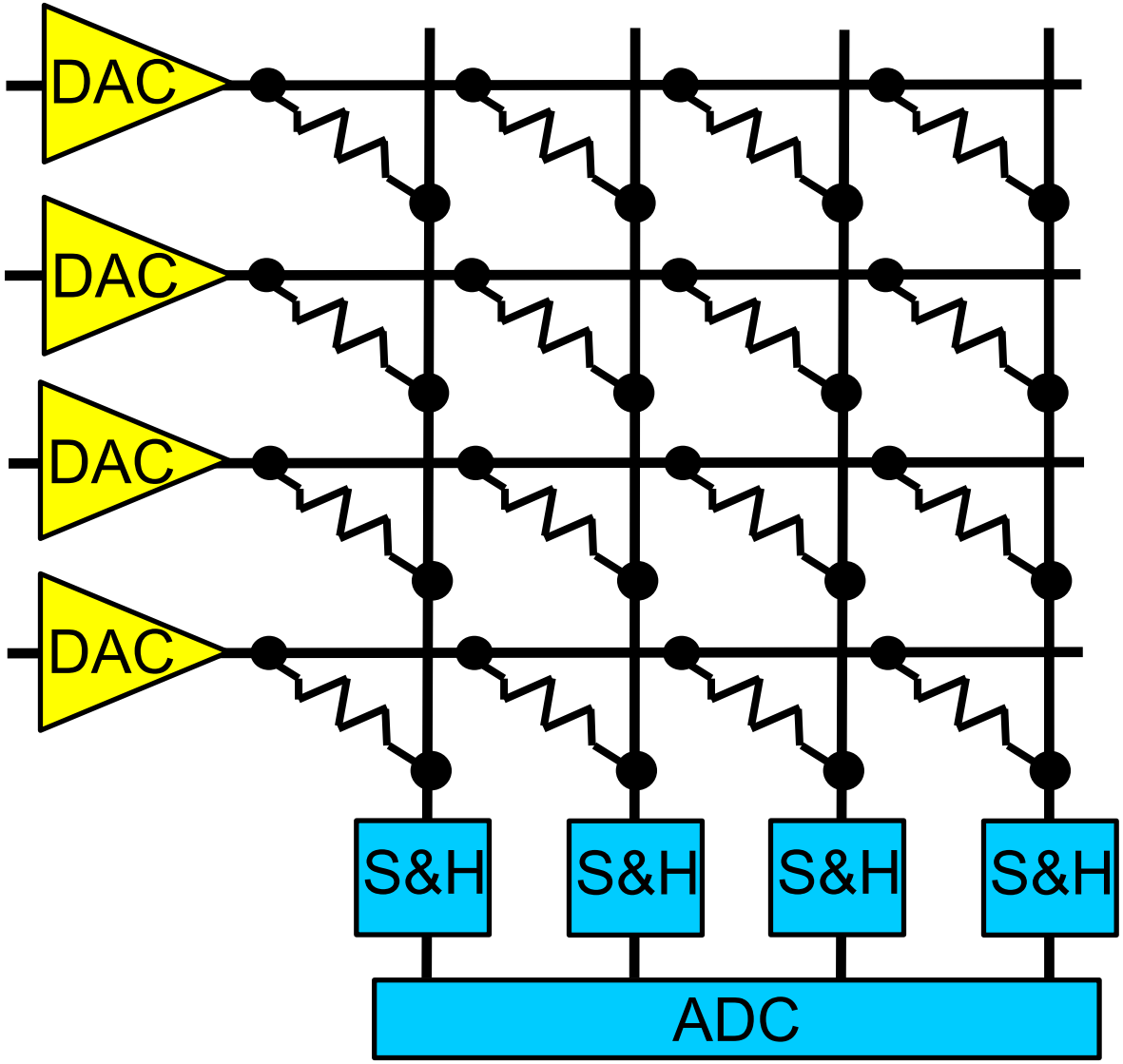
Analog Acceleration

- Many electronic phenomena correspond to multiplication and addition
- Analog phenomena are also noisy; perhaps, this is not an issue when dealing with neural networks

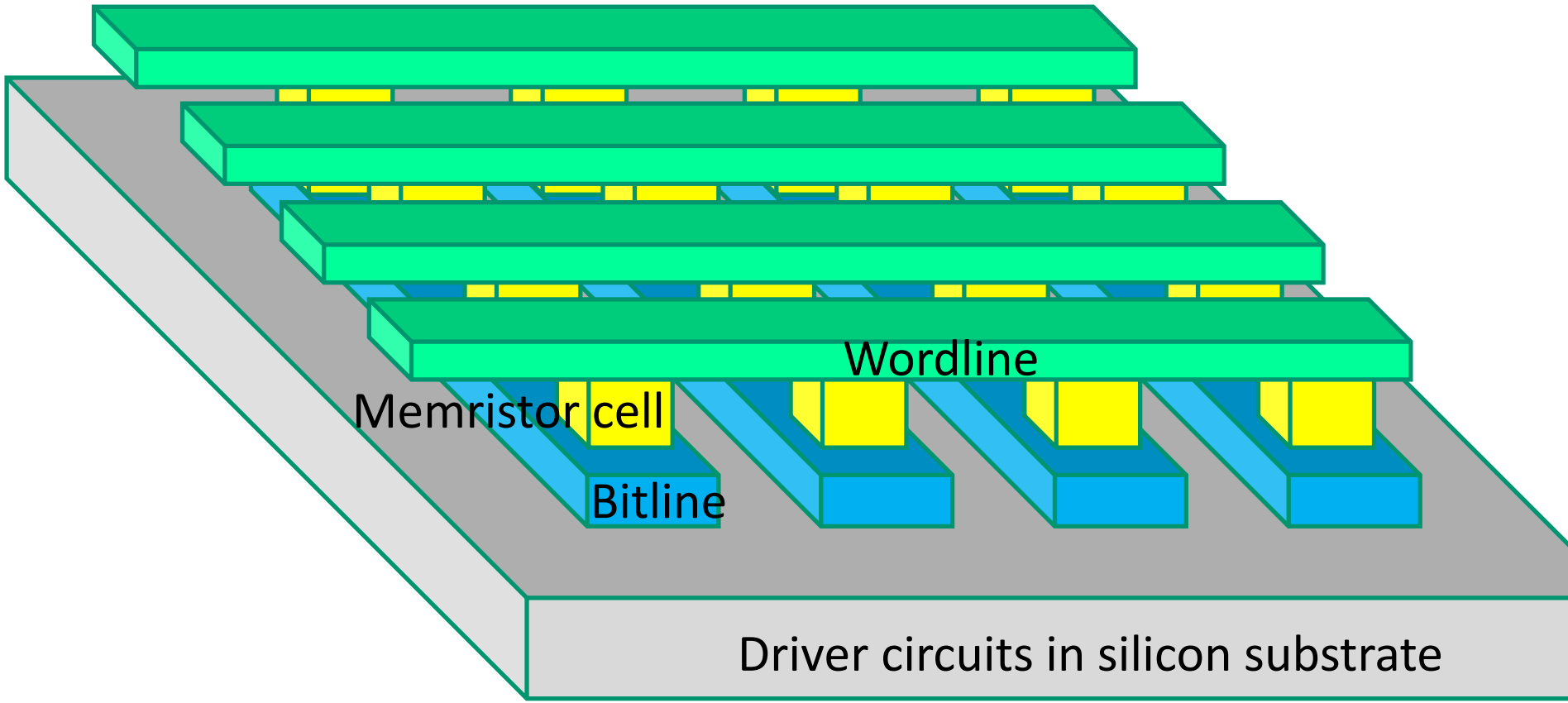
Wires as ALUs



Crossbar for Vector-Matrix Multiplication



Physical View



Physical view of a memristor crossbar array

Challenge

- High ADC/DAC area/energy
  - You could stay in analog forever, but then you'd need expensive analog buffering and you'd introduce significant noise that accumulates across network layers
- Unfortunately, some ADC overheads increase exponentially with resolution
- Resolution increases with computational density

$$A = log(R) + v + w, \quad if\ v > 1\ and\ w > 1$$

$$A = log(R) + v + w - 1, \quad otherwise$$

1. Input One Bit at a Time

- Need a trivial DAC
- Must perform multiplication over 16 iterations
- Results are aggregated with shift-and-adds

2. Spread the Weights

- A single weight is spread across 8 2-bit cells in a row
- The outputs of 8 columns have to be shifted and added
- Low bits per cell is good for precision and for ADC efficiency

3. Few Rows Per Crossbar

- Requires us to use many small crossbars
- A neuron with many inputs is spread across multiple xbars
- Must aggregate partial sums from many xbars

4. Weight Encoding

- If the weights are large, store their "complements"
- Reduces ADC resolution by 1 bit
- Inputs are provided in 2's complement form
  - The MSb represents -2^{15} -- need a shift-and-subtract
  - Irrelevant if we are using ReLU
- Weights are stored with a bias: a bias of 2^{15} allows unsigned integers to represent weights between -2^{15} and $2^{15} - 1$

Analog Accelerator Challenge

- High ADC/DAC area/energy

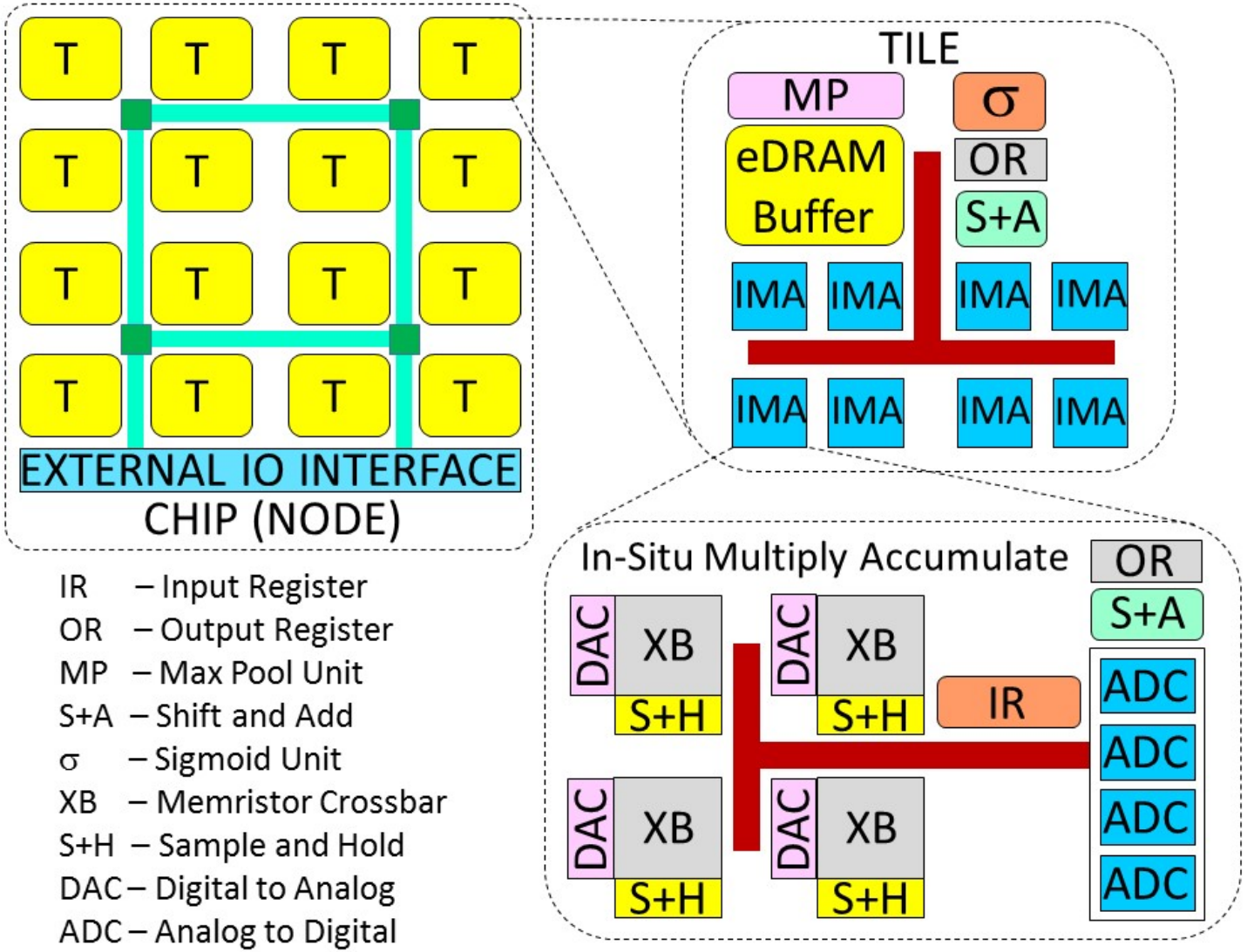
$$A = log(R) + v + w, \quad if\ v > 1\ and\ w > 1$$

$$A = log(R) + v + w - 1, \quad otherwise$$

1. 1-bit input at a time (small *v*)
2. 2-bit cells (small *w*)
3. Few rows per array (small *R*)
4. Encoding tricks to produce small numbers

Spread the computation across a single xbar, across multiple xbars, and across time to reduce ADC size

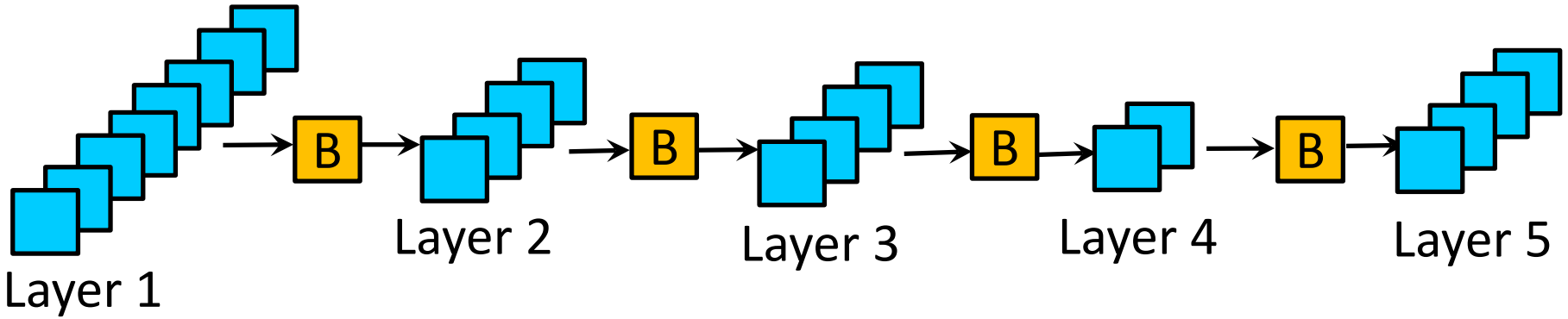
ISAAC Architecture



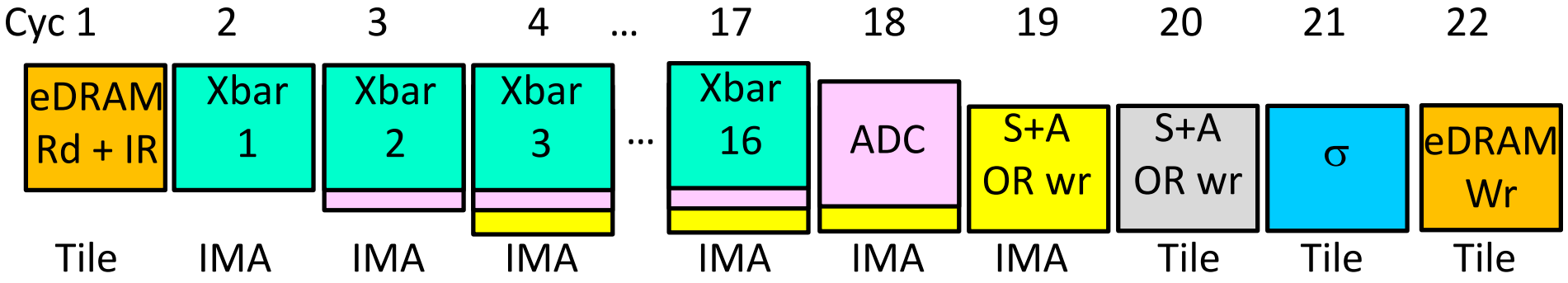
IR – Input Register
OR – Output Register
MP – Max Pool Unit
S+A – Shift and Add
σ – Sigmoid Unit
XB – Memristor Crossbar
S+H – Sample and Hold
DAC – Digital to Analog
ADC – Analog to Digital

ISAAC Pipeline

(a) Example of different layers in action at the same time



(b) Example of one operation in layer *i* flowing through its pipeline



Pipeline Variants

- Most digital accelerators use “temporal pipelines” – all units work on 1 layer, then all work on the next layer, etc.
  - Good for low latency and cache locality
  - A spatial pipeline would give nearly the same throughput, but higher latency per inference

- Analog accelerators use “spatial pipelines” – parts of the chip are hard-coded to execute specific layers
  - Required by design since weight updates are slow
  - Latency impact is small (no batching required and for the most part, all layers work on the same image)

Replication in Early Layers

The ISAAC Pipeline

- Pipelining within an IMA/tile/layer
- Pipelining across layers
- Network is mapped to avoid hazards; balanced replication where possible to avoid storage/compute under-utilization
- Design space exploration to identify the best use of chip real estate

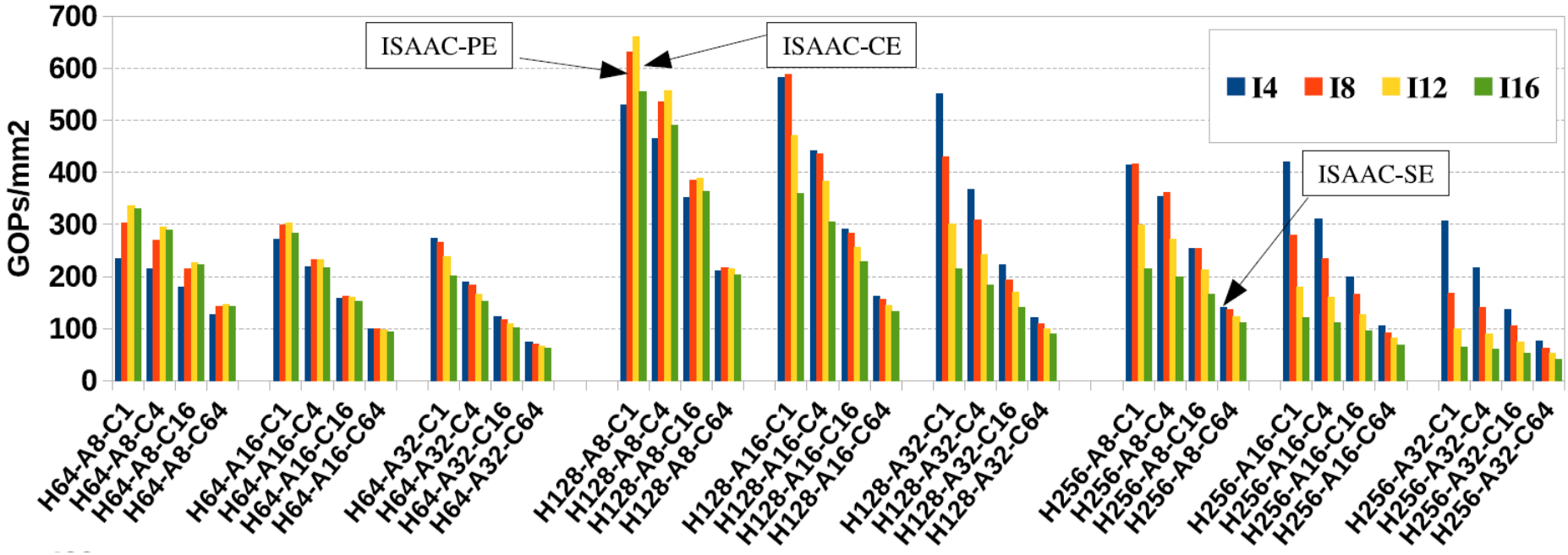
Power/Area Breakdowns

ISAAC Tile at 1.2 GHz, 0.37 mm^2				
Component	**Params**	**Spec**	**Power**	**Area** (mm^2)
eDRAM Buffer	size num_banks bus_width	64KB 4 256 b	20.7 mW	0.083
eDRAM -to-IMA bus	num_wire	384	7 mW	0.090
Router	flit size num_port	32 8	42 mW	0.151 (shared by 4 tiles)
Sigmoid S+A MaxPool OR	number number number size	2 1 1 3 KB	0.52 mW 0.05 mW 0.4 mW 1.68 mW	0.0006 0.00006 0.00024 0.0032
Total			**40.9 mW**	**0.215** mm^2

Power/Area Breakdowns

IMA properties (12 IMAs per tile)				
ADC	resolution frequency number	8 bits 1.2 GSps 8	16 mW	0.0096
DAC	resolution number	1 bit 8×128	4 mW	0.00017
S+H	number	8×128	10 uW	0.00004
Memristor array	number size bits per cell	8 128×128 2	2.4 mW	0.0002
S+A IR OR	number size size	4 2 KB 256 B	0.2 mW 1.24 mW 0.23 mW	0.00024 0.0021 0.00077
IMA Total	number	12	**289 mW**	**0.157** mm^2
1 Tile Total **168 Tile Total**			**330 mW** **55.4 W**	**0.372** mm^2 **62.5** mm^2
Hyper Tr	links/freq link bw	4/1.6GHz 6.4 GB/s	10.4 W	22.88
Chip Total			**65.8 W**	**85.4** mm^2

Design Space Exploration

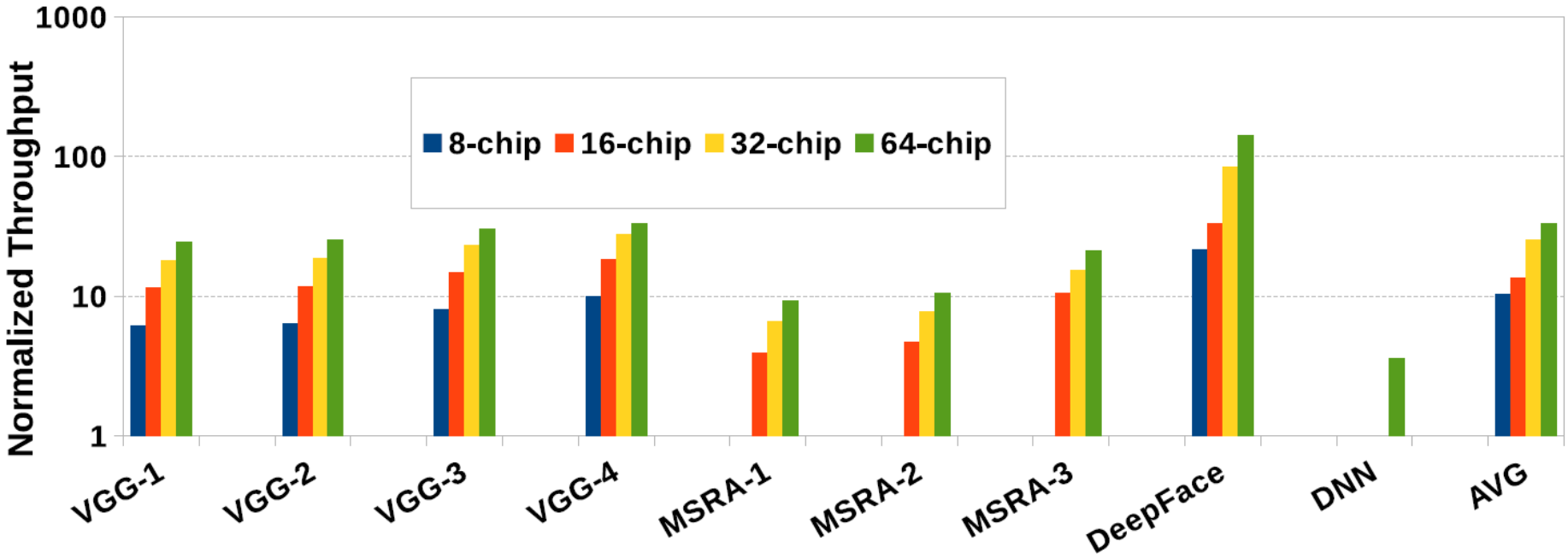


Comparison to DaDianNao

Architecture	CE $GOPs/(s \times mm^2)$	PE $GOPs/W$	SE MB/mm^2
DaDianNao	63.46	286.4	0.41
ISAAC-CE	478.95	363.7	0.74
ISAAC-PE	466.8	380.7	0.71
ISAAC-SE	140.3	255.3	54.8

- 7.5X higher computational density
- 14.8X higher throughput on CNN benchmarks
- 5.5X lower energy
- The chip has a 3X higher power density

Throughput on CNNs



Other Analog Innovations

- AN codes for reliability (Feinberg et al., HPCA'18)
- Crossbars applied to scientific computing (Feinberg et al., ISCA'18)
- More efficient ADCs (e.g., PipeLayer, HPCA'17)
- Memristor-aided logic (Kvatinsky et al., IEEE Trans. On Circuits and Systems, 2014) – activate two rows and ground a third row to perform a NOR operation within the crossbar

References

- “ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars”, A. Shafiee et al., Proceedings of ISCA, 2016