Proteus: Exploiting Numerical Precision Variability in Deep Neural Networks

Patrick Judd, Jorge Albericio, Tayler Hetherington, Tor Aamodt, Natalie Enright Jerger and Andreas Moshovos.

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

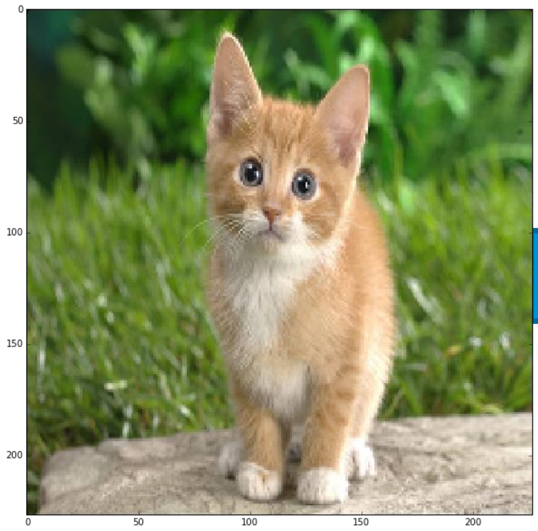
Motivation

- Deep Neural Networks (DNNs) are machine learning algorithms that are state of the art for a range of complex tasks
- Computationally demanding, large memory footprints
  - E.g. VGG-19: **19 Billion FLOPs** and **576 MB** [1]
- DNNs are very tolerant to approximation
- Big opportunity to improve performance and efficiency via approximate computing

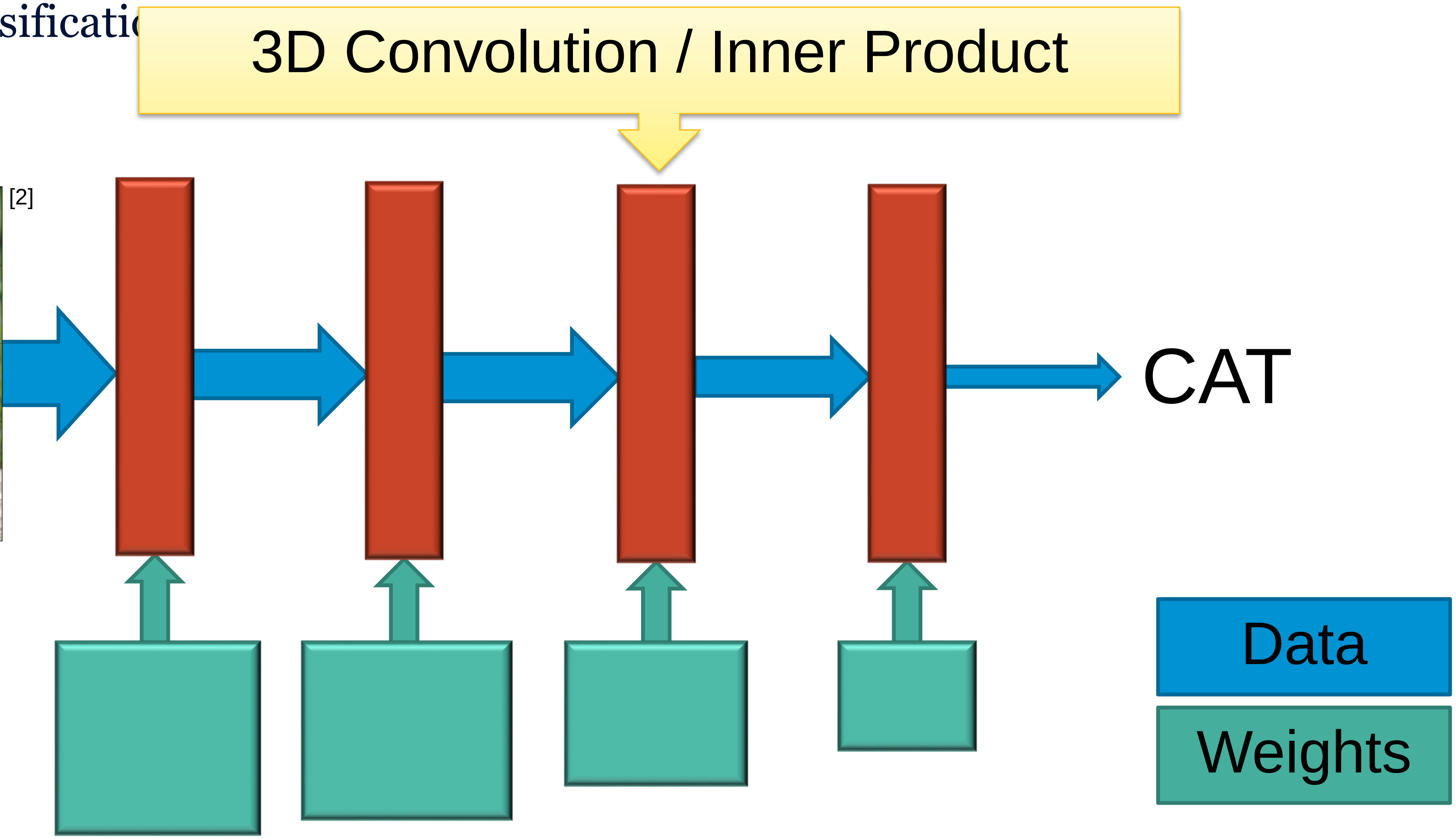
[1] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556*

Deep Neural Networks

- Applies successive layers of computation using learned weights to perform difficult tasks such as:
  - Speech recognition
  - Image classificati



[2]

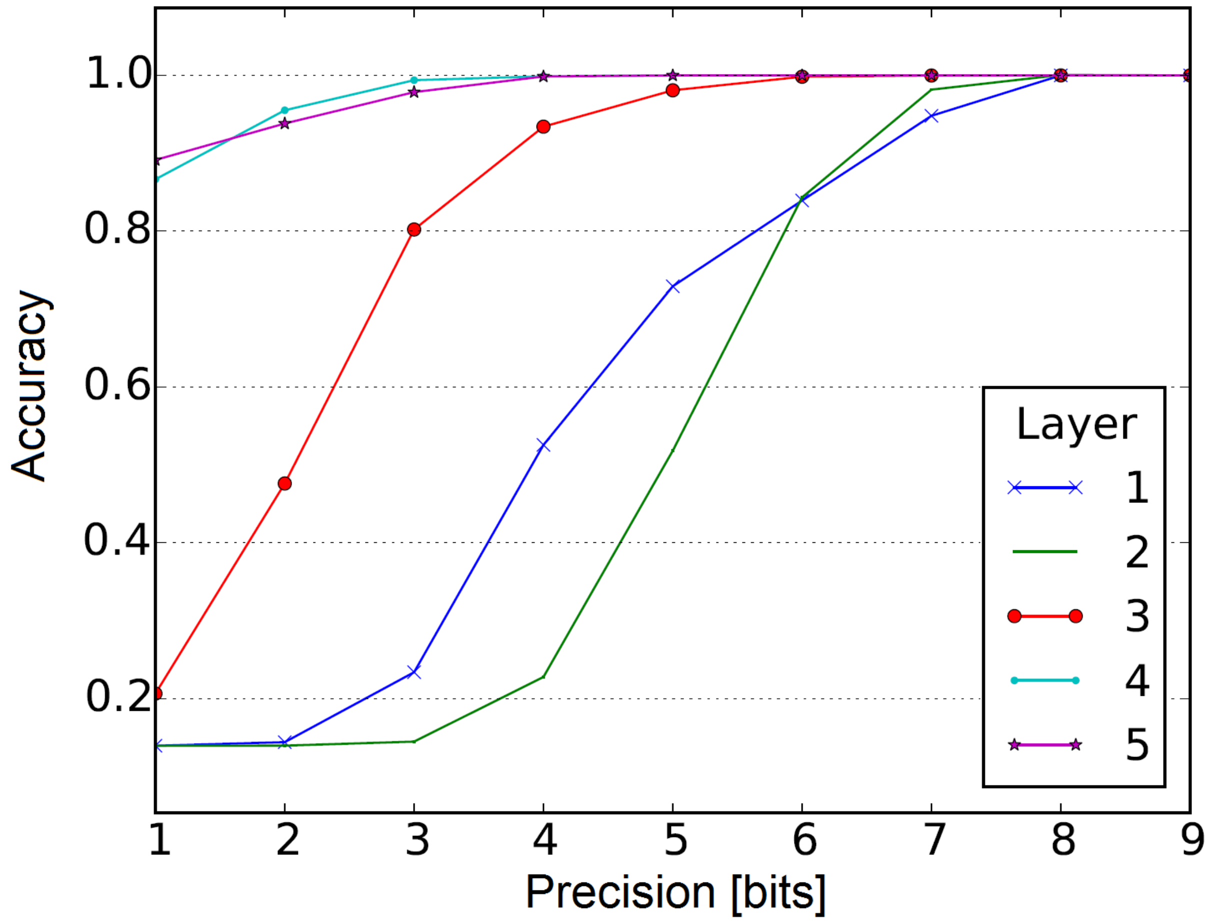


[2] https://github.com/BVLC/caffe/blob/master/examples/00-classification.ipynb

Prior Work

In prior work [3], we analysed the sensitivity of DNNs to reducing the precision of **fixed-point** representations for **weights** and **data**.

00000110.10110000



[3] Judd et al. “Reduced-Precision Strategies for Bounded Memory in Deep Neural Nets,” *arXiv:1511.05236*

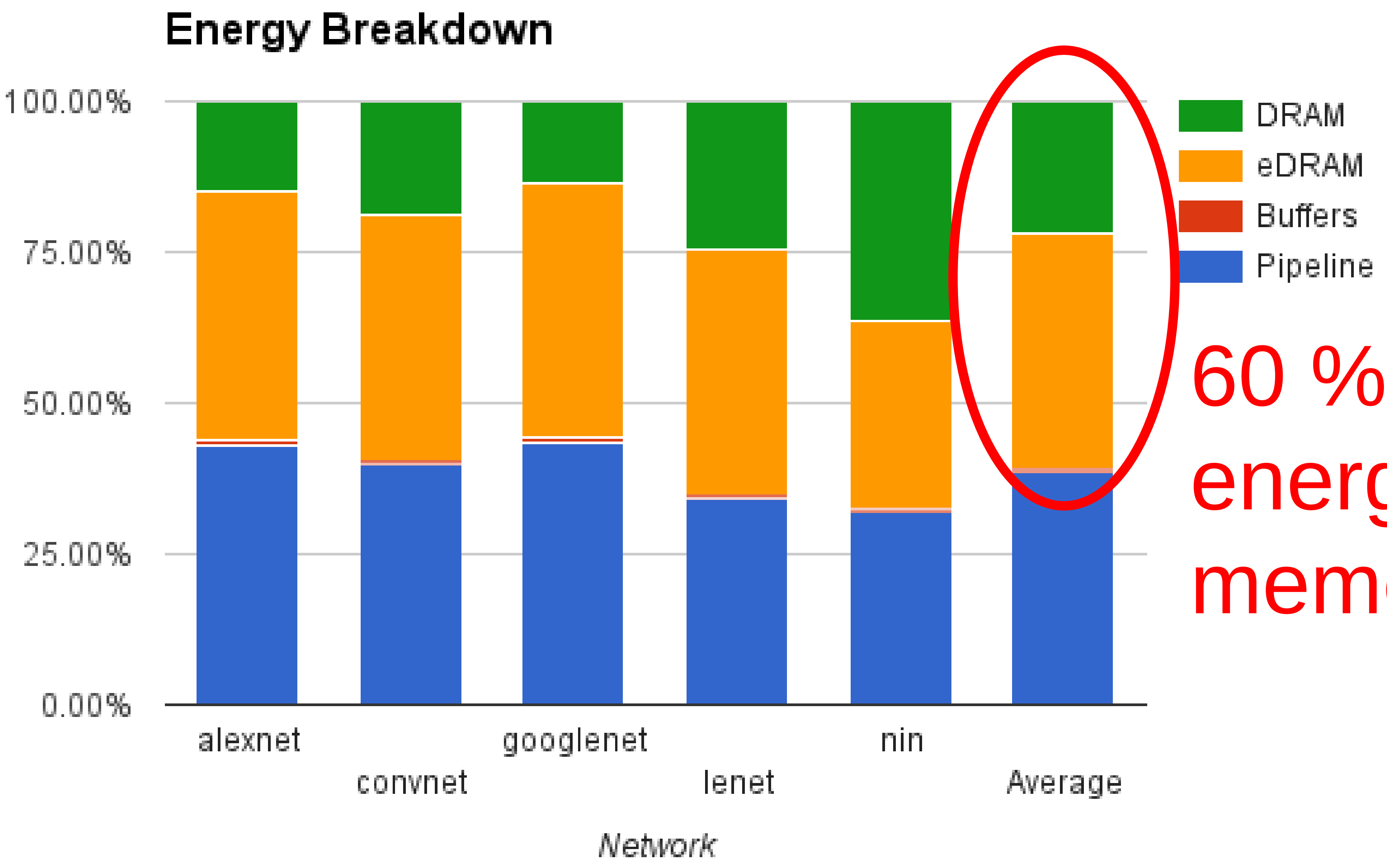
Prior Work

- Minimize precision per layer while maintaining output prediction **accuracy within 1%** vs. a 16 bit baseline

Network	Bits per data element per layer	Traffic Ratio	Bits per weight	Traffic Ratio
LeNet	2,4,3,3	0.16	7	0.44
Convnet	8,7,7,5,5	0.48	9	0.56
AlexNet	10,8,8,8,8,8,6,4	0.56	10	0.63
NiN	10,10,9,12,12,11,11,11,10,10,9	0.64	10	0.63
GoogLeNet	14,10,12,12,12,12,11,11,11,10,9	0.72	9	0.56

Accelerator Energy Breakdown

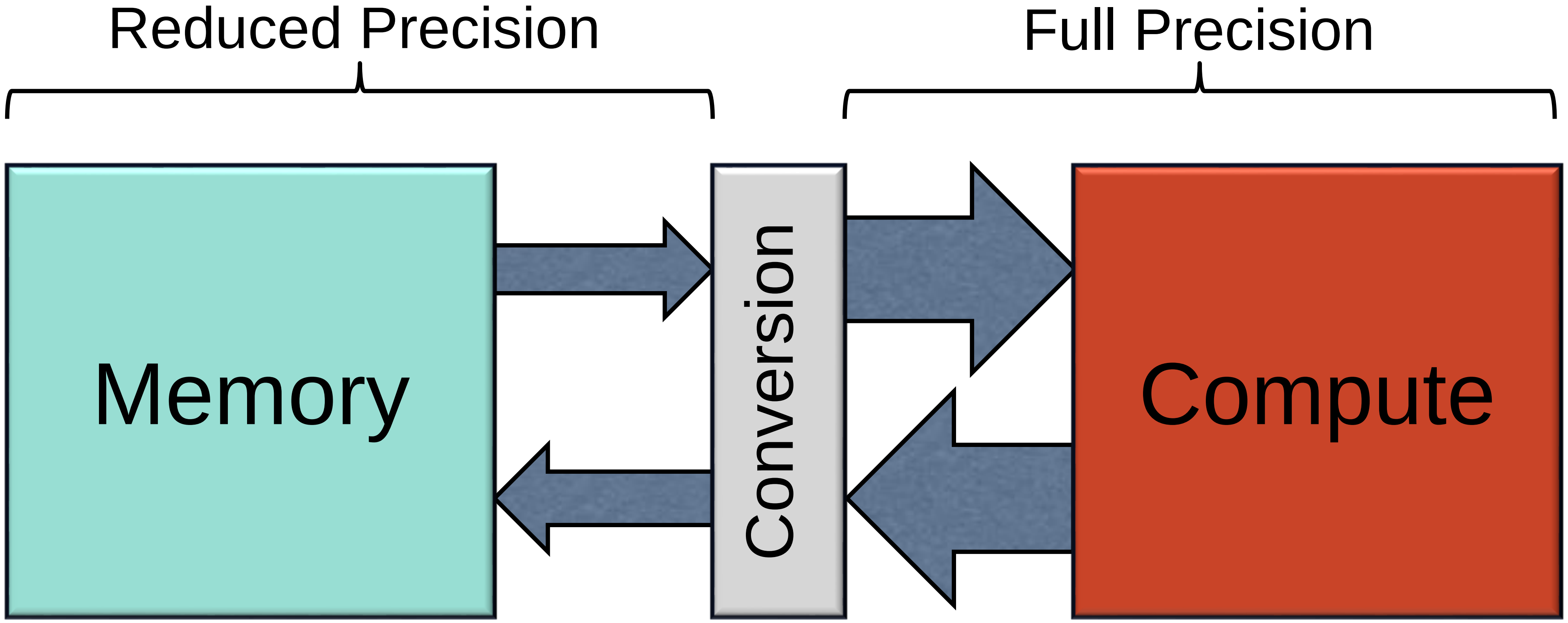
- Model energy of a DNN Accelerator



60 % of energy in memory

Proteus

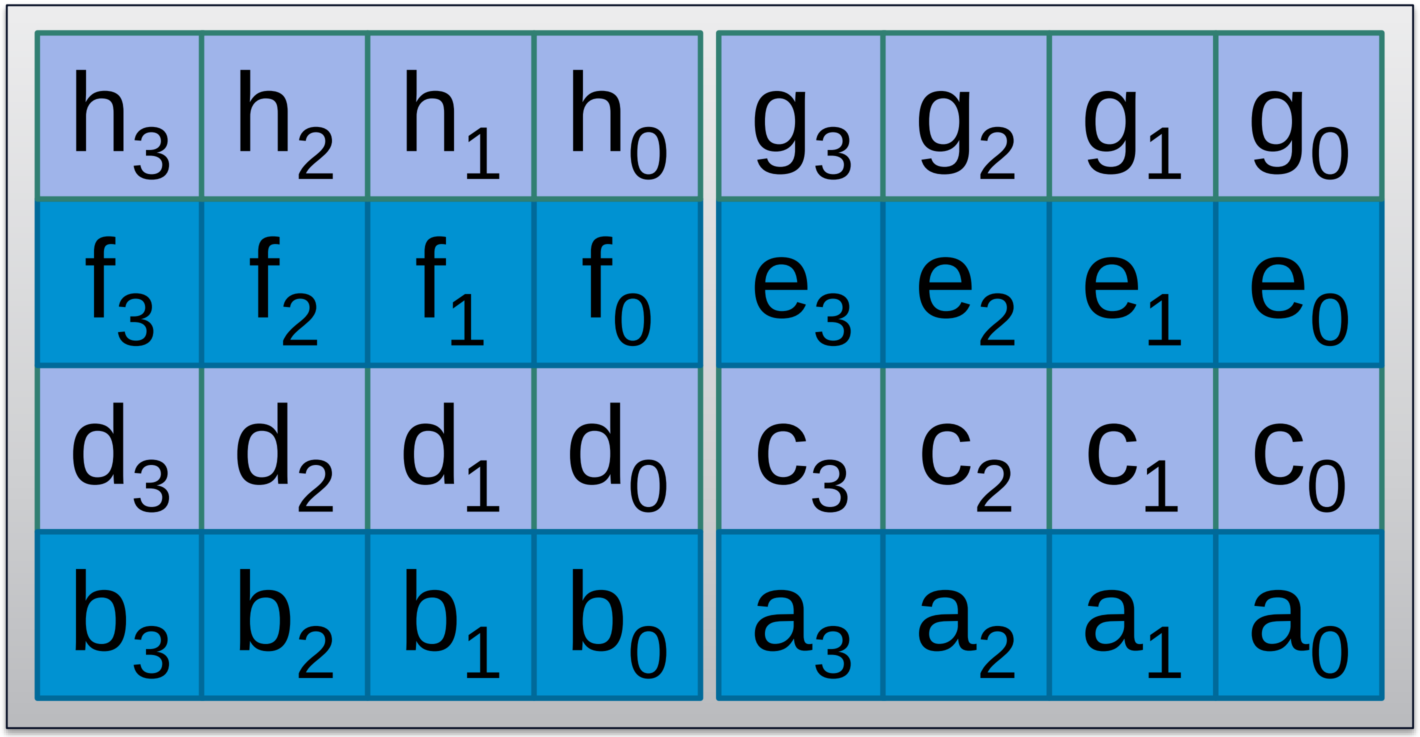
- Dynamically configurable, bit aligned reduced precision hardware memory compression



Baseline Memory

Example

- 4 bit words
- 2 words per row

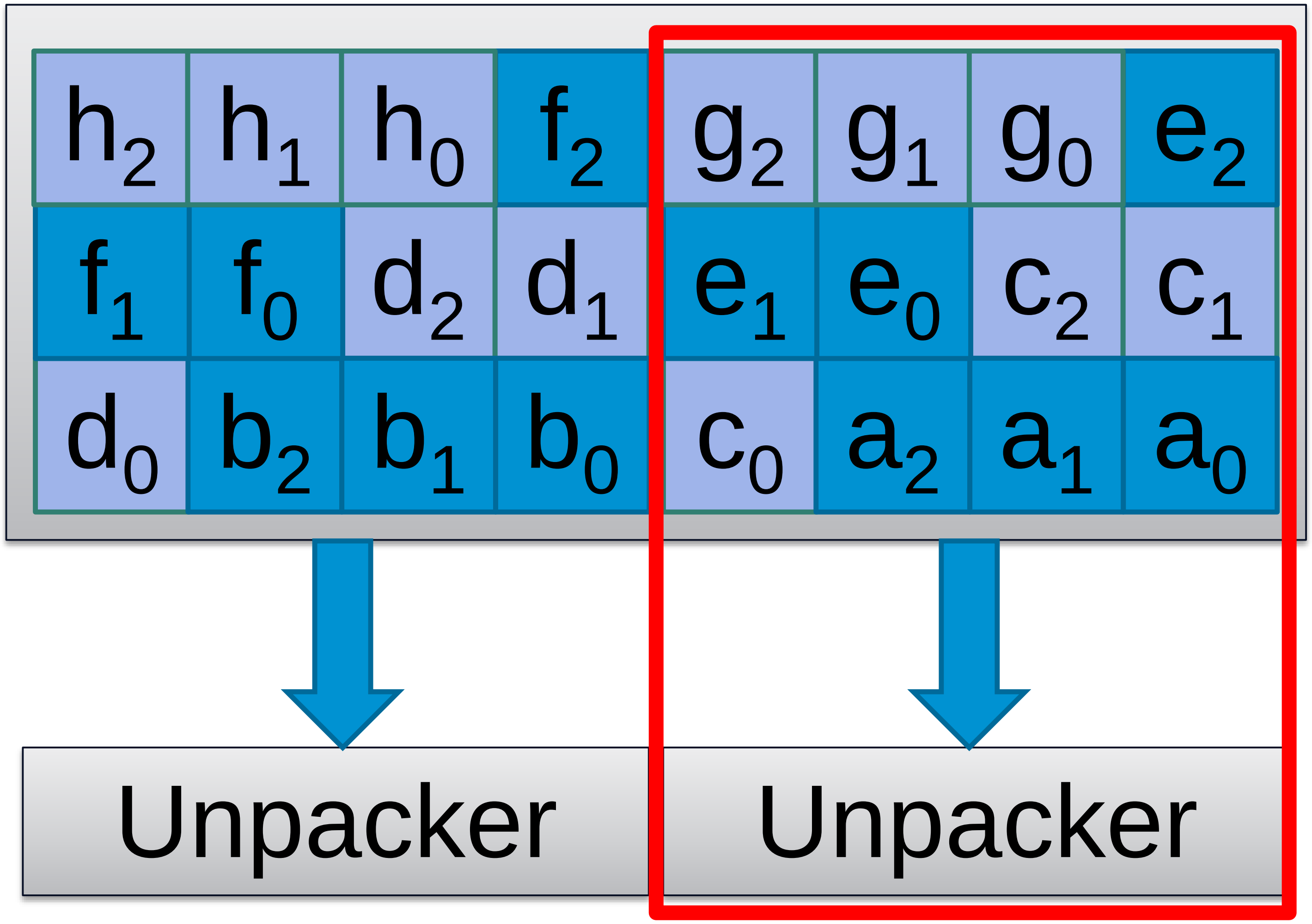


h_3	h_2	h_1	h_0	g_3	g_2	g_1	g_0
f_3	f_2	f_1	f_0	e_3	e_2	e_1	e_0
d_3	d_2	d_1	d_0	c_3	c_2	c_1	c_0
b_3	b_2	b_1	b_0	a_3	a_2	a_1	a_0

Packed Memory

Example

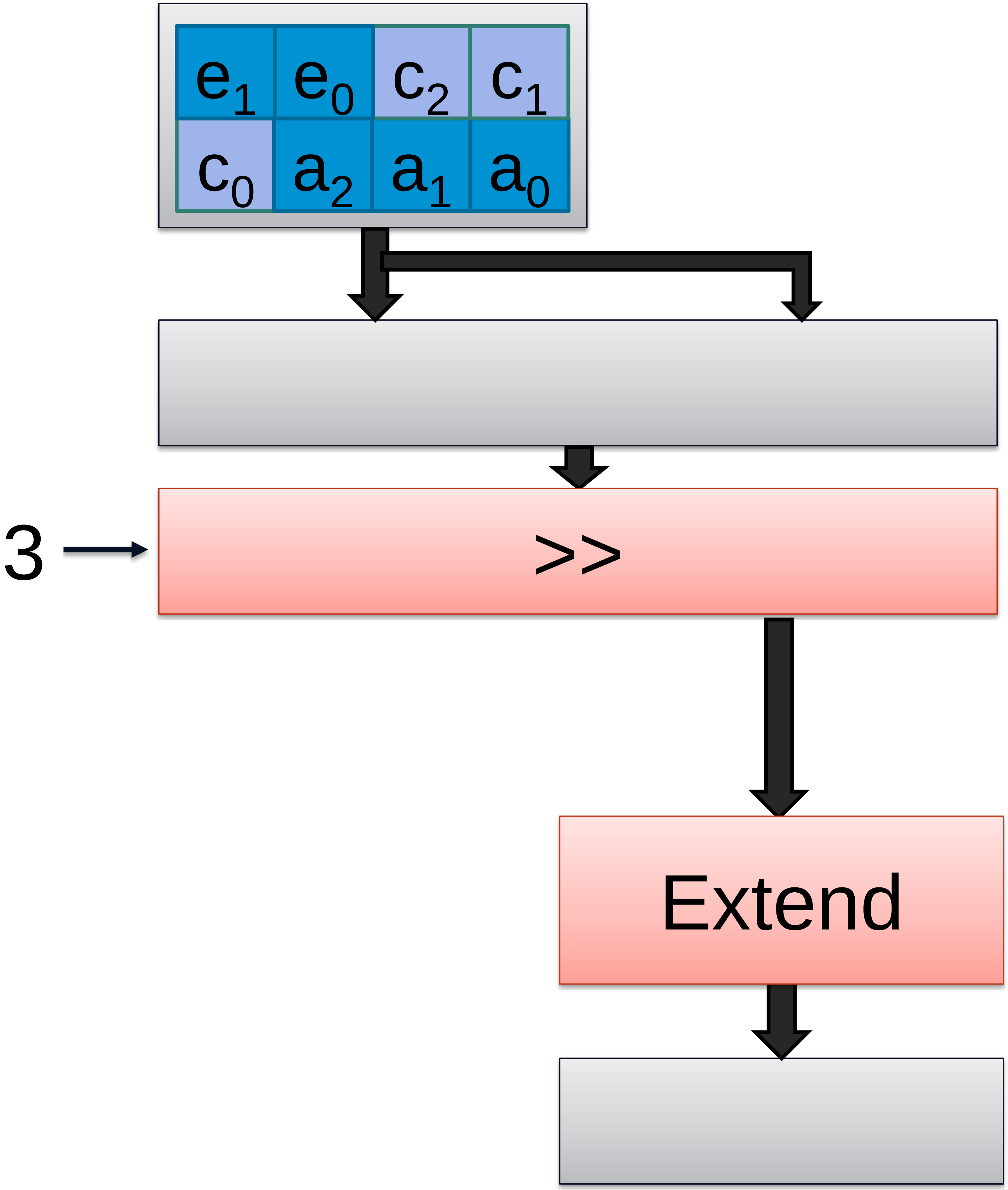
- 4 bit words
- 2 words per row
- 3 bit reduced precision
- Footprint = ¾ baseline
  - *(ideally)*



h_2	h_1	h_0	f_2	g_2	g_1	g_0	e_2
f_1	f_0	d_2	d_1	e_1	e_0	c_2	c_1
d_0	b_2	b_1	b_0	c_0	a_2	a_1	a_0

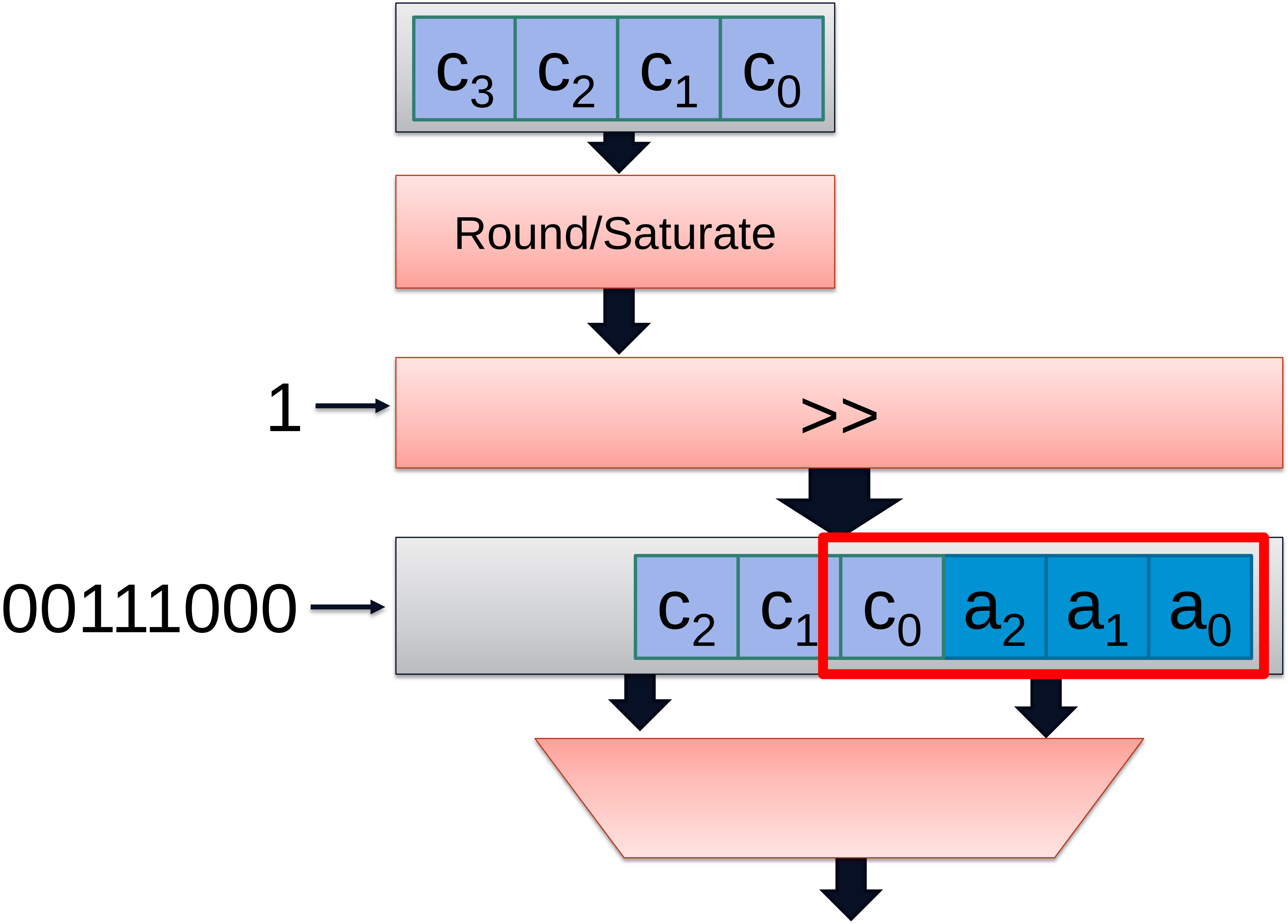
Design for one column of words

Unpacker



e_1	e_0	c_2	c_1
c_0	a_2	a_1	a_0

Packer



Packer / Unpacker

Pros

- Simple design

- Negligible performance impact
  - 2 additional pipeline stages

Cons

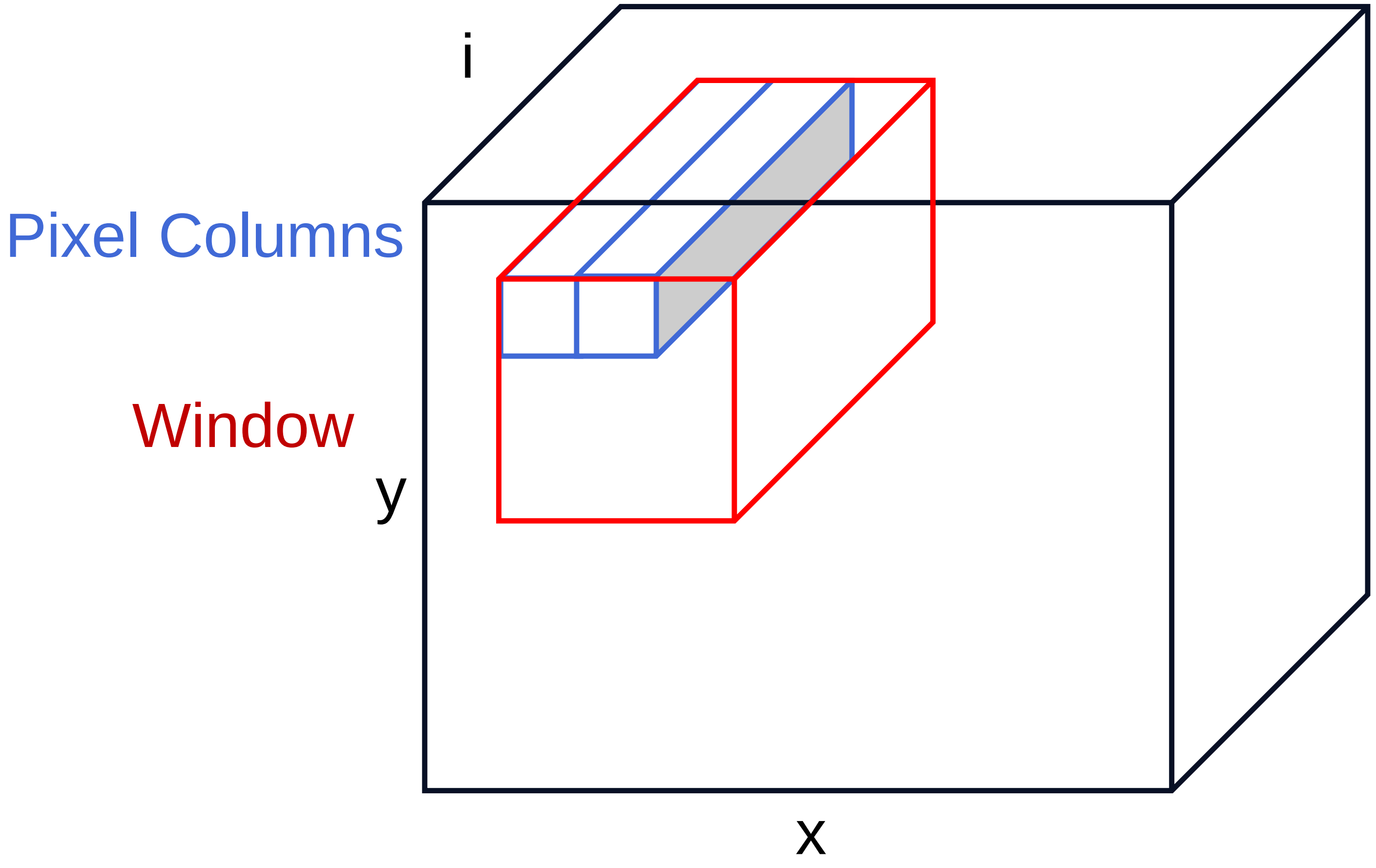
- Forces a **read/write order** on the data
  - Won’t work for certain applications/architectures

- Imposes **alignment constraint** on the data
  - May not get ideal compression

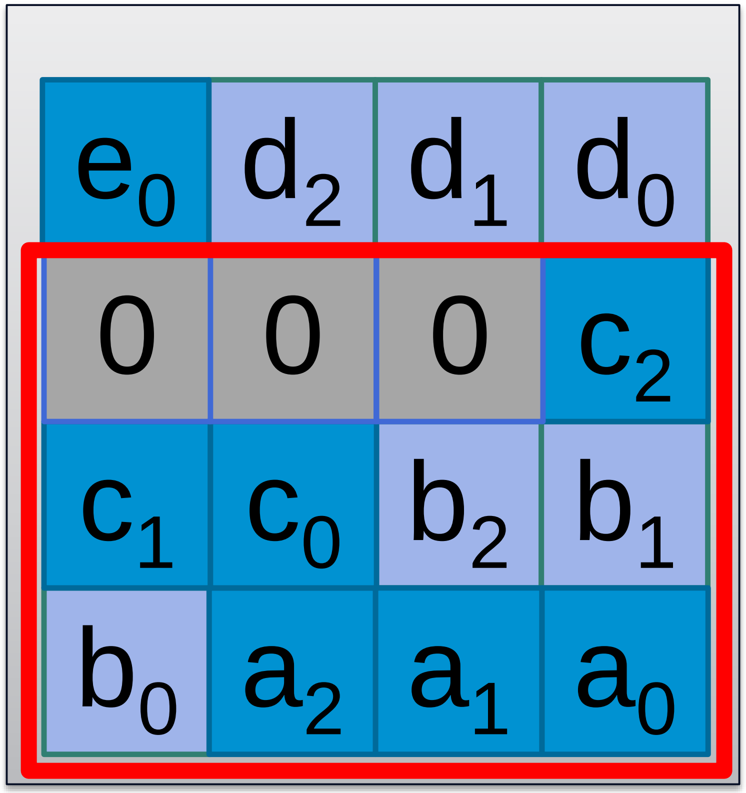
Alignment Constraint

Constraint: must unpack one data element every cycle

Then: first data element in each window must be aligned



e.g. Ni = 3



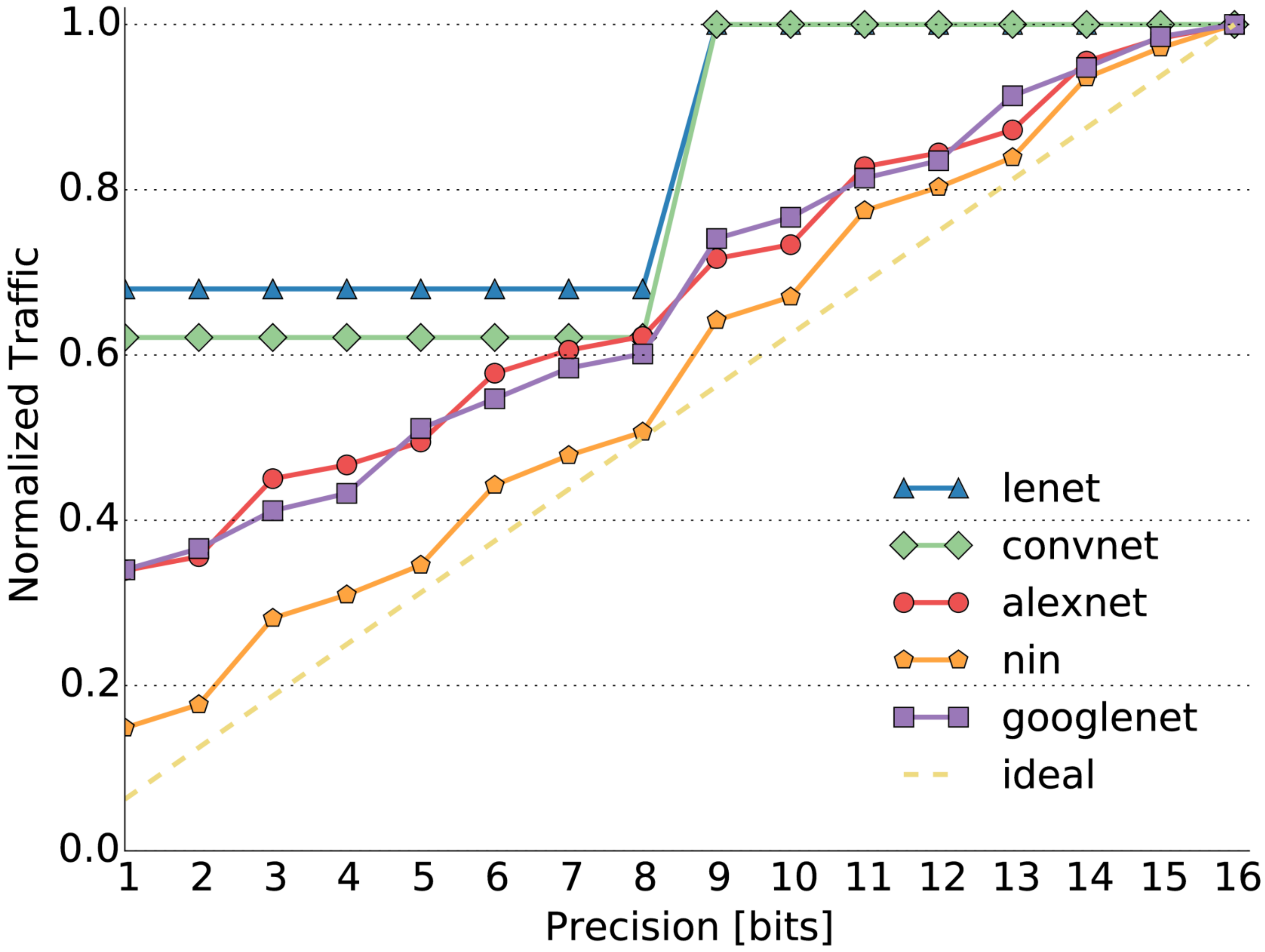
e_0	d_2	d_1	d_0
0	0	0	c_2
c_1	c_0	b_2	b_1
b_0	a_2	a_1	a_0

No compression

Note: Only affects data, not weights

Alignment Penalty

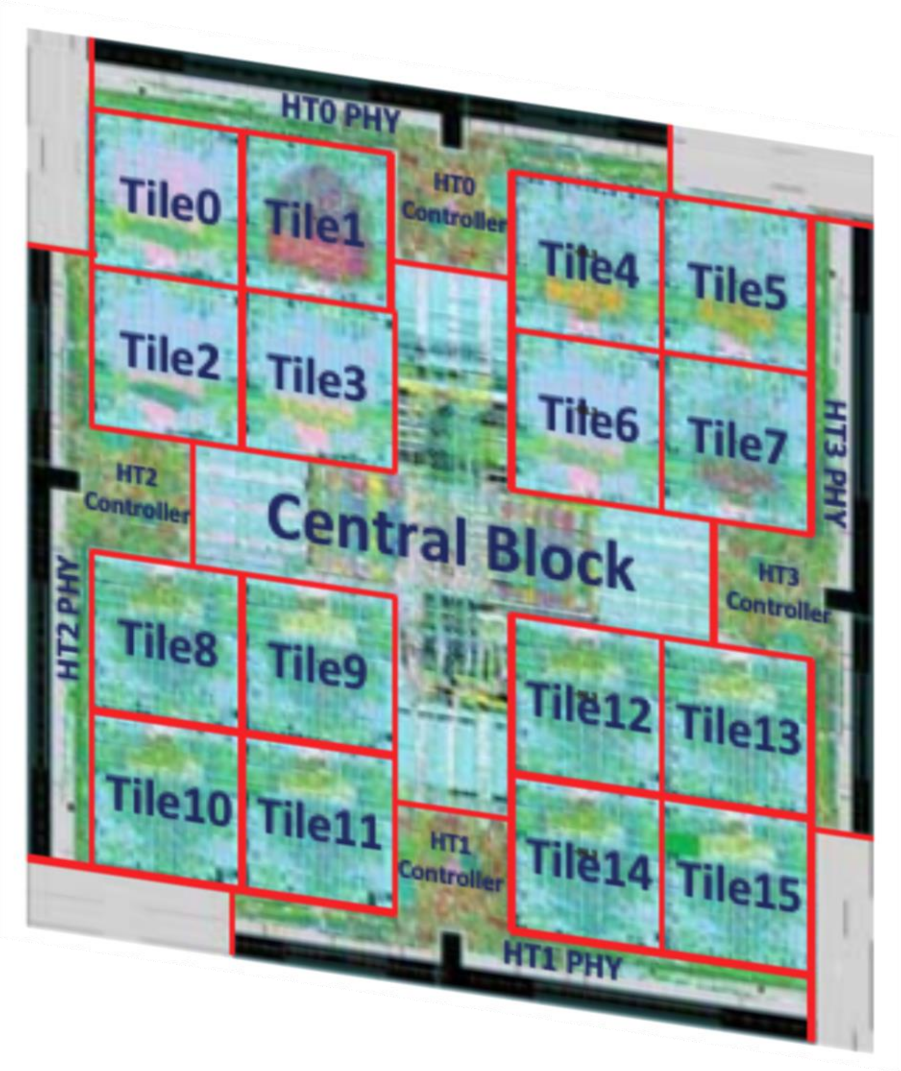
- Alignment yields non-ideal memory traffic scaling



Accelerator

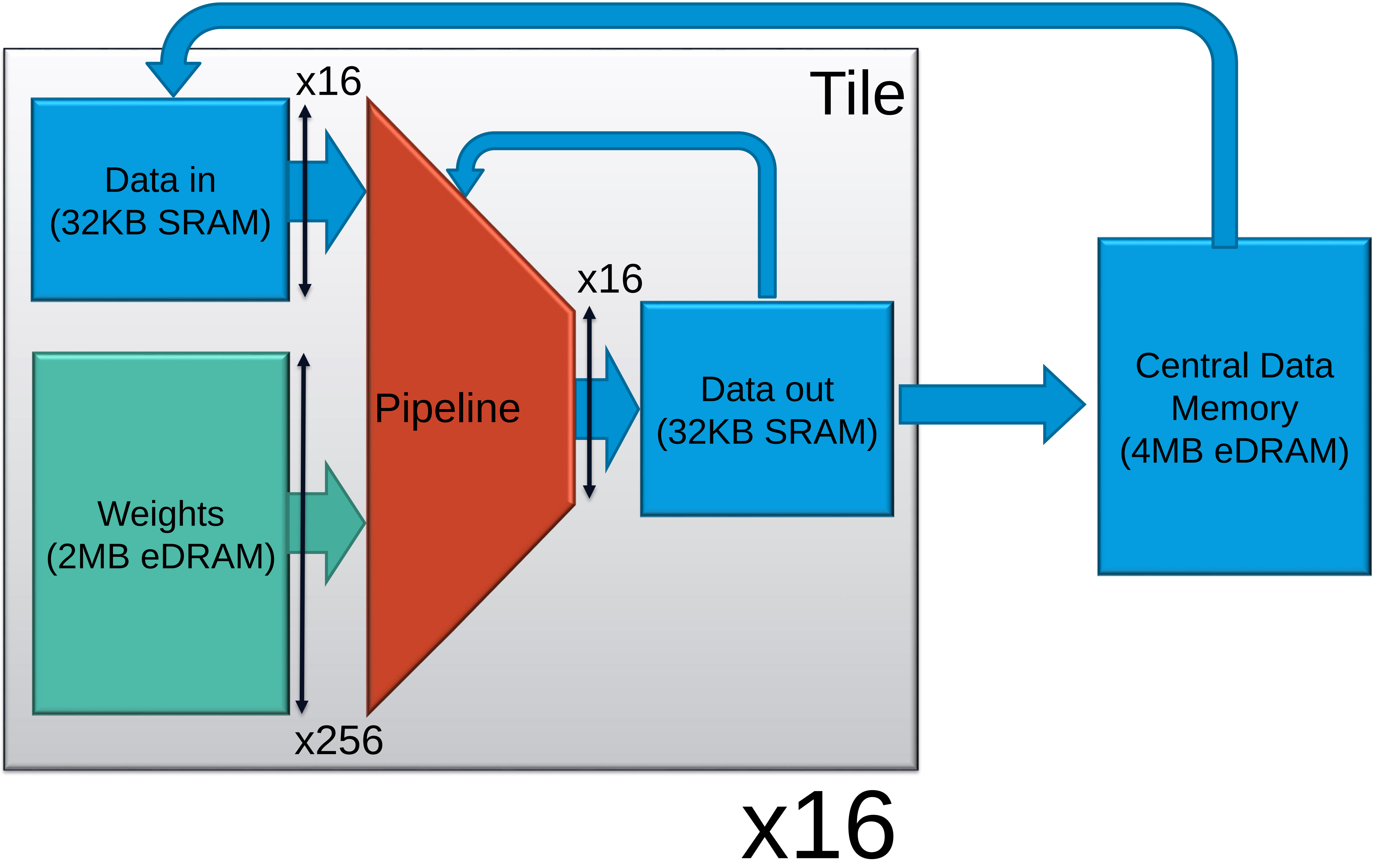
DaDianNao [4]

- State of the art neural network accelerator
- **16 bit fixed-point** computation
- **16 Tiles** with compute and local memory
- **36 MB of on chip eDRAM**
- Area: 68 mm^2
- Power: 16 W
- Frequency: 606 MHz

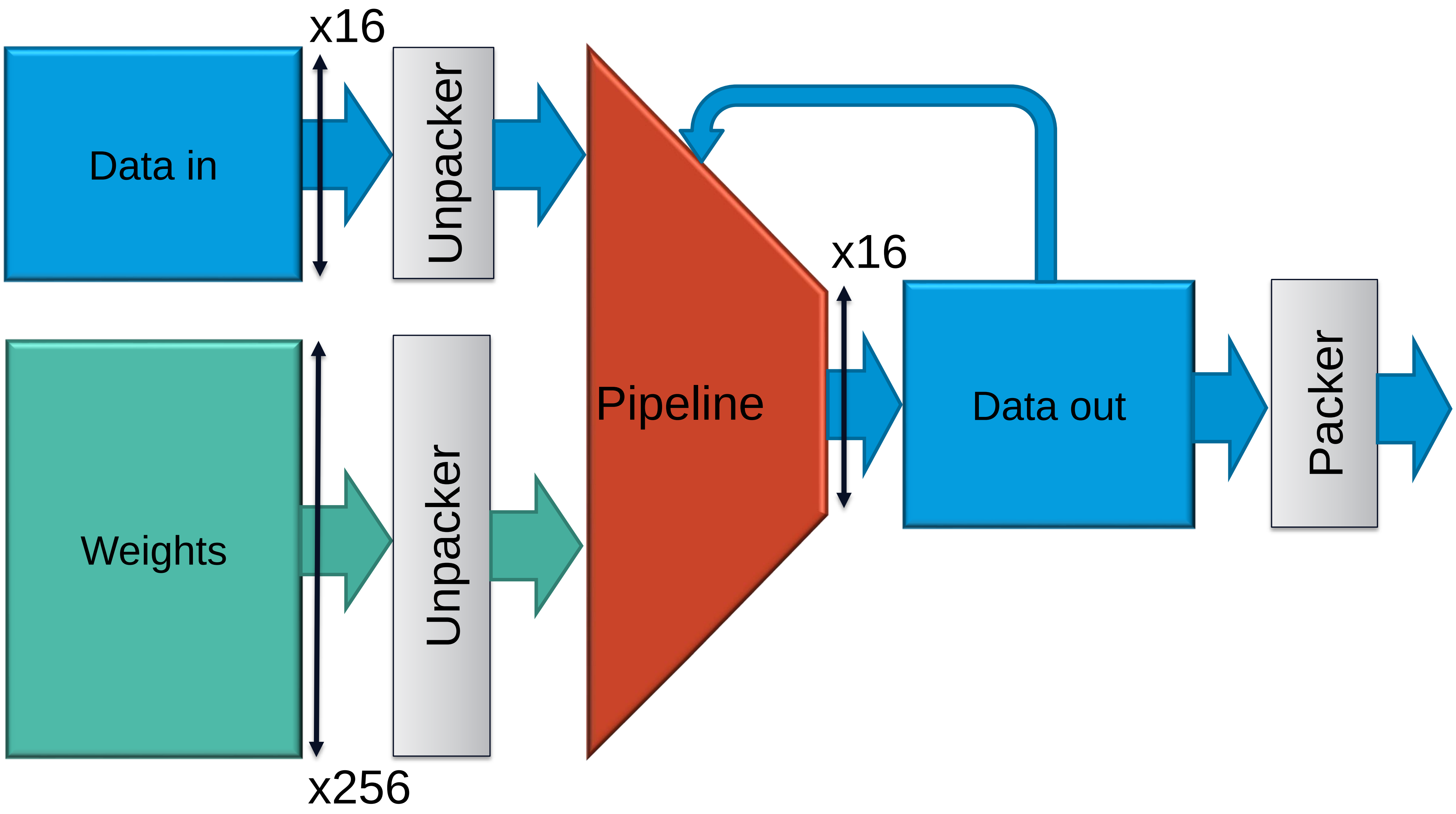


[4] Chen et. Al, "DaDianNao: A Machine-Learning Supercomputer," MICRO 2014

DaDianNao Tile



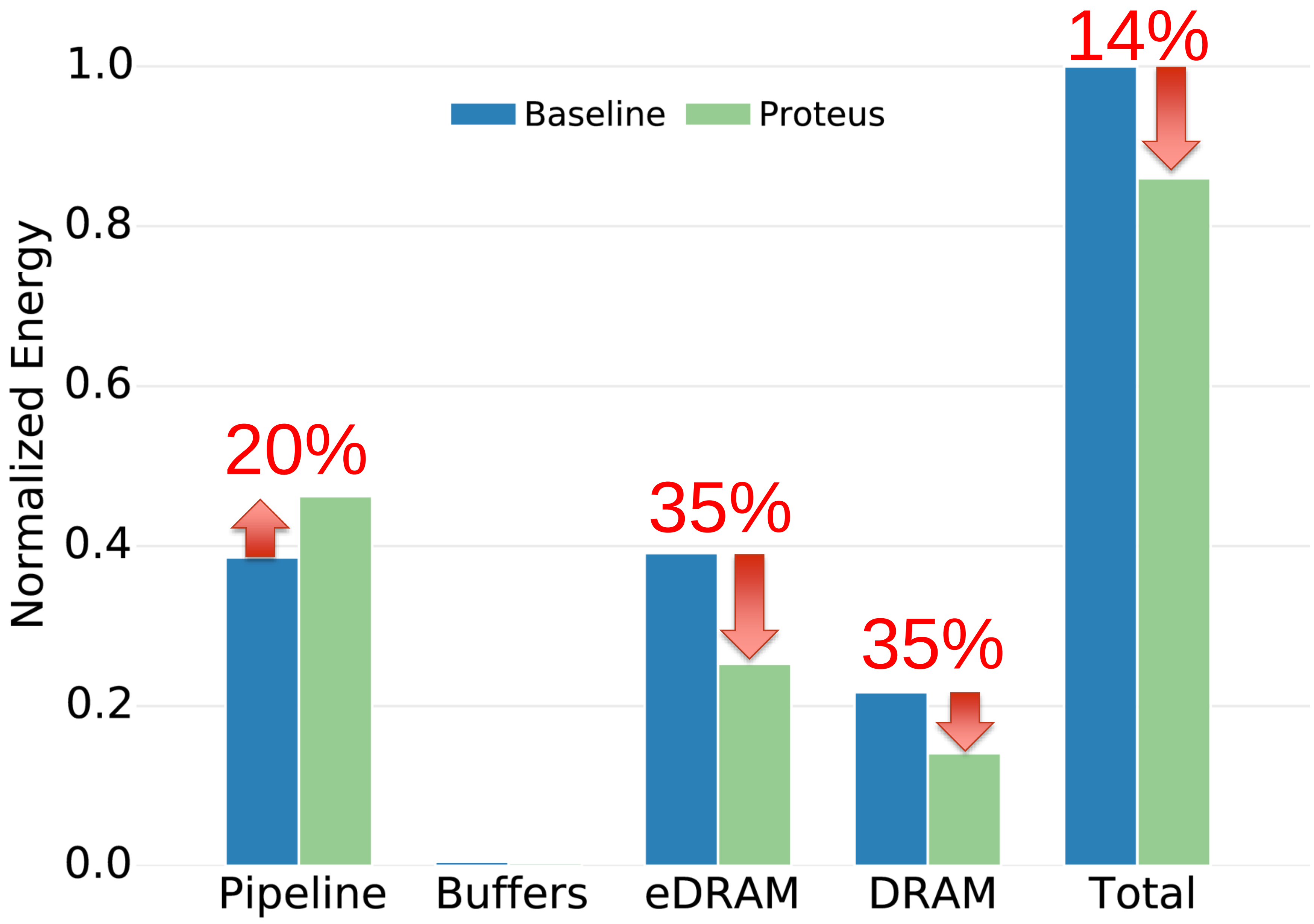
Proteus on DaDianNao



Methodology

- Baseline: DaDianNao - 16 bit fixed-point storage + compute
  - + 2GB DDR3 for storing network weights
- Logic : pipeline, packers and unpackers
  - Synthesized with Synopsis Design Compiler with the 45nm FreePDK library
- Energy models for memory:
  - SRAM buffers: CACTI v5.3
  - eDRAM: Destiny modeling tool
  - DRAM: Dramsim2

Energy Savings



Conclusion

- We leverage the reduced precision tolerance of DNNs to enable dynamically configurable, bit aligned memory compression

- Integrate a simple packer/unpacker design into a state of the art neural network accelerator

- Reduce energy by 14% without impacting speed with at most 1% loss of accuracy

Future work

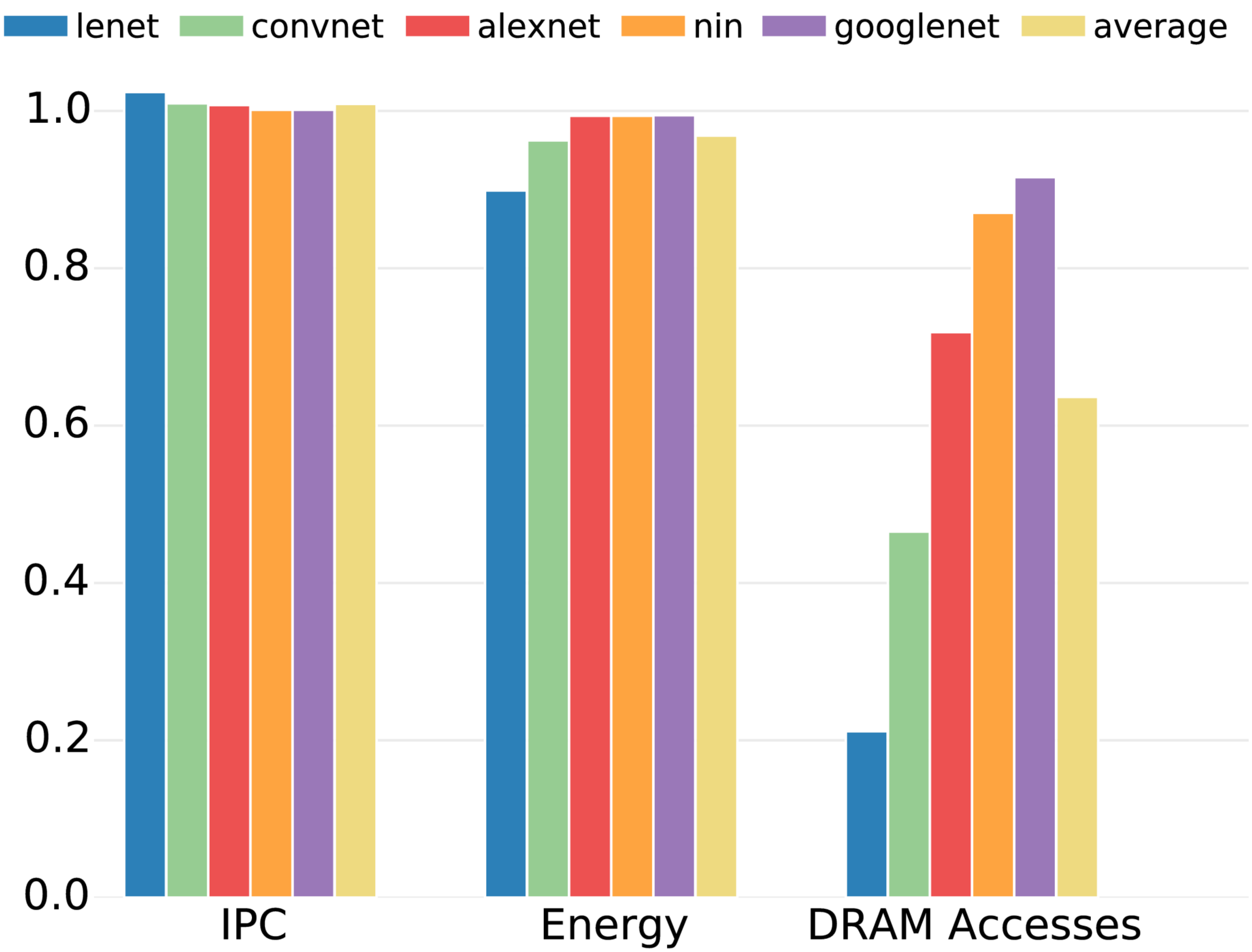
- Static Energy
  - Turn off memory banks due to reduced footprint
- Improve DaDianNao Energy Model
  - Add power models for interconnect and off chip communication
- Reduced precision compute

Thanks!

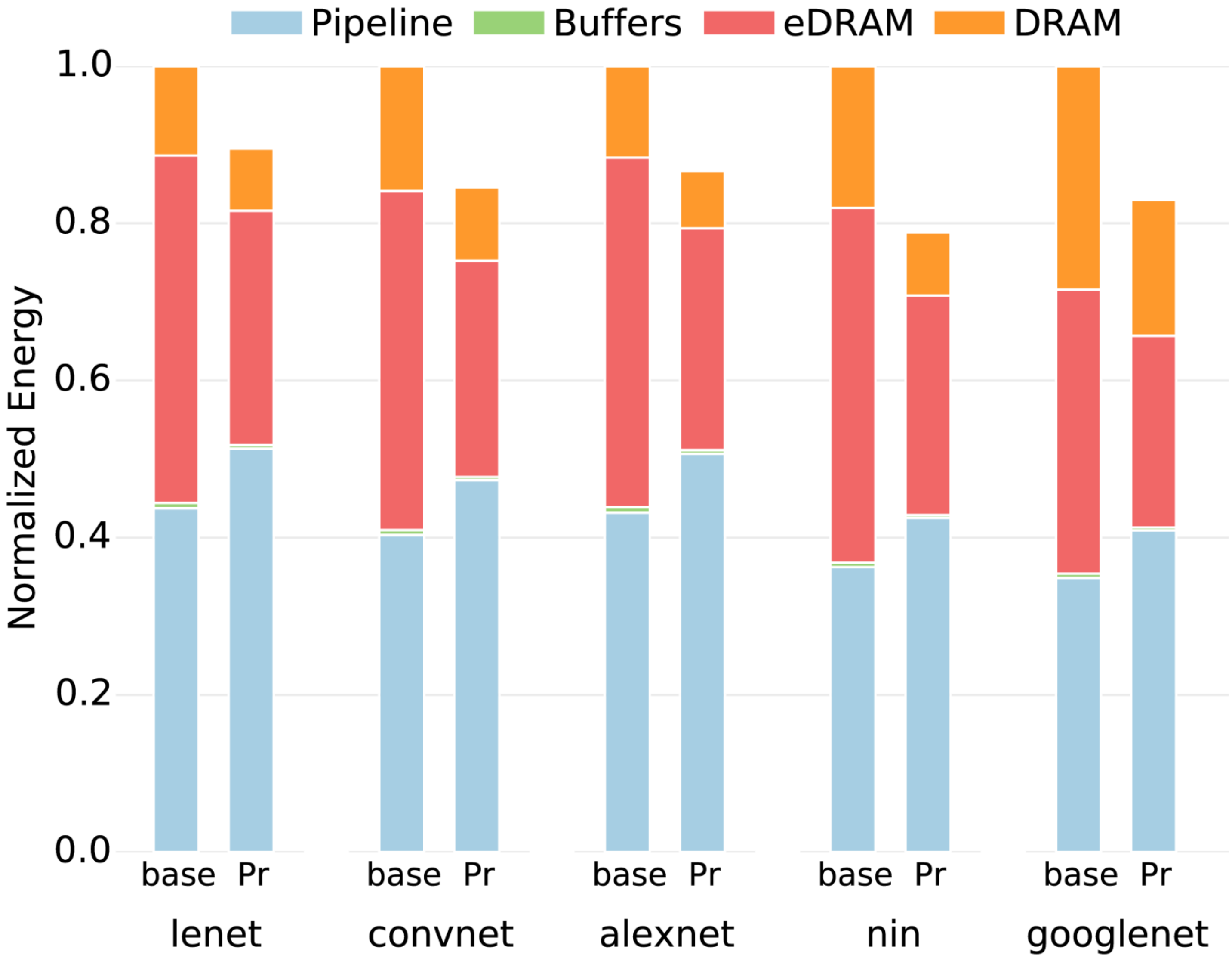
Questions?

Email: Patrick.judd@mail.utoronto.ca

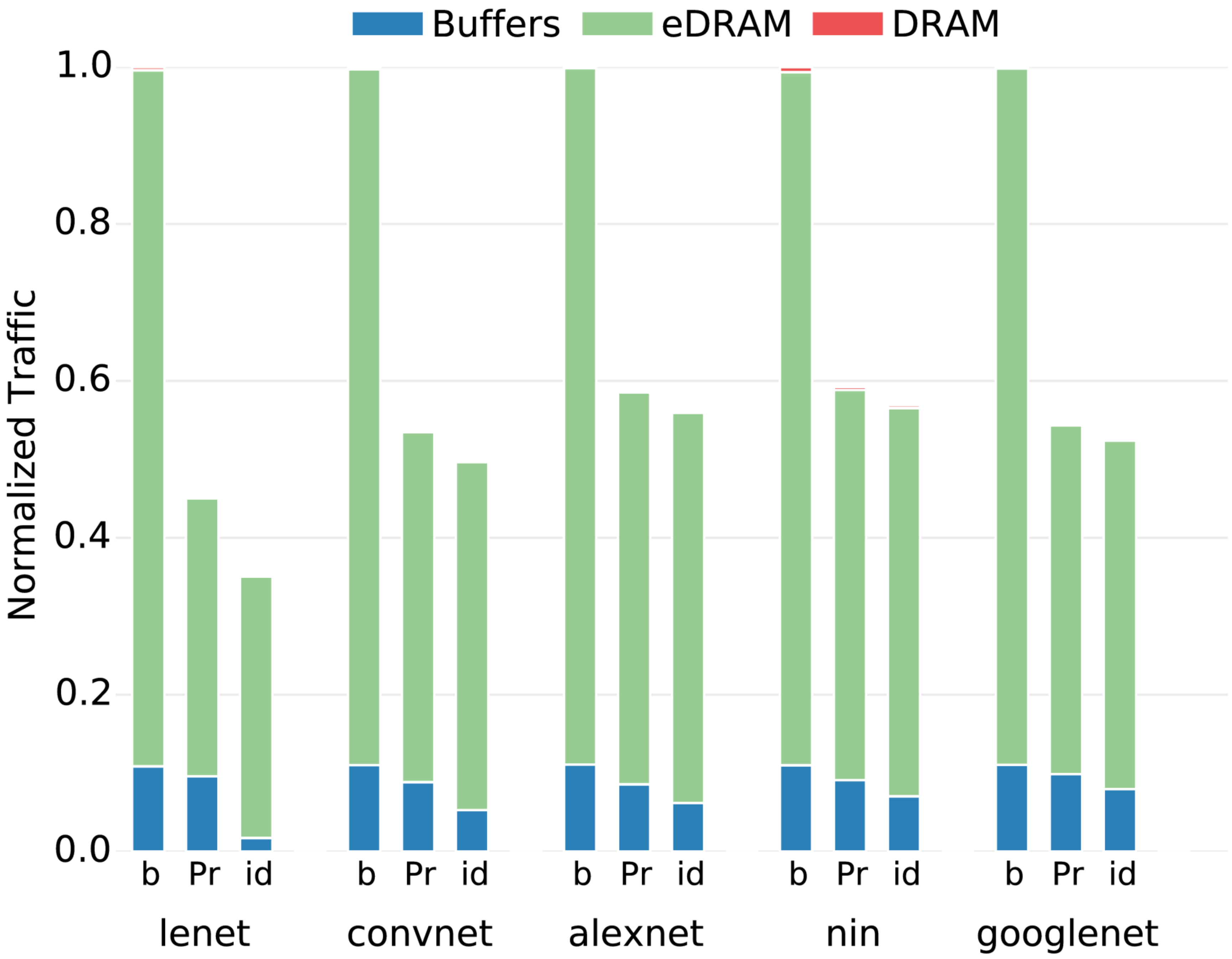
GPU Evaluation



Energy Savings per Network

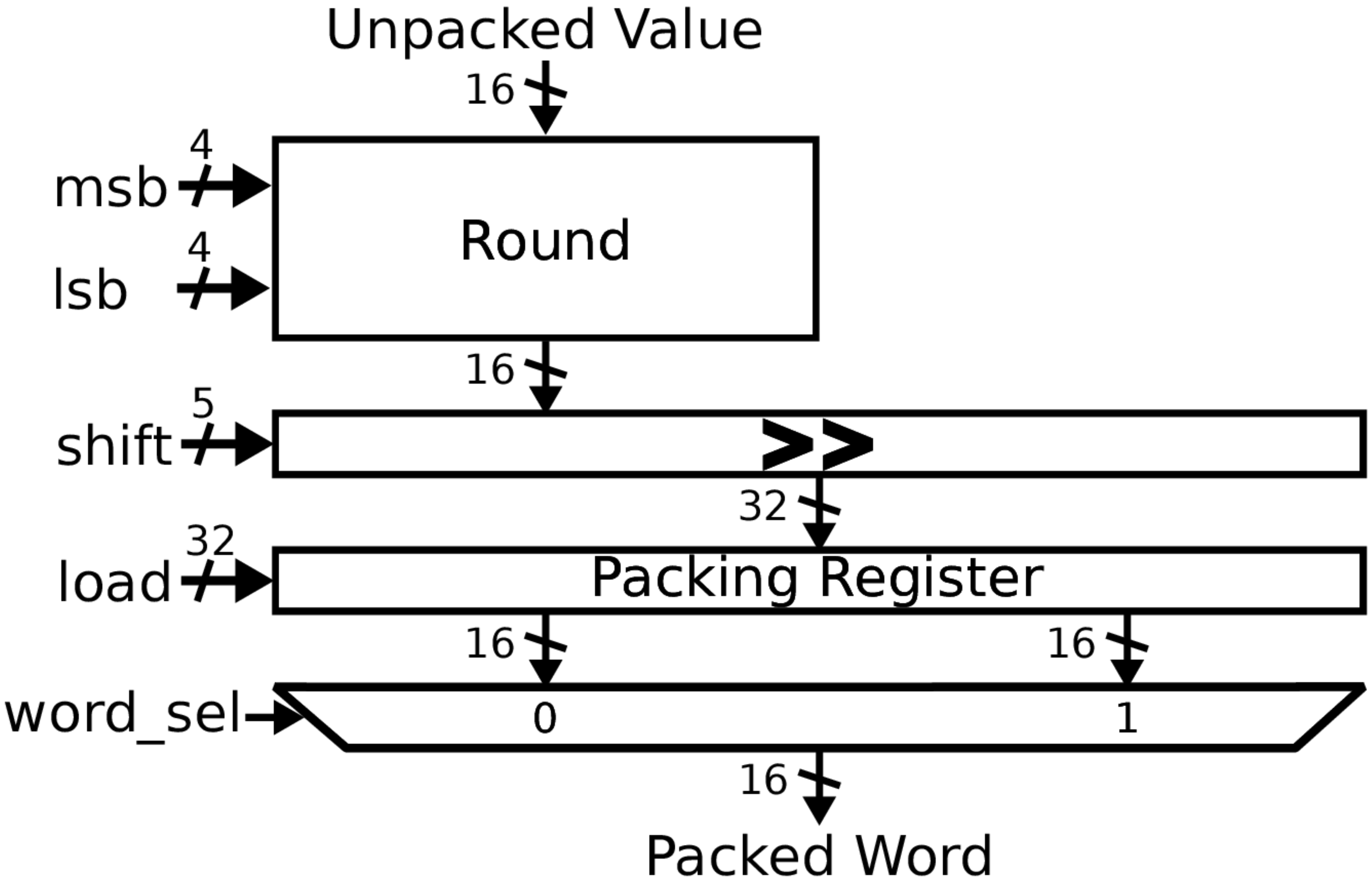


Traffic Breakdown

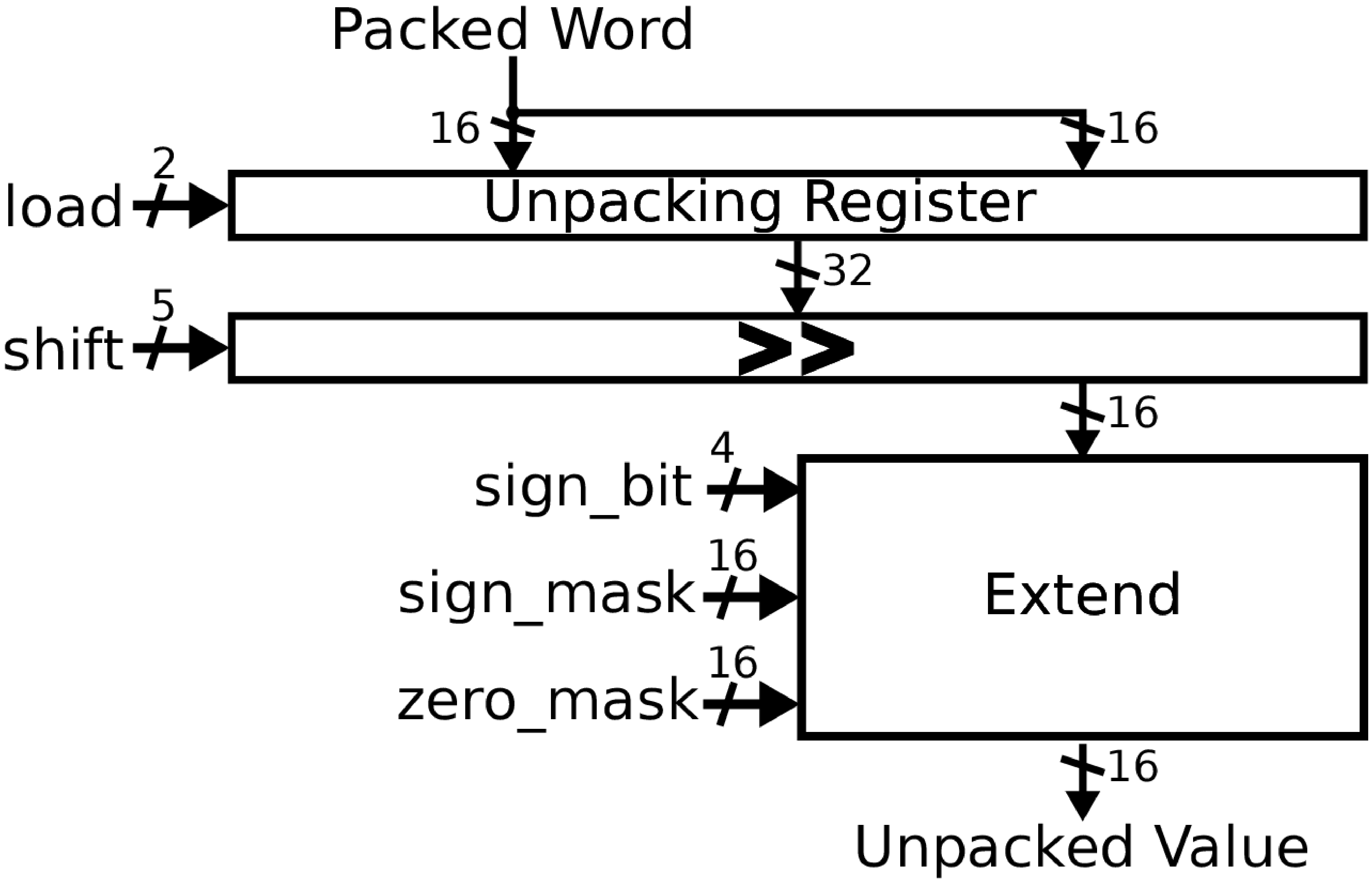


Packer

- Packer is essentially the reverse set of operations
- The unpacked value (full precision) needs to be rounded and potentially saturated to produce the closest reduced precision value



Unpacker


Packed Word
16
16
2
load
Unpacking Register
32
5
shift
>>
16
4
sign_bit
16
sign_mask
16
zero_mask
Extend
16
Unpacked Value

3-D Convolution