Tensilica Day 2019

Exploration of Memory Energy-Reduction Strategies for a Deep Learning ASIP

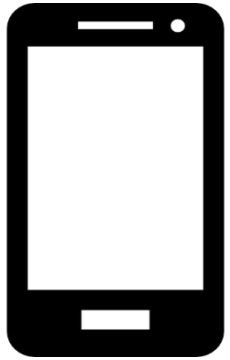
Lennart Reimann, 23.09.2019

Institut für Kommunikationstechnologien und eingebettete Systeme
RWTH AACHEN UNIVERSITY

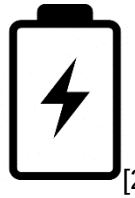
Motivation

- **Deep Learning:**
  - **Image Recognition: Convolutional Neural Networks**
  - **High amount of Data: 5.4 Mbit for the input feature maps in AlexNet**

  ➢ **Computationally intensive + external memories required**


[1]

- **Problem: Energy-constrained mobile embedded systems**


[2]

- **Solution: Reduction of the energy consumption of the external memory**
  - **Strategy: Reduction of the memory transactions by reducing the amount of data**

Compressing the data before transmitting

Outline

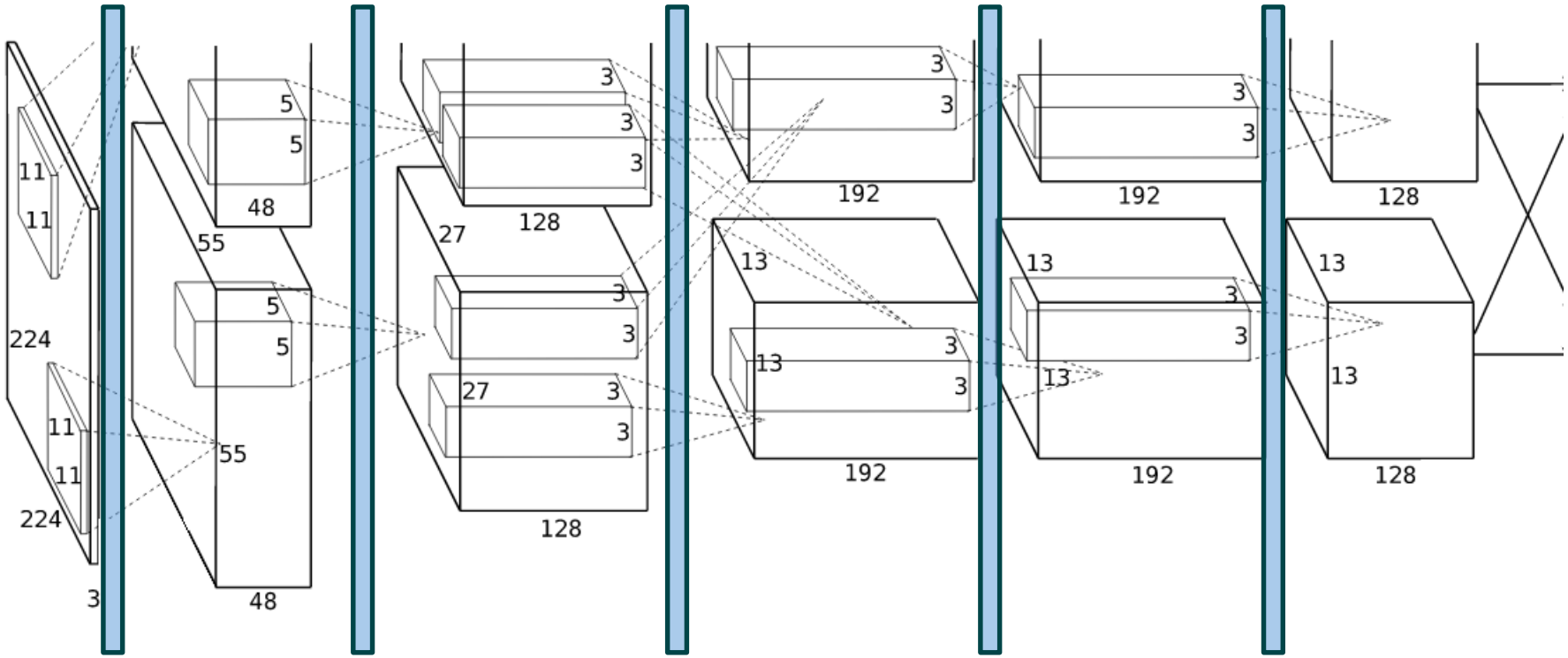
1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

AlexNet Structure

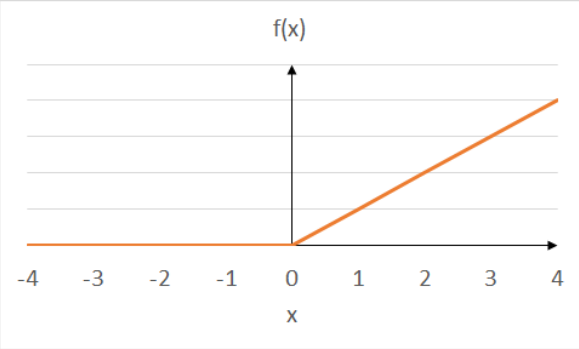
- 5 convolutional layers
- External memory loads/stores row by row:
  - Loads: Input feature maps and filter parameters
  - Stores: Output feature maps



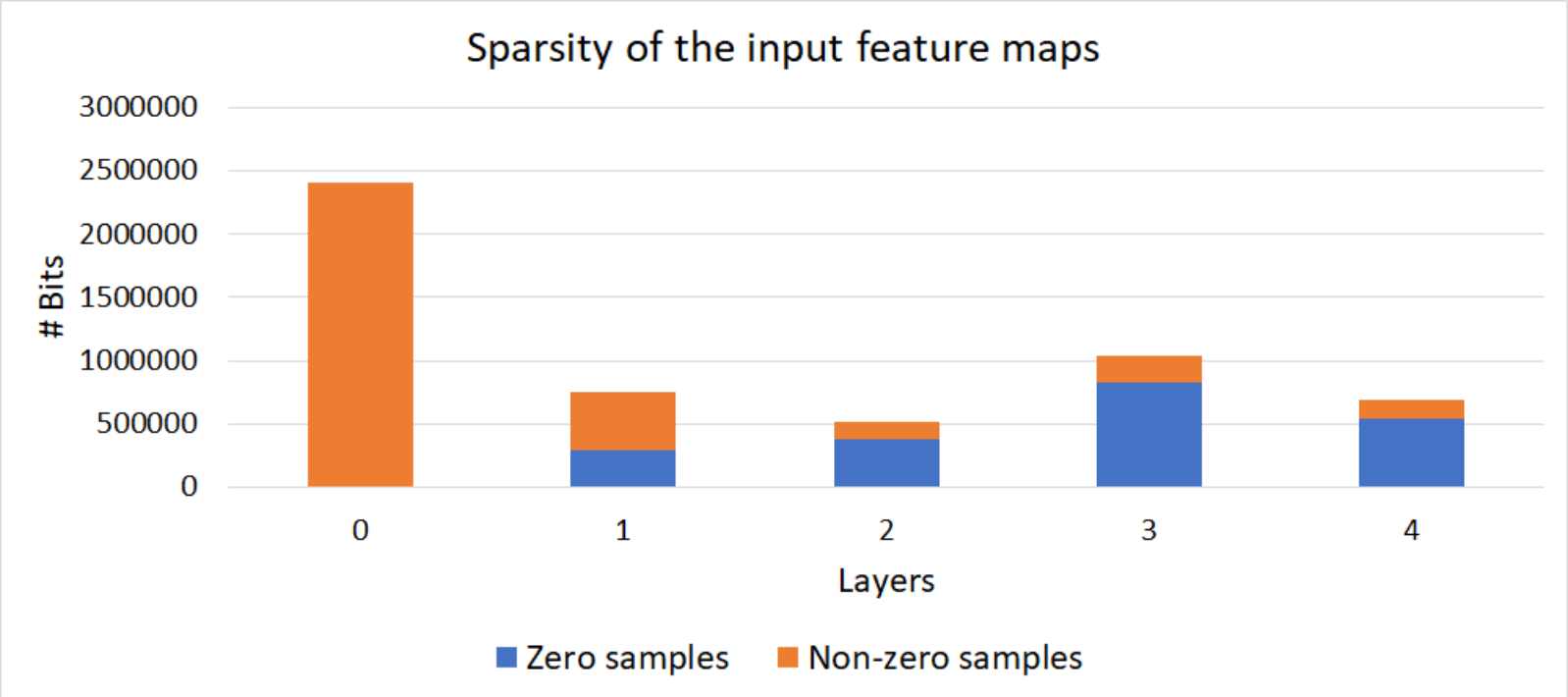
AlexNet structure

Sparsity of the Input Feature Maps (AlexNet)

- Non-linear function used at the end of each convolutional layer: ReLU (f(x) = max(x,0))



- Sparsity increases with the layers
- Highest number of input feature map bits: Layer 0



Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

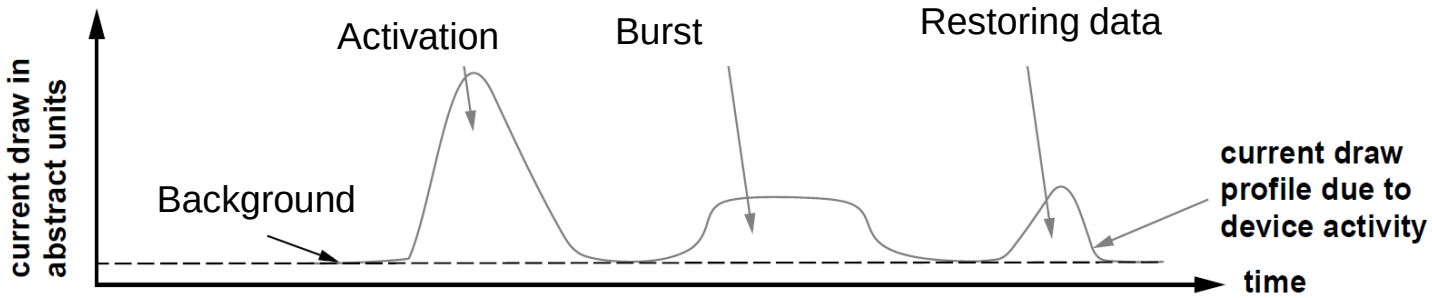
DDR3 SDRAM Memory

- When is energy consumed?
  - Refresh, background, activation, bursts

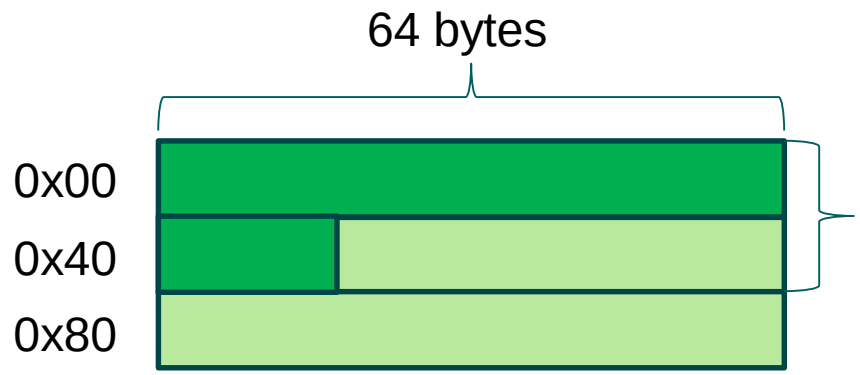
- Simulation environment (DRAMSim2)
  - Trace based simulation (using list of transactions)
  - Memory has to be chosen first
  - Memory controller settings have to be set
  - DDR3 SDRAM was chosen

- Per access 64 bytes are written

- So if data is reduced to 65 bytes
  - ➡ Still two bursts needed



Current Draw [4]



Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

Huffman Coding

- Simple: Maps the most probable symbols on a lower amount of bits
- Block size determines the number bits are grouped to form a symbol
- Example (Block size: 2):

Symbol	Probability
a („b‘00“)	0.4
b („b‘01“)	0.3
c („b‘10“)	0.2
d („b‘11“)	0.1

Huffman-Tree

Symbol	Codeword
a („b‘00“)	1
b („b‘01“)	01
c („b‘10“)	001
d („b‘11“)	000

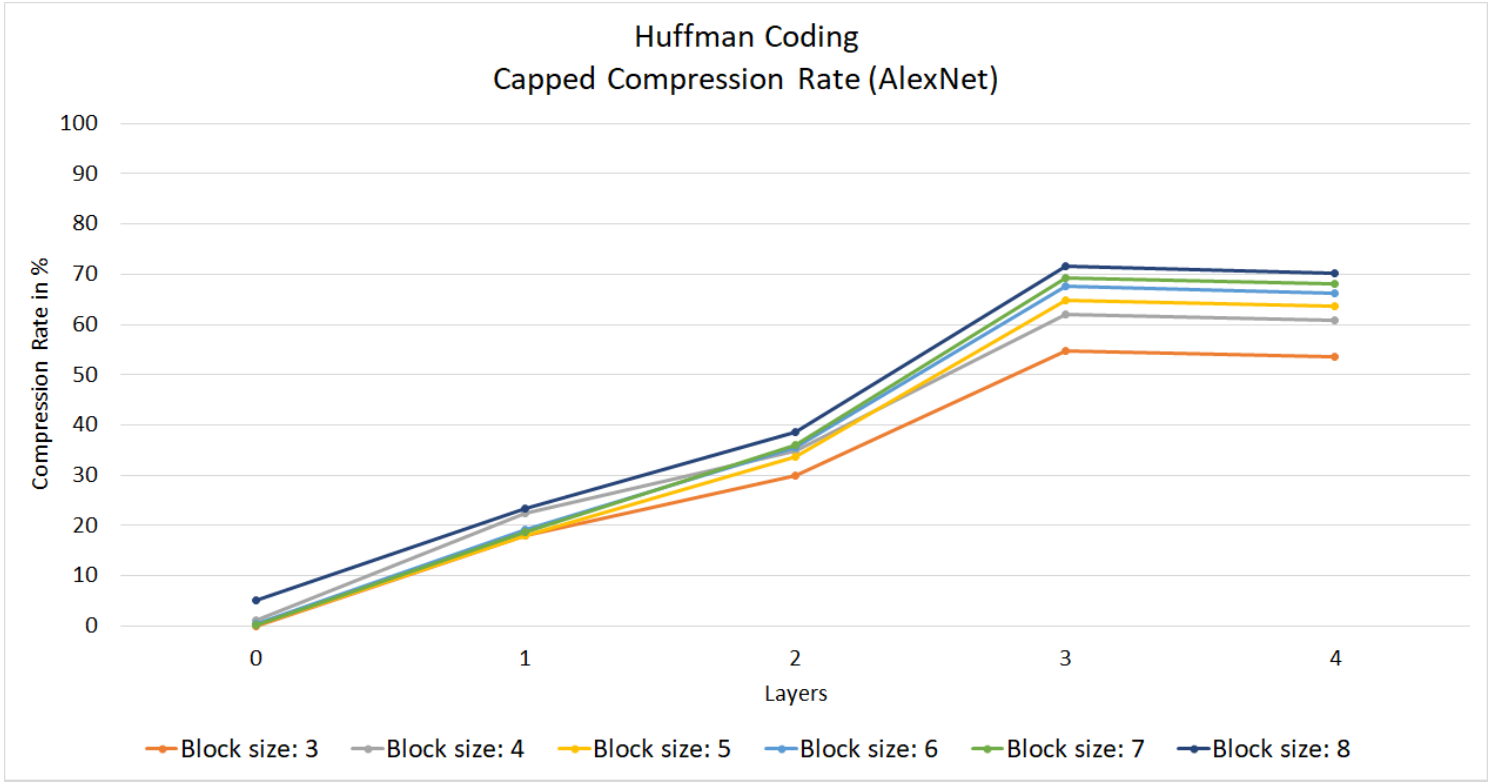
Ex. 1: Input:	b‘00000000	Encoded Output: b‘1111	Length Input: 8 bits	Length Output: 4 bits
Ex. 2: Input:	b‘00101110	Encoded Output: b‘1001000001	Length Input: 8 bits	Length Output: 10 bits

Huffman Coding Average Compression Rate for AlexNet (Capped)

- Compression Rate (only IF-Maps):

- Assigning incompressible lines a rate of 0% (capped)
- Compression rate increases with bigger block sizes

Block size	Aver. Comp. Rate
3	23%
4	26.6%
5	26.4%
6	27.7%
7	28.1%
8	32.2%



Wordwise Run-Length (RL) Compression

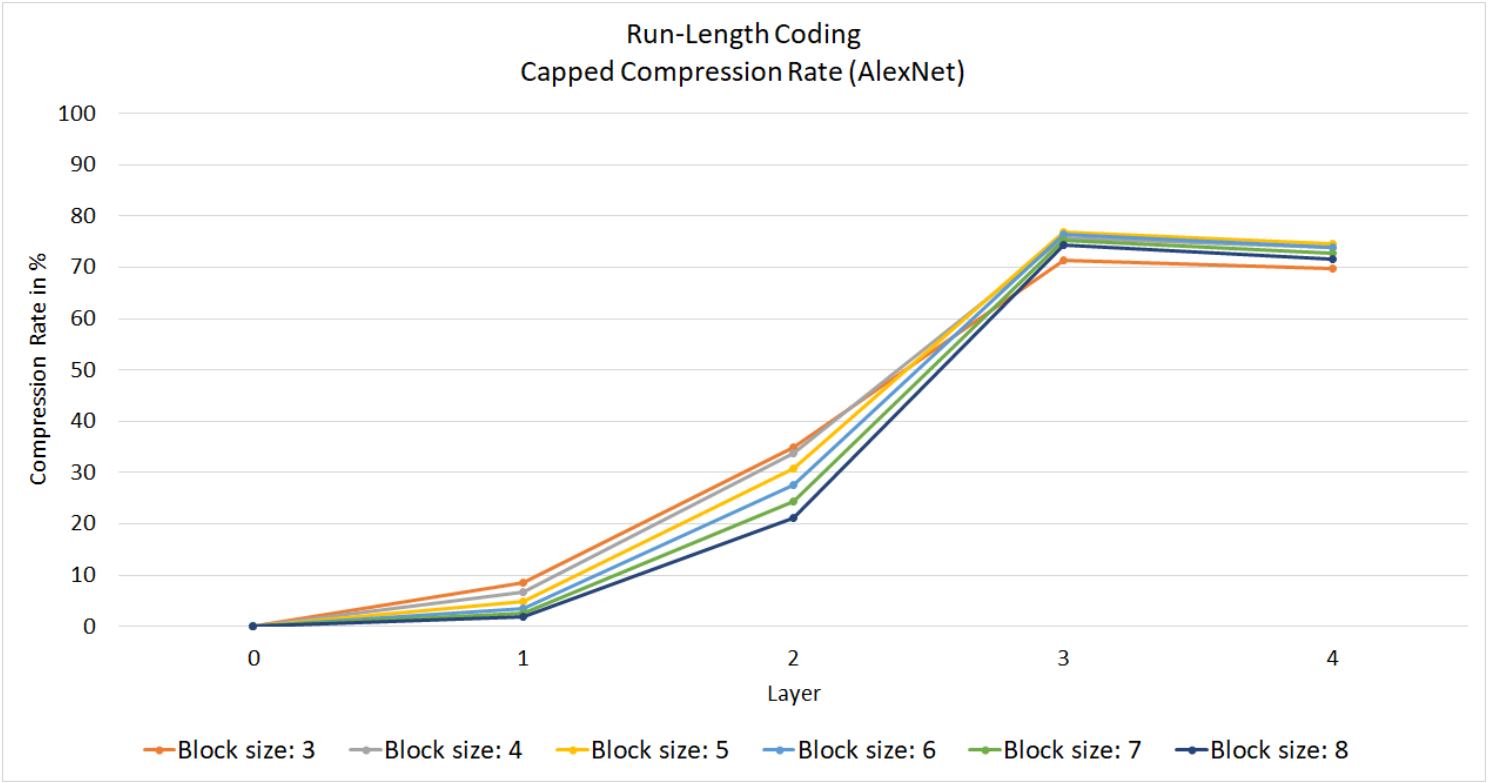
- Describe a number of successive zero samples with its run length
- Copy non-zero samples (levels) to the output
- A level is followed by a run length and vice versa
- The block size determines the number of bits to encode the run length
- Example (Block size: 4):

Input:	**b‘0000000000000000**	**b‘1001011000001101**	**b‘0000000000000000**	**b‘0000000000000000**
RL + Levels:	1	38413	2	
Output:	b‘0001	b‘1001011000001101	b‘0010	

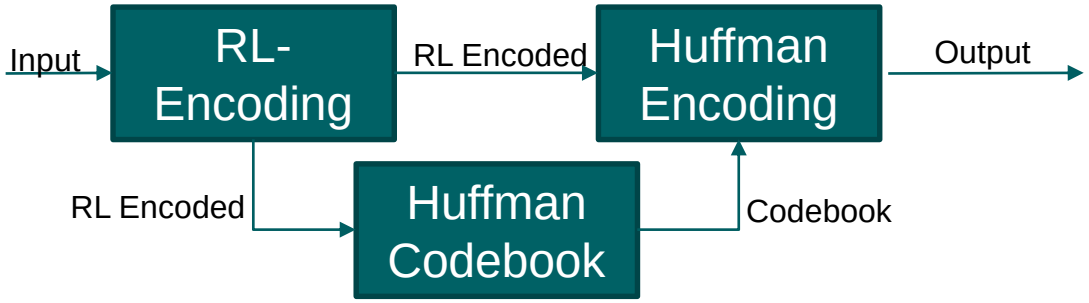
Run-Length Coding Average Compression Rate for AlexNet (Capped)

- Smaller block sizes dominate the first layers (less overhead)
- Bigger block sizes dominate the last layers (longer run lengths)

Block size	Aver. Comp. Rate
3	27.2%
4	28.2%
5	28%
6	27.3%
7	26.5%
8	25.8%

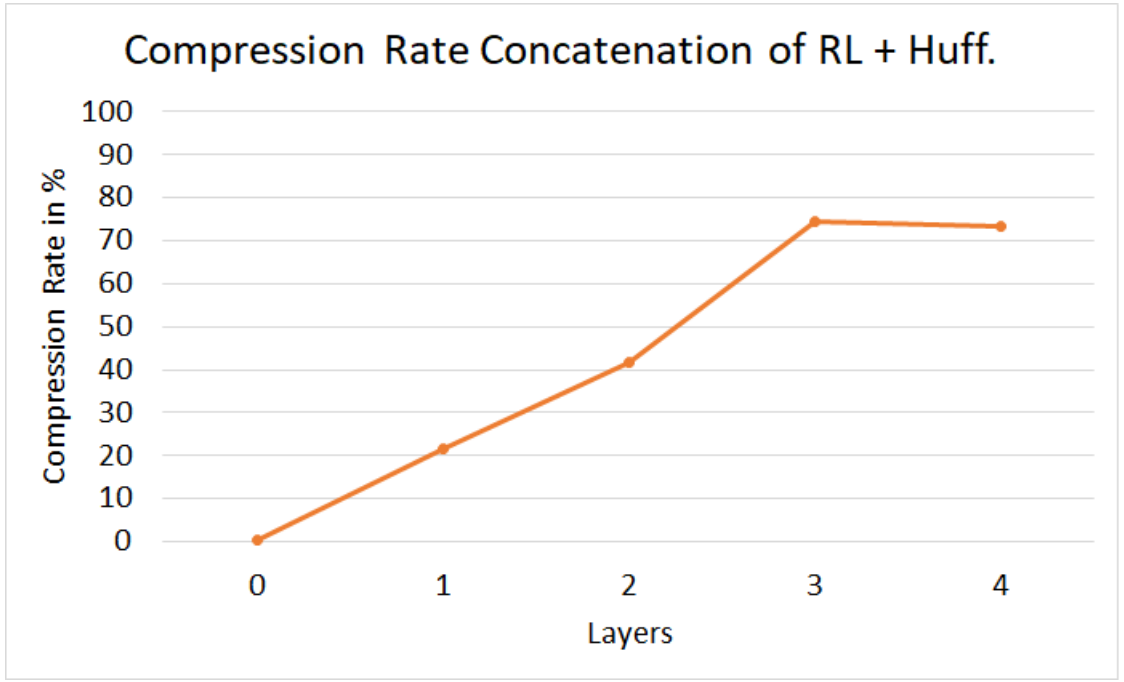


Concatenation of both algorithms (AlexNet)



- Concatenate RL and Huffman Coding (Block size: 4 bit)
- Probabilities of symbols are more distributed

Algorithm	Aver. Comp. Rate
Huff. (BS: 4)	26.6%
RL (BS: 4)	28.2%
RL + Huff	31.4%

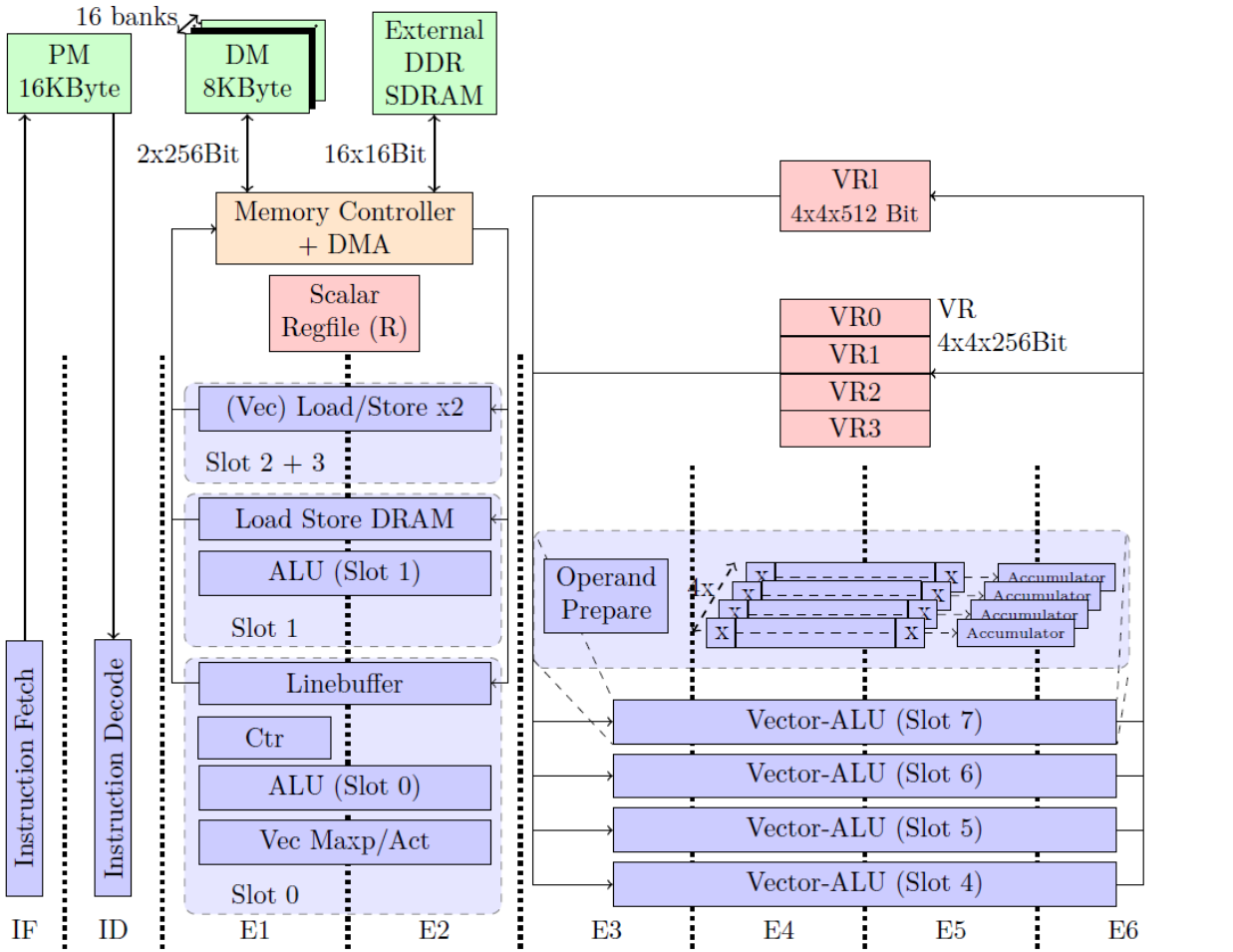


Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

Base Architecture: CNN ASIP (ConvAix)

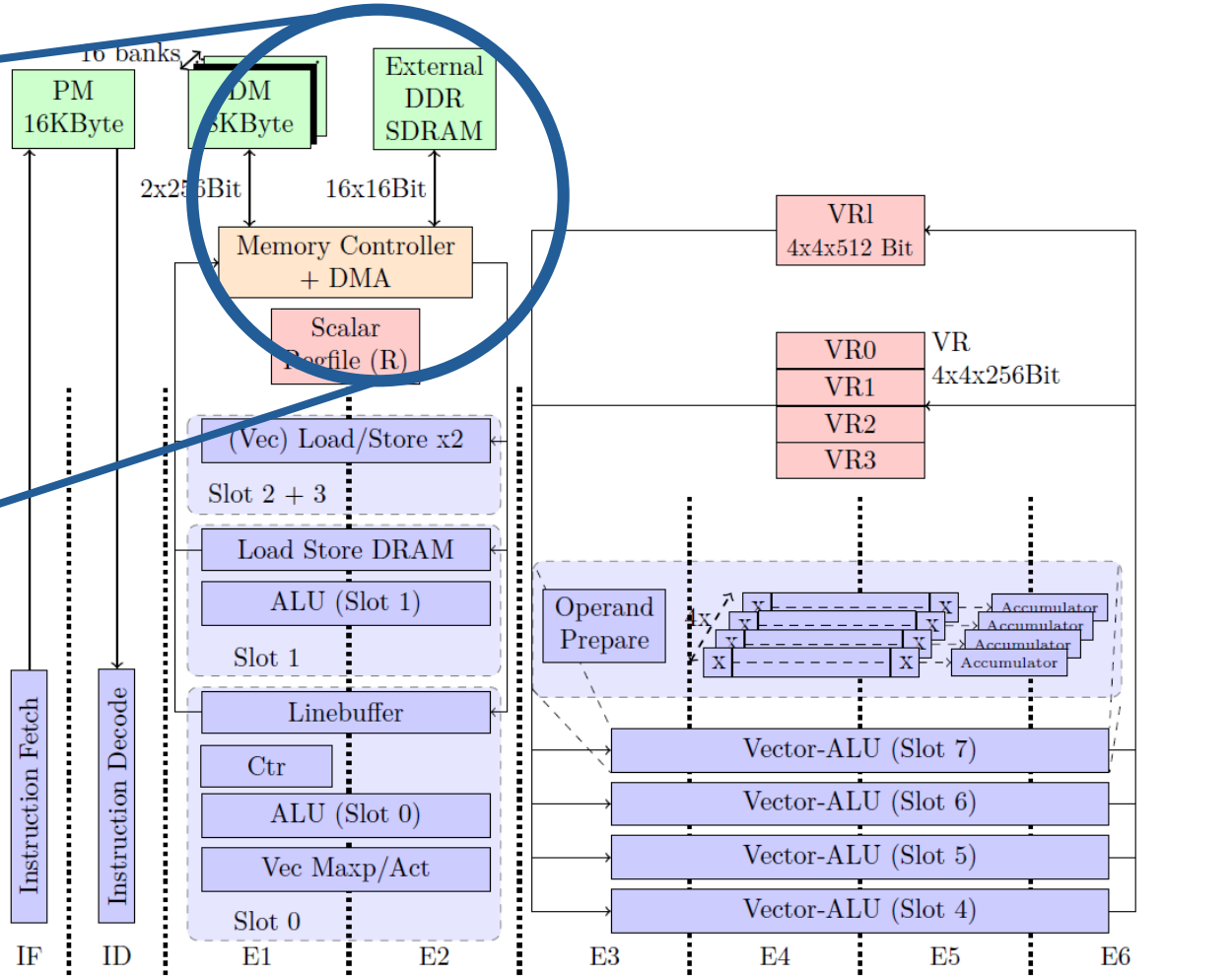
- Instruction-level parallelism: 8 slots in the VLIW allow to utilise several functional units in parallel
- Data-level parallelism: 4 Vector ALUs increase the amount of data that can be processed at a time
  - 1 Vector equals 16x16 bit
- Two internal memories (PM + DM) and an external SDRAM



ConvAix ASIP Architecture

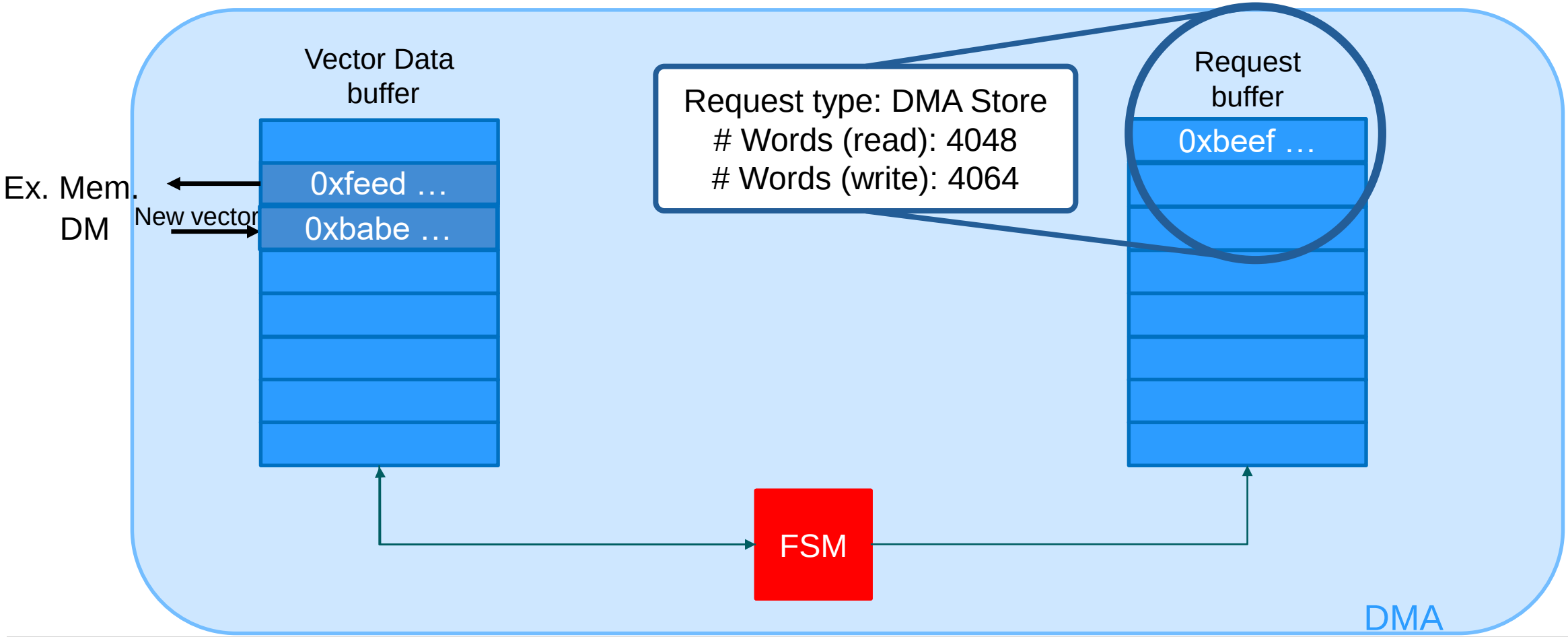
Base Architecture: CNN ASIP (ConvAix)

Data transfers between the on- and off-chip memory are controlled by the DMA

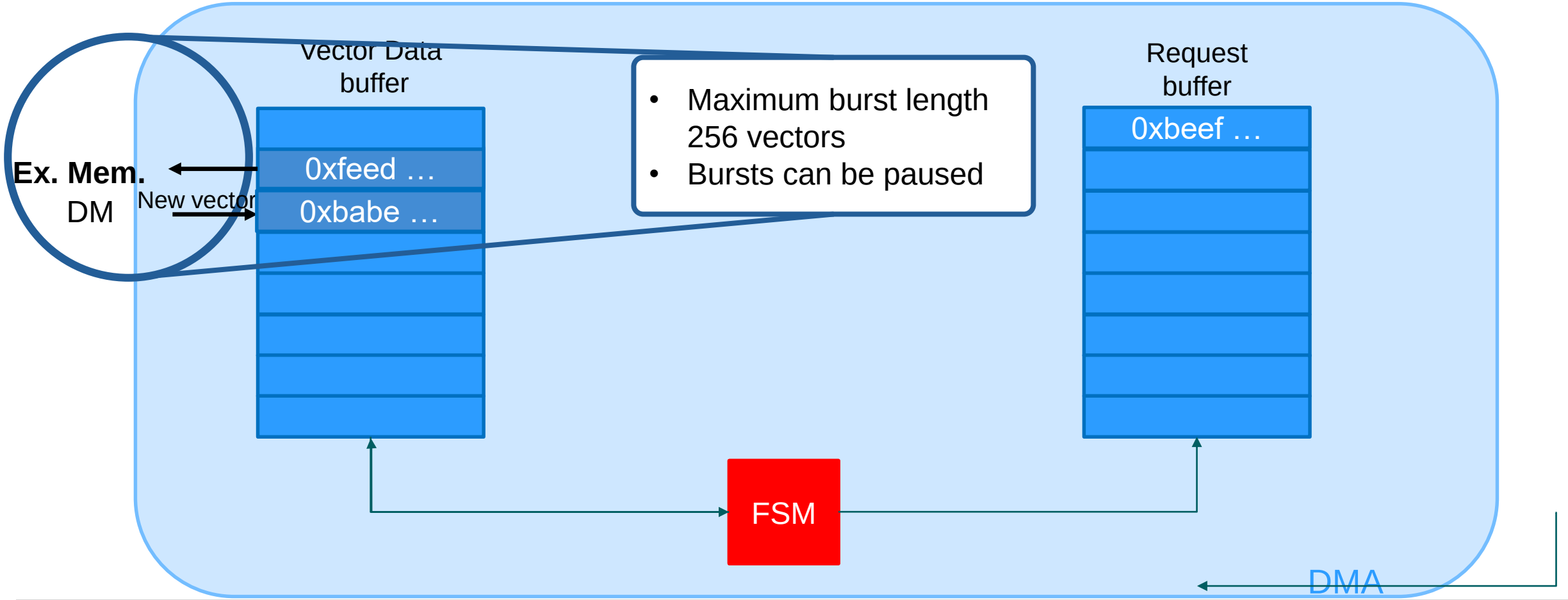


ConvAix ASIP Architecture

DMA Functionality



DMA Functionality

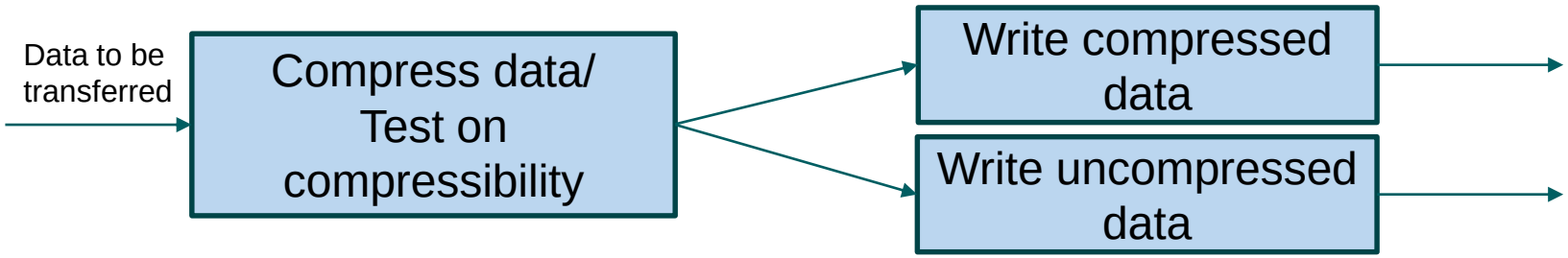


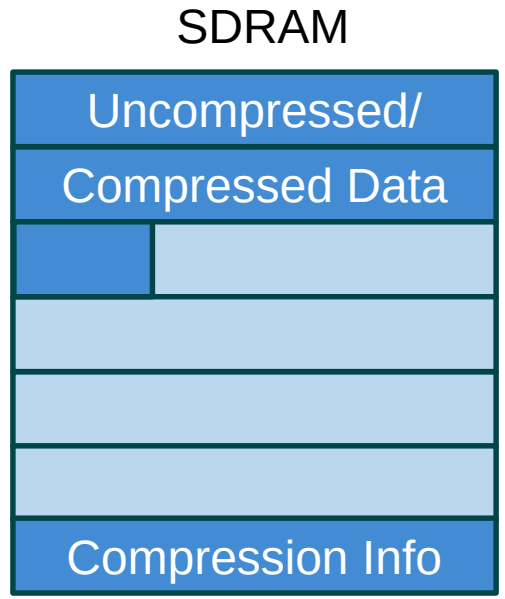
Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

Requirements for the Compression Unit

- A capping procedure to handle incompressible data transfers:





- Testing data before sending requires data to be buffered
  - Buffering all 256 vectors results in enormous register files
    - ➢ Burst has to be split into smaller bursts

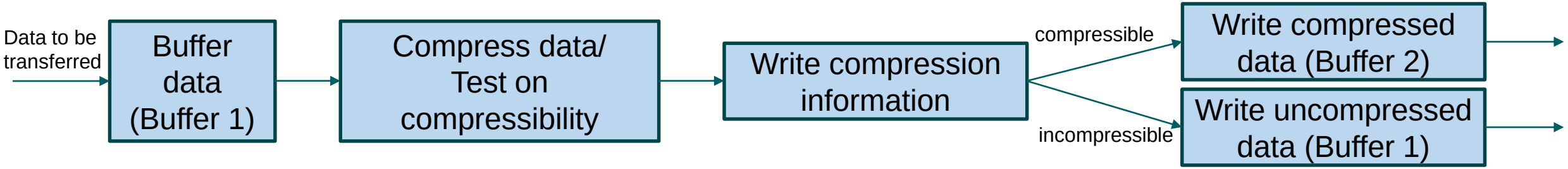


Illustration of the Compression Unit

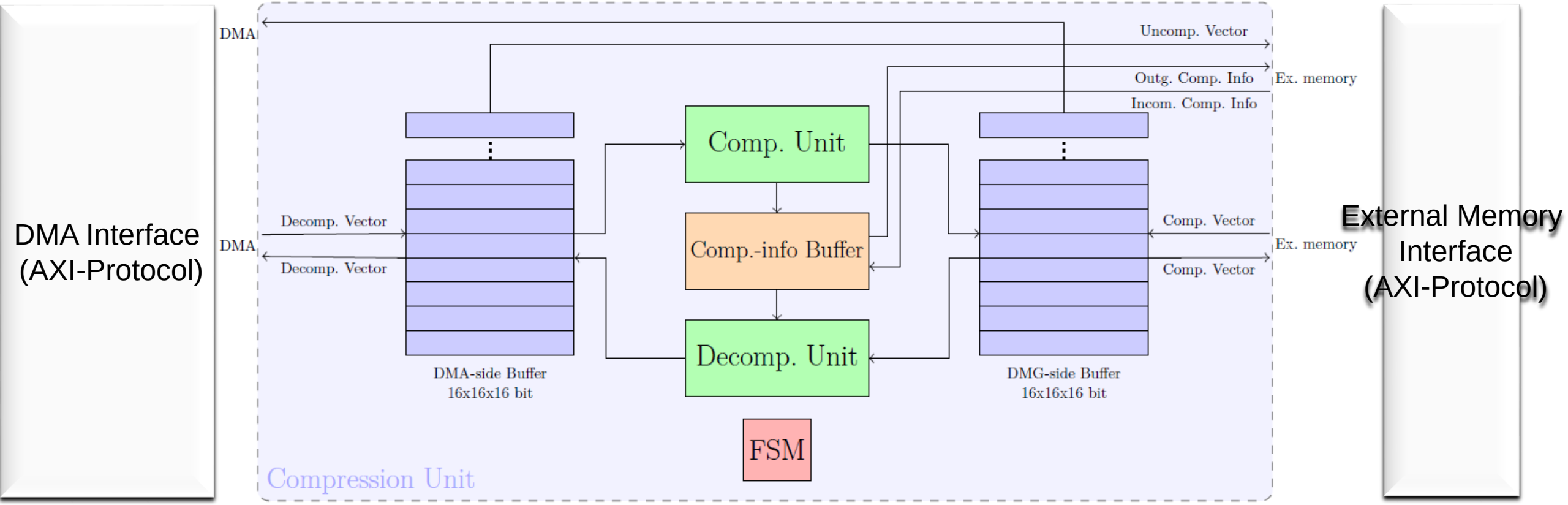
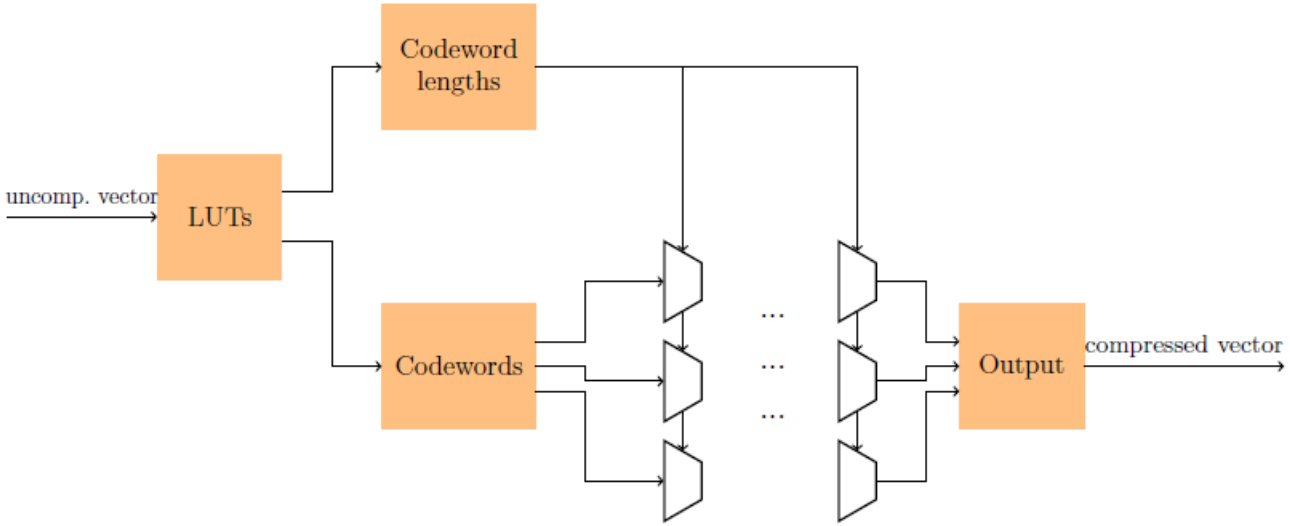


Illustration of the Huffman implementation

- Data in the DMA-side buffer is compressed over several cycles
- Same timing constraint as the base architecture
- Length of the multiplexer tree limits the number of bits that can be compressed in a cycle



Outline

1. CNN Structure
2. SDRAM Power Consumption
3. Compression Algorithms
4. ASIP Architecture
5. Hardware Modifications
6. Hardware Evaluation

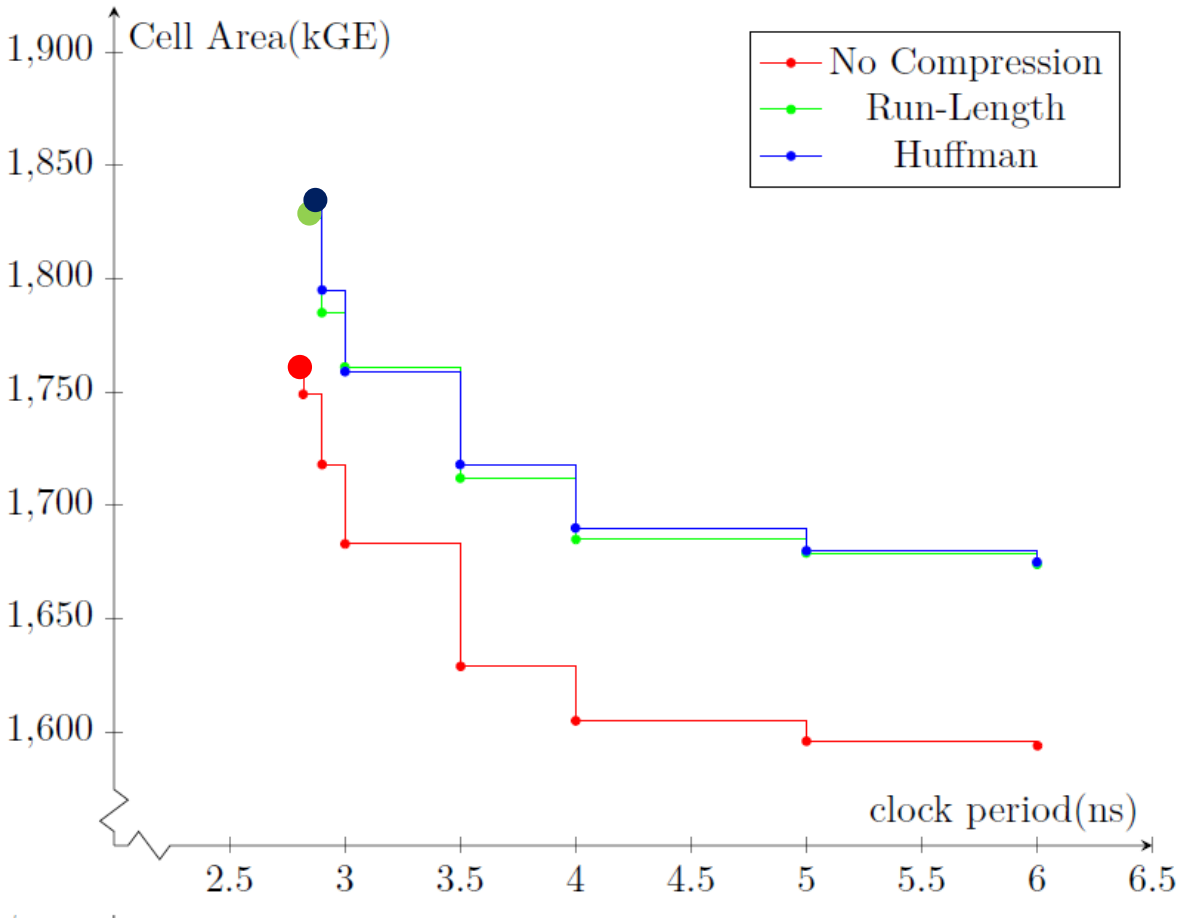
Hardware Synthesis Results: Area over Time Diagram

- Goal: Low additional latency + similar critical path length as base design
  - ➢ All designs achieve the AT-optimum with a Timing-Constraint of: 2.8ns

- Huffman requires slightly more cell area

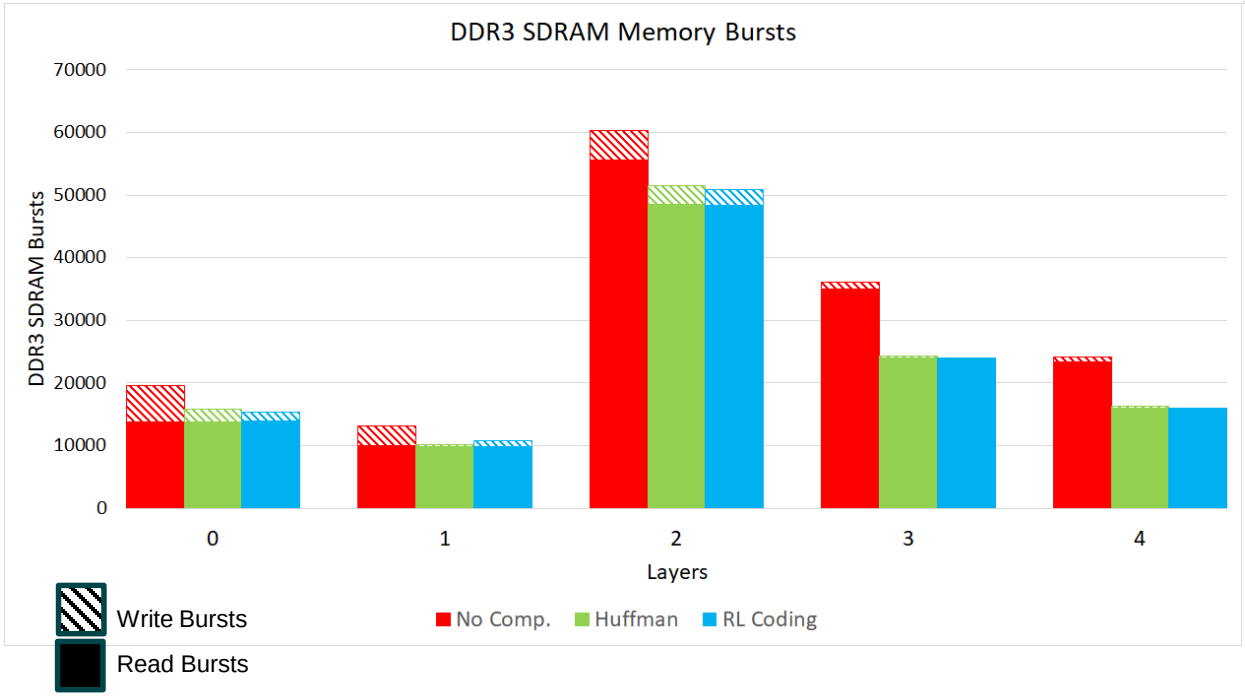
- Total area increases by (AT-optimal):
  - 4.2% for the Huffman implementation
  - 3.8% for the RL implementation

- UMC 90nm; 25°C; Std. Performance; Std. Vt; 1Vcc

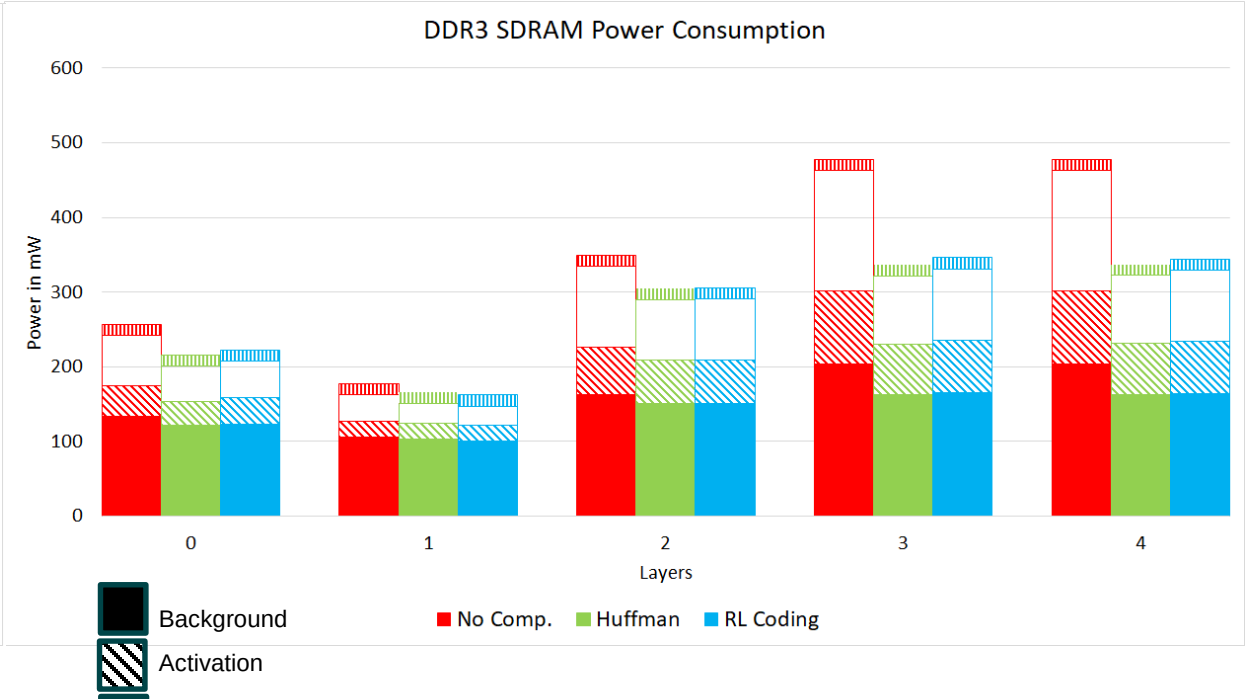


Area-Time Diagram for the three implementations (Base Architecture, Base + Huffman Comp. Unit, Base + RL Comp. Unit)

DRAM Power Consumption (AlexNet)

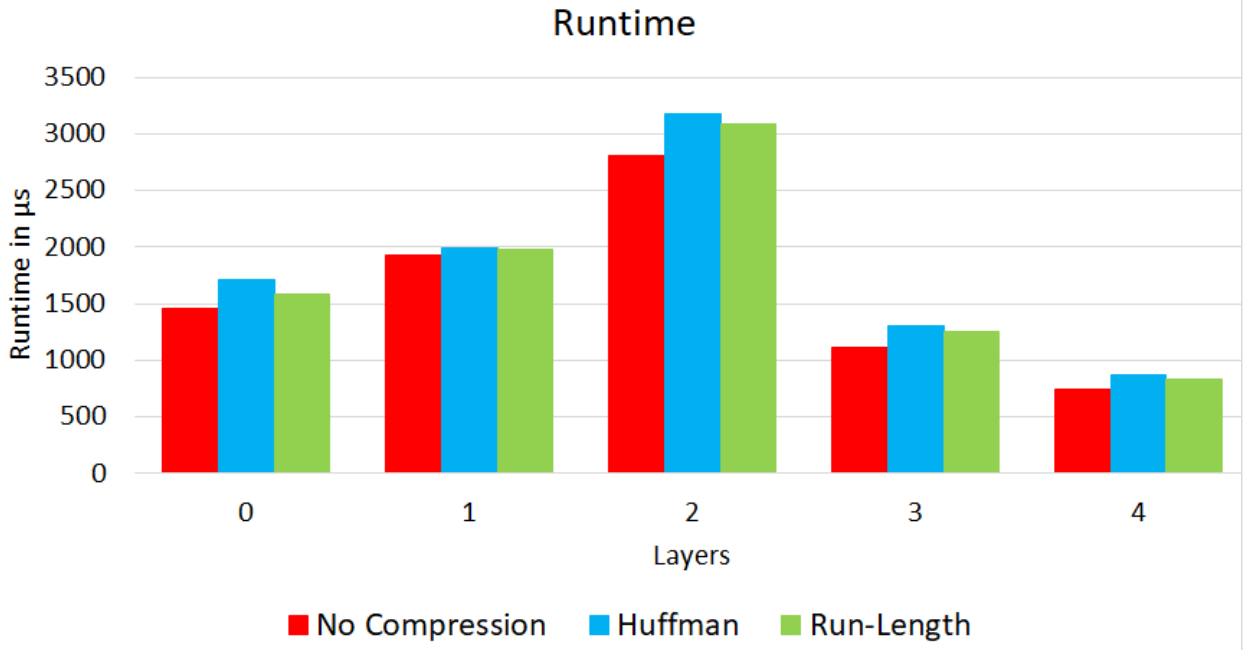


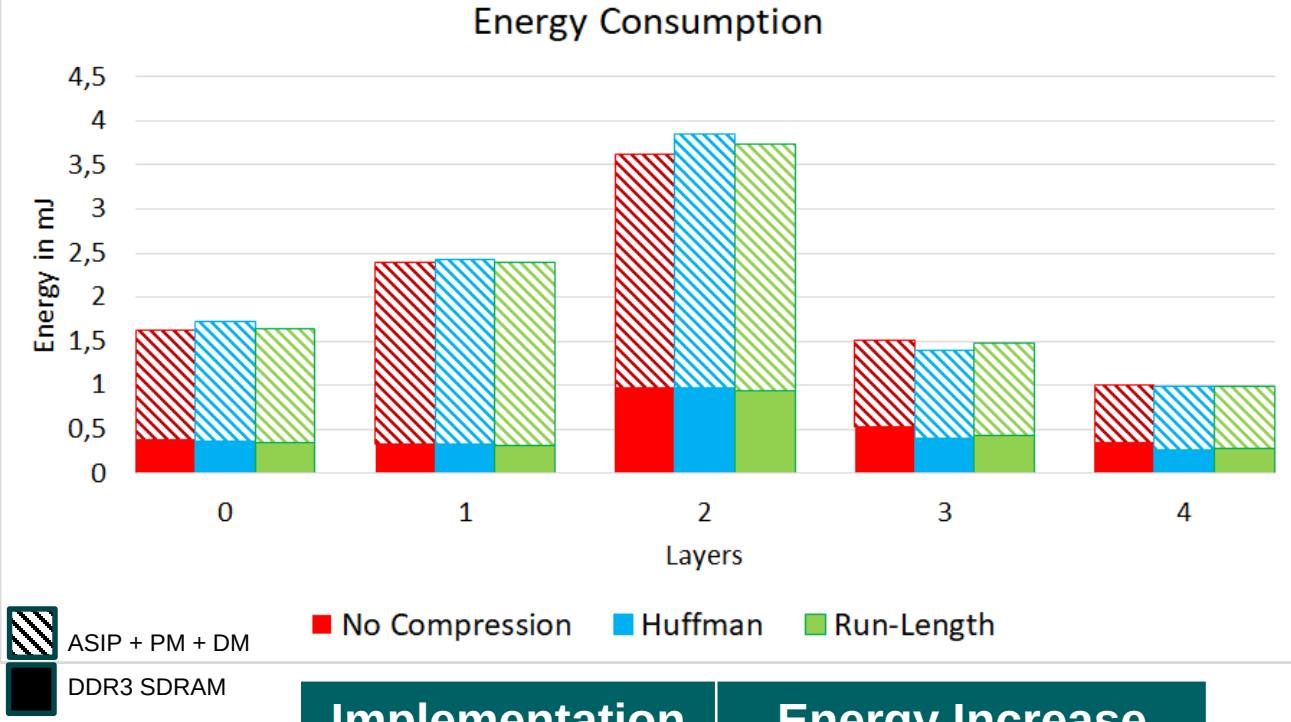
Implementation	Burst Reduction
Run-Length	-23.6%
Huffman	-22.9%



Implementation	Power Reduction
Run-Length	-20.6%
Huffman	-21.8%

Runtime + Energy Consumption (AlexNet)





Implementation	Runtime Increase
Run-Length	+8.4%
Huffman	+12.3%

Implementation	Energy Increase
Run-Length	+0.7%
Huffman	+2.2%

Conclusion + Future Work

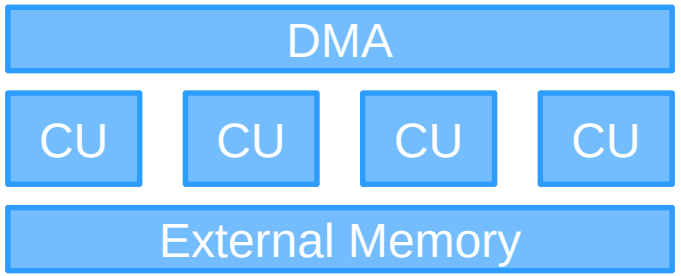
Summary + Conclusion:

- DDR3 SDRAM power consumption is reduced in all layers
- Total runtime is increased in all layers
- Total energy consumption of the external memory is reduced
- Energy increase of the core outweighs energy reduction of the memory

➢ Compression units do not show an energy reduction for this embedded system

Future Work:

- Different Compression algorithms:
  - Adaptive codebook algorithms might allow better results in every layer
- Reducing the runtime by using multiple compression units (CU) in parallel:



Thank you for your attention!

References

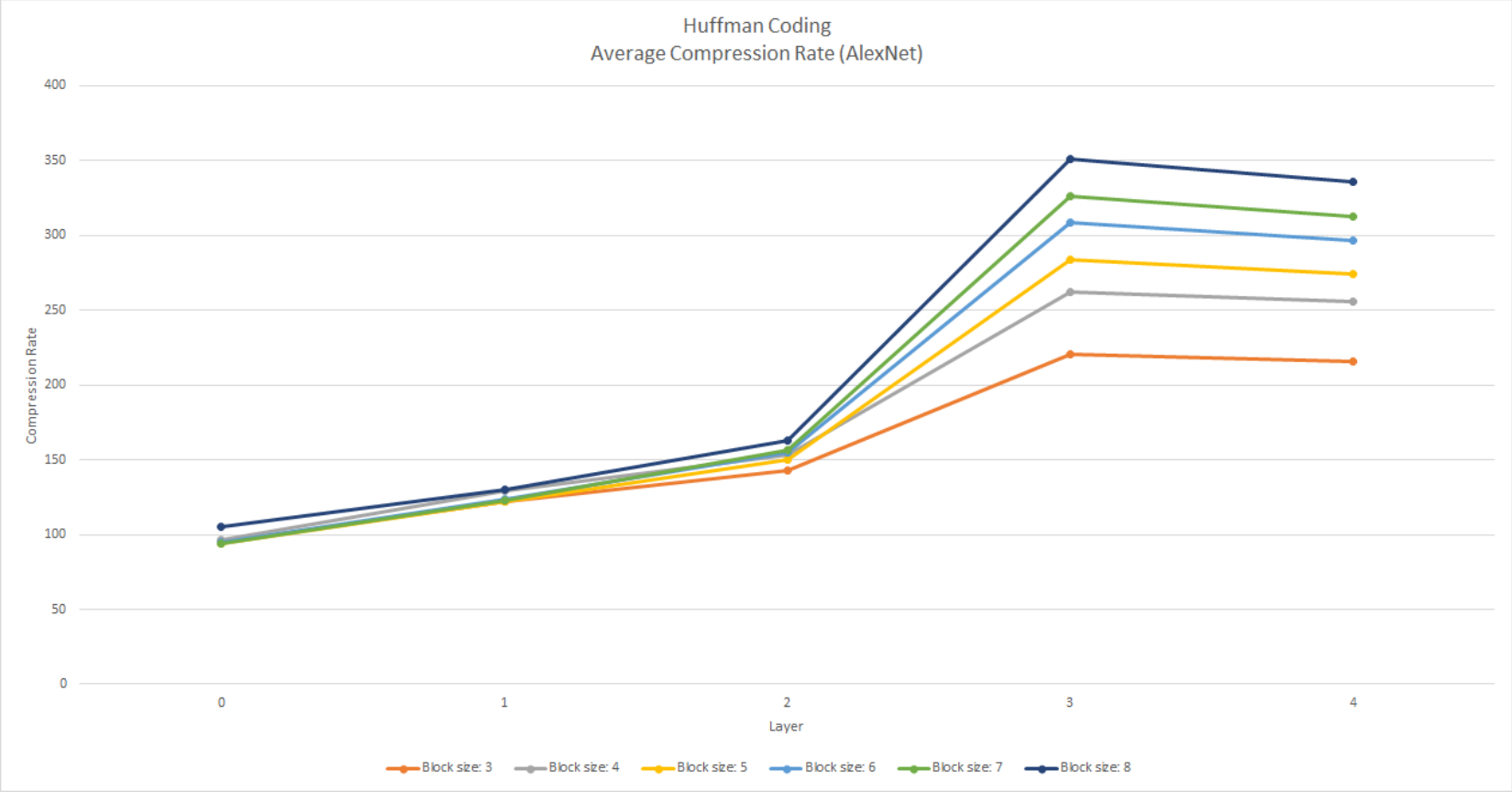
[1]:https://imageog.flaticon.com/icons/png/512/0/488.png?size=1200x630f&pad=10,10,10,10&ext=png&bg=FFFFFFFF

[2] : https://image.freepik.com/free-icon/battery-with-a-bolt-symbol_318-62272.png

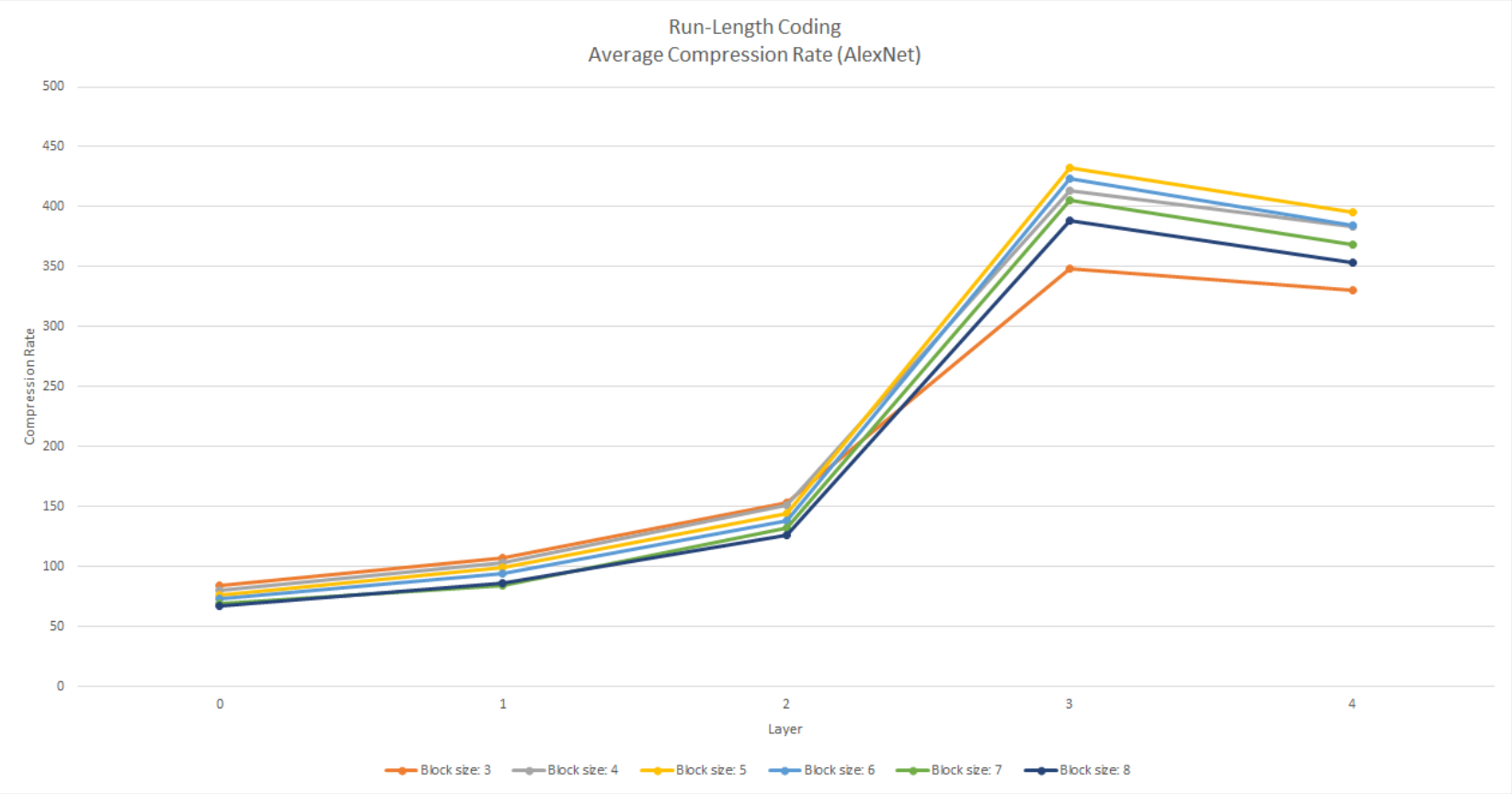
[3] : A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Proc. Conf. Neural Information Processing Systems, 2012, pp. 1097-1105.

[4] : B. Jacob, S. W. Ng, and D. Wang, Memory Systems: Cache, DRAM, Disk. Morgan Kaufmann Publishers, 2008, pp. 425-456.

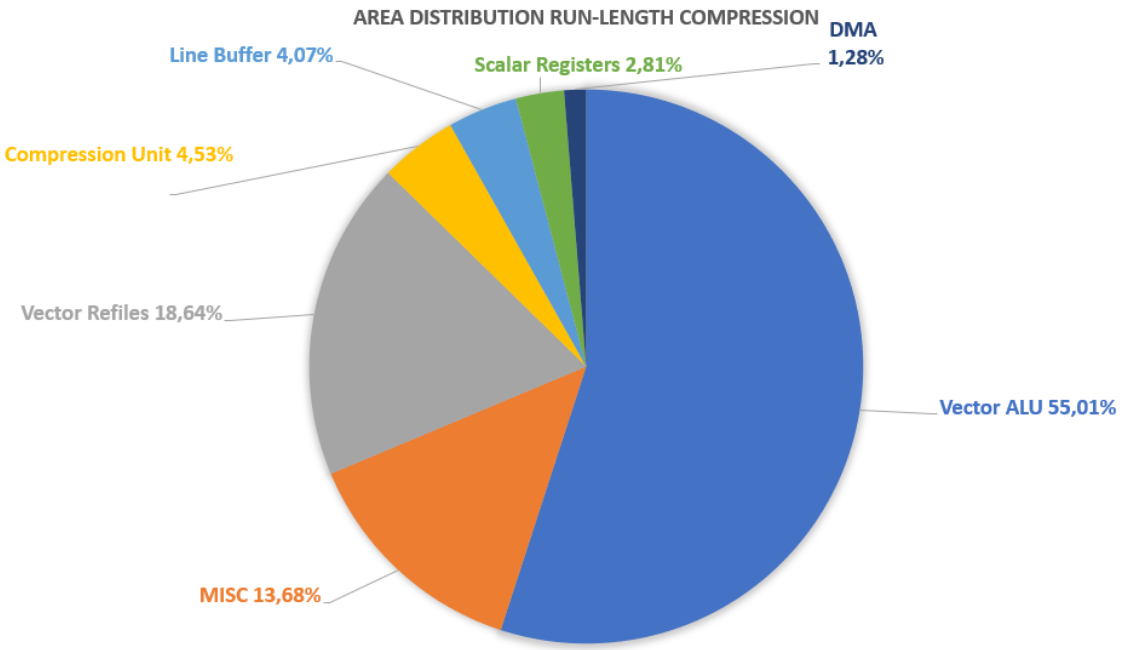
Huffman Coding Average Compression Rate

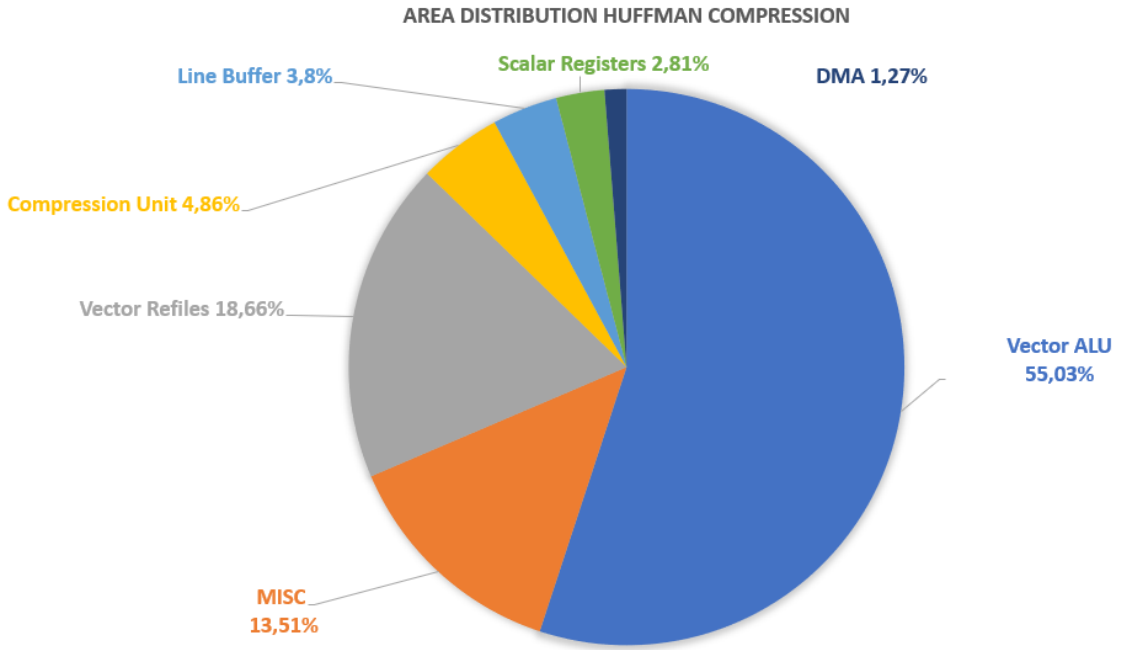


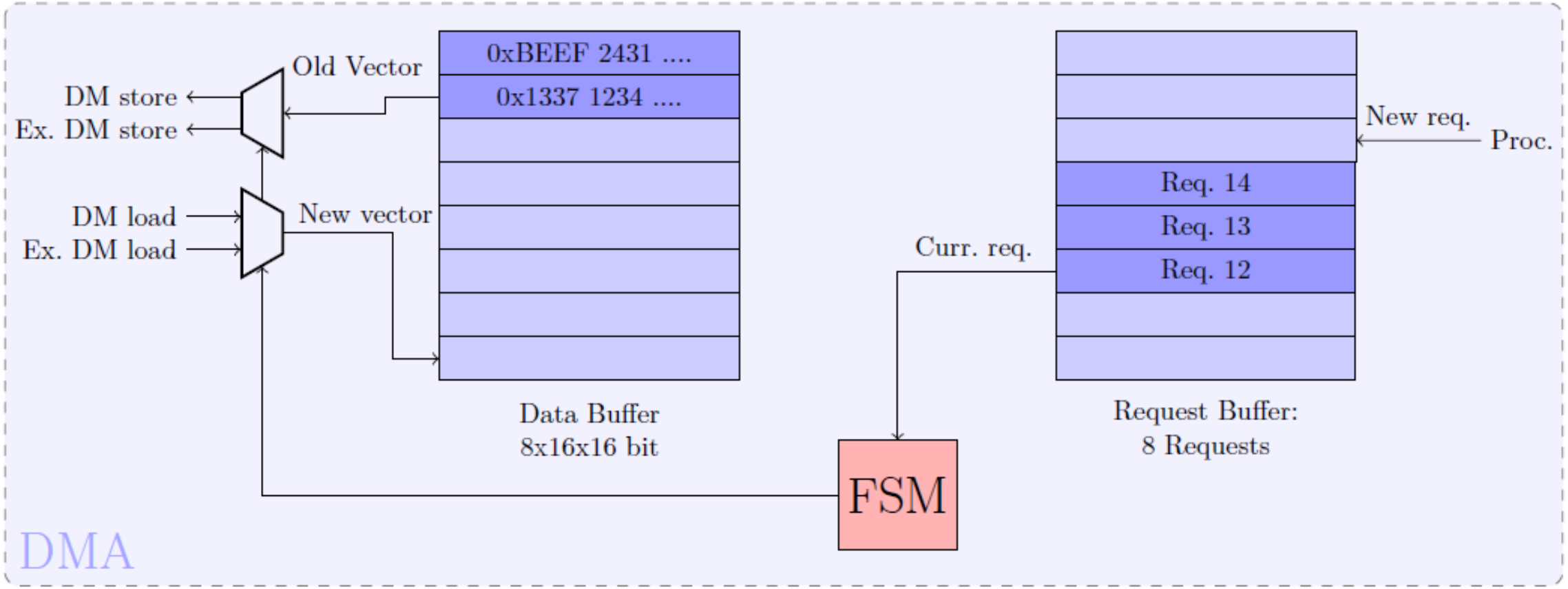
Run-Length Coding Average Compression Rate (AlexNet)



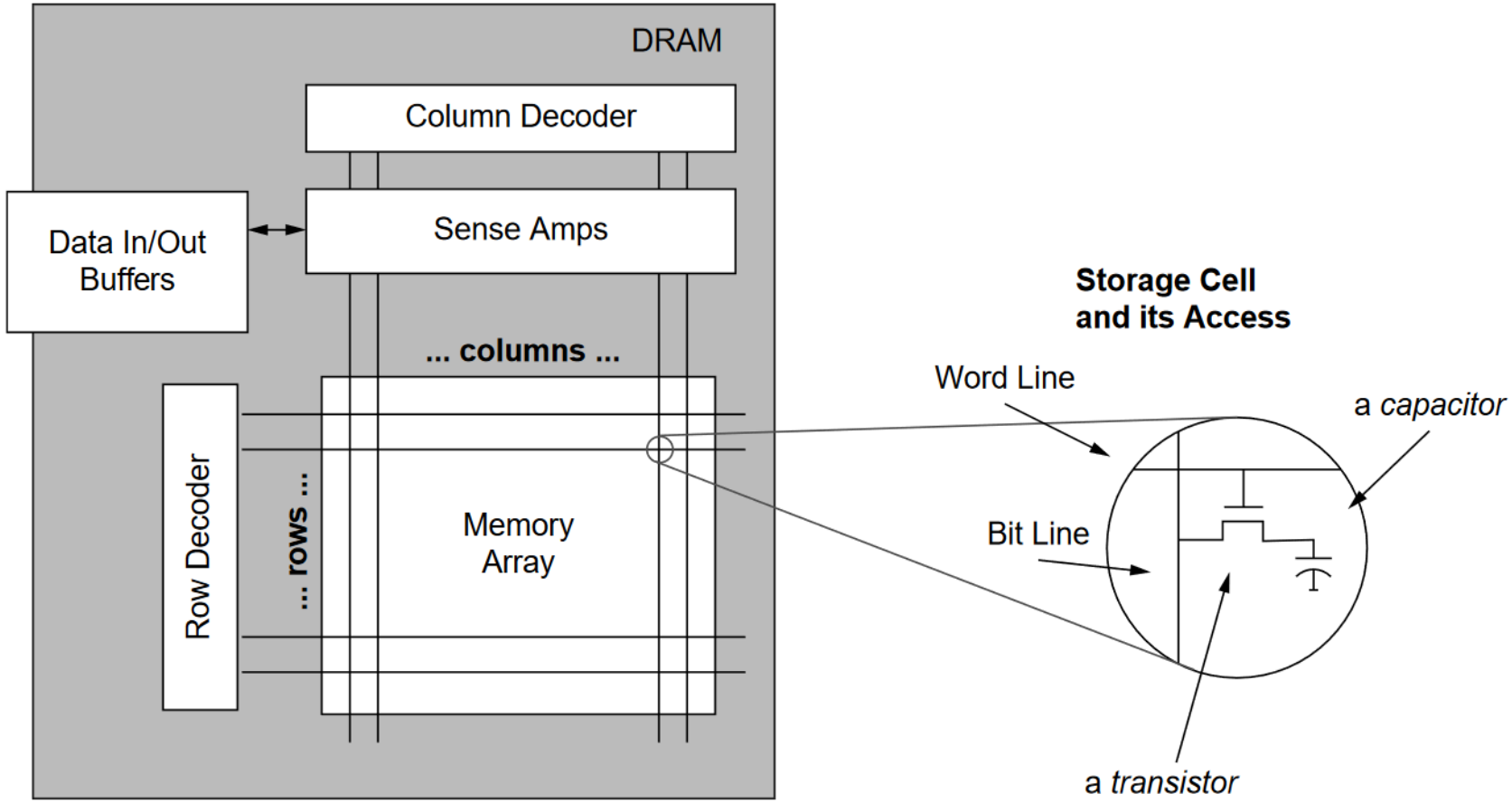
Hardware Synthesis Results: Area Distribution of the two modified Designs



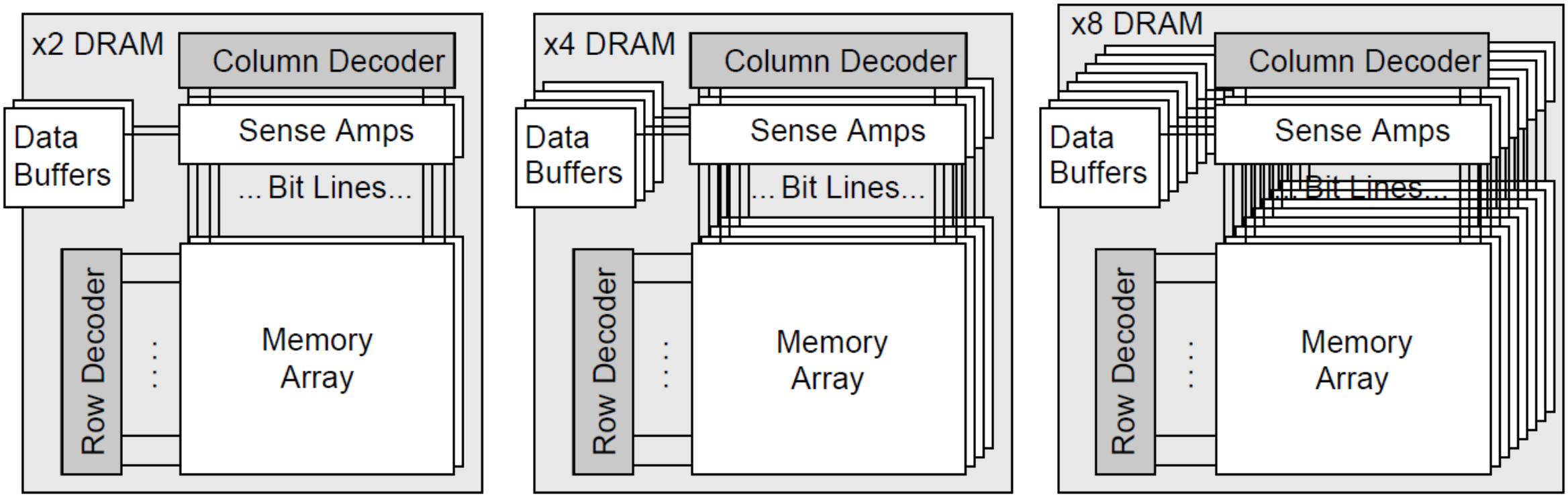



DM store
Ex. DM store
Old Vector
0xBEEF 2431 ....
0x1337 1234 ....
DM load
Ex. DM load
New vector
Data Buffer
8x16x16 bit
Curr. req.
FSM
Req. 14
Req. 13
Req. 12
New req.
Proc.
Request Buffer:
8 Requests
DMA

SDRAM Memory Array

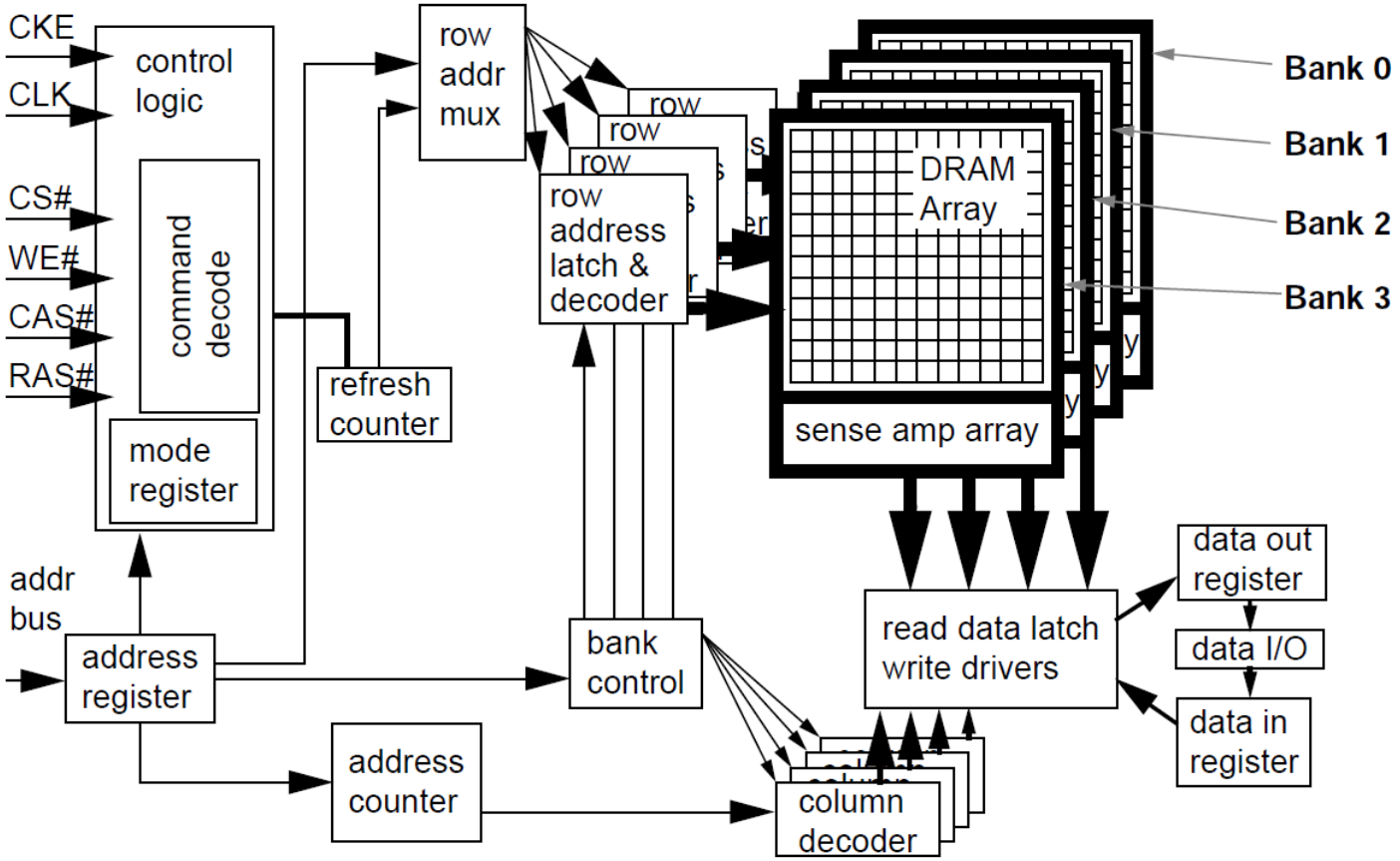


SDRAM Column Width

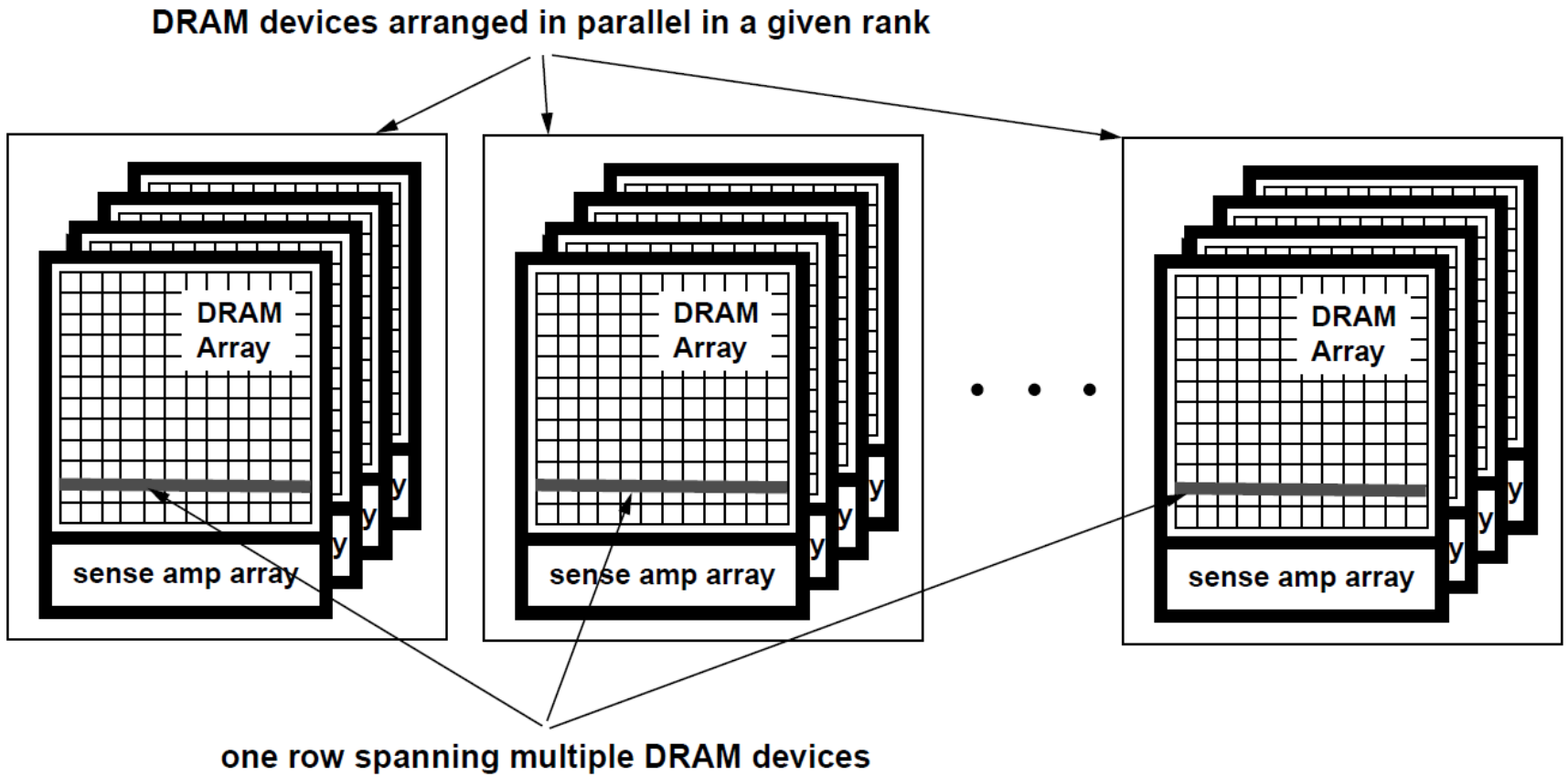


x2 DRAM **x4 DRAM** **x8 DRAM**

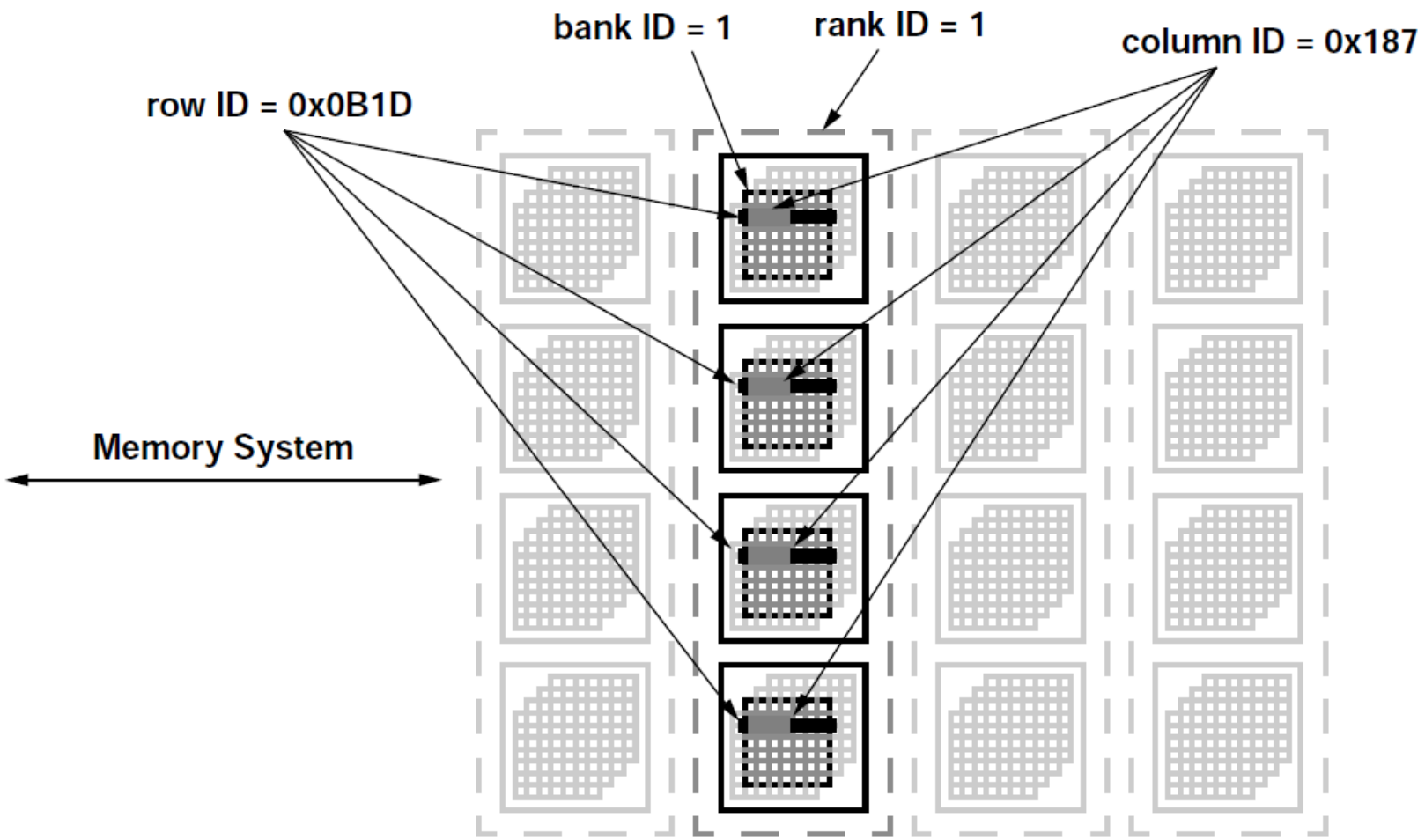
SDRAM Bank organisation



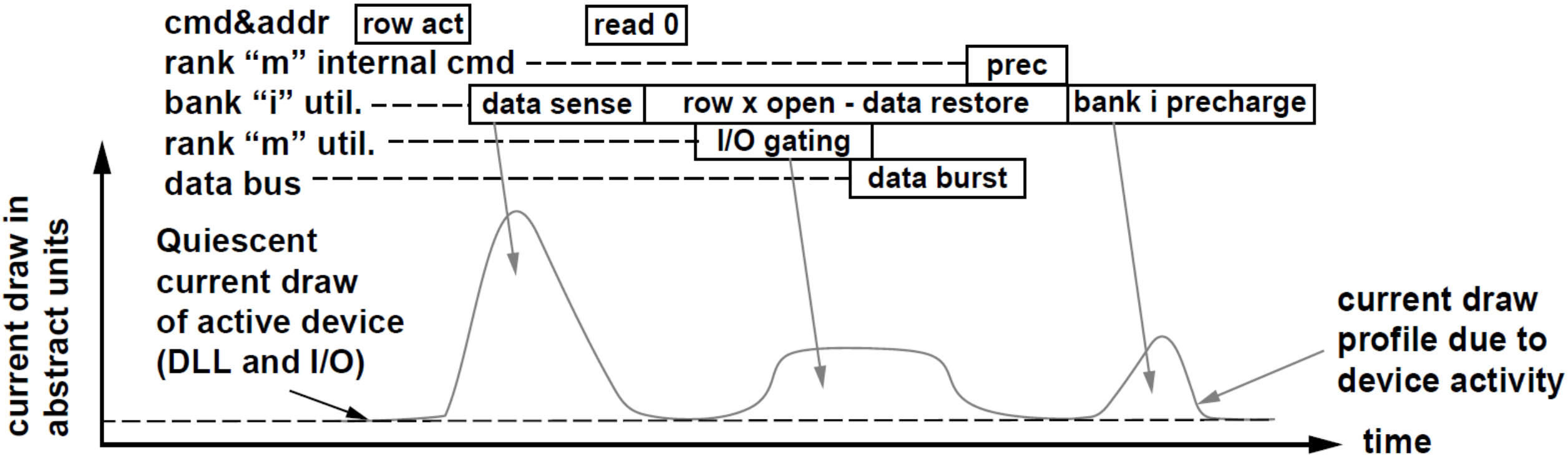
SDRAM Rank organisation



High level structure

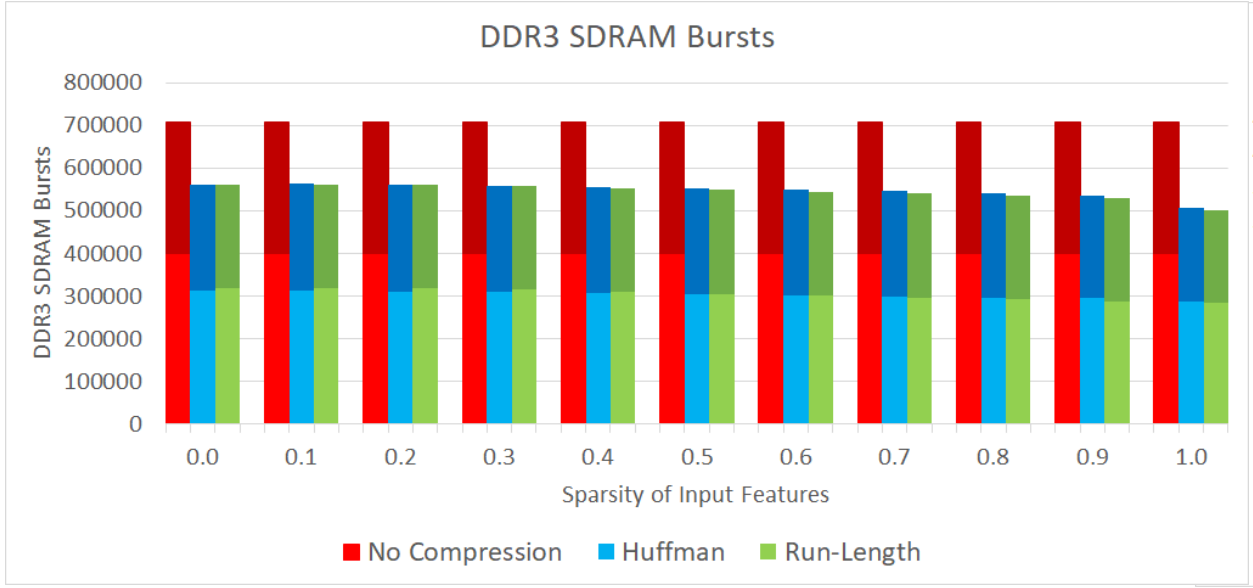


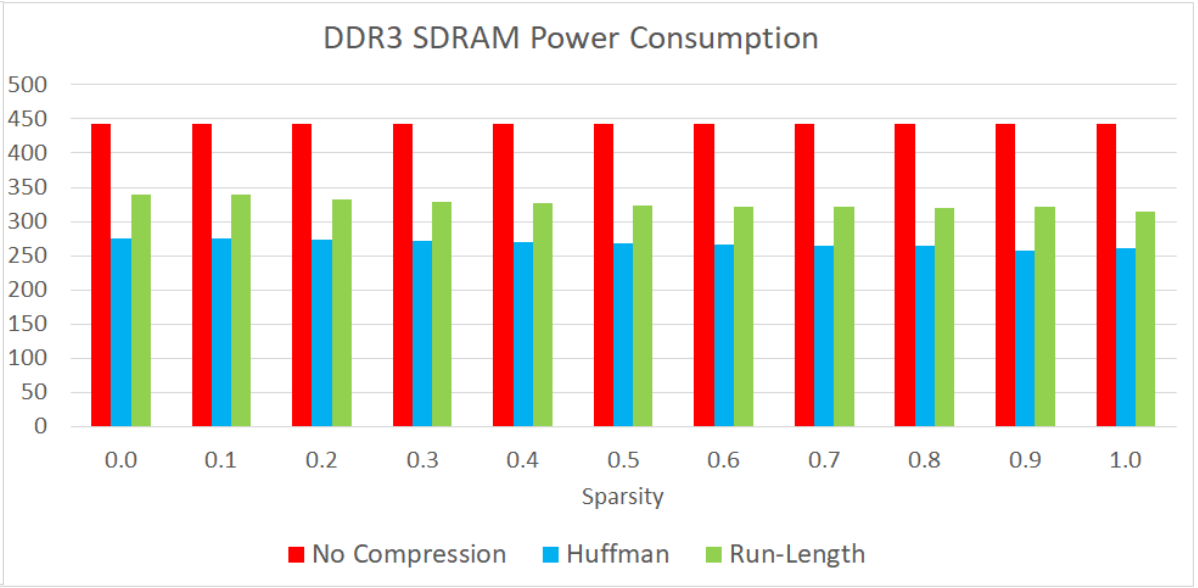
Current Draw During a Read Operation



Abstract Illustration of the current draw during a read operation [1]

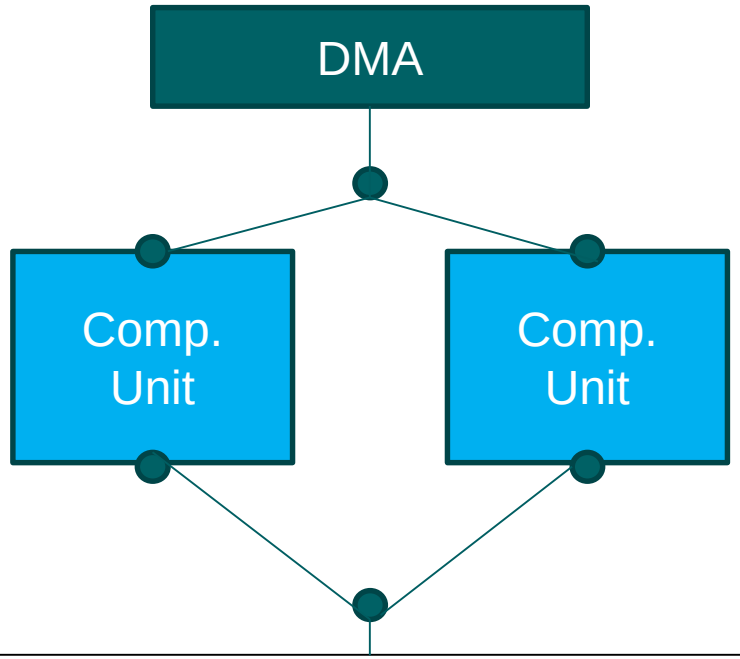
Impact of the Input Feature Map‘s Sparsity





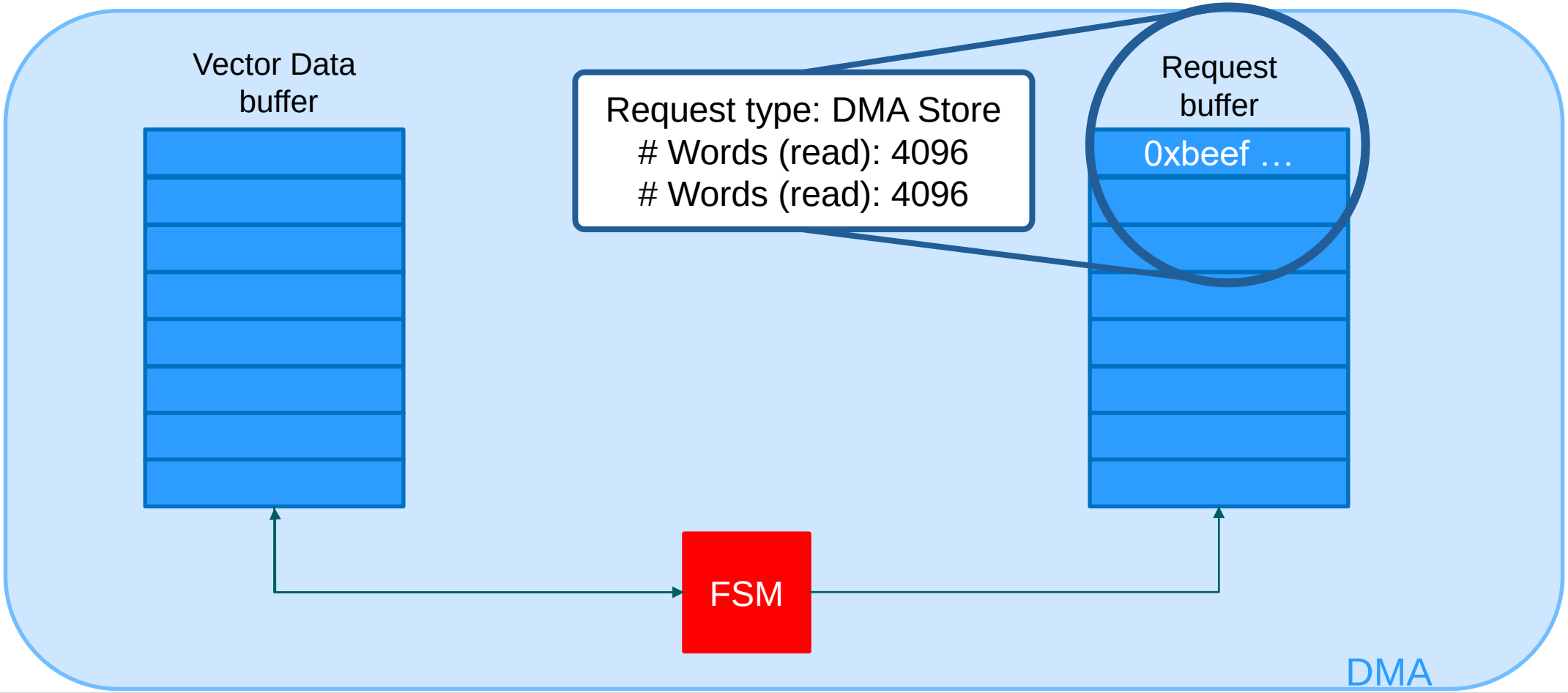
Future Work

- Different Compression algorithms:
  - Adaptive codebook algorithms might allow better results in every layer
- Reducing the runtime by using multiple compression units in parallel:

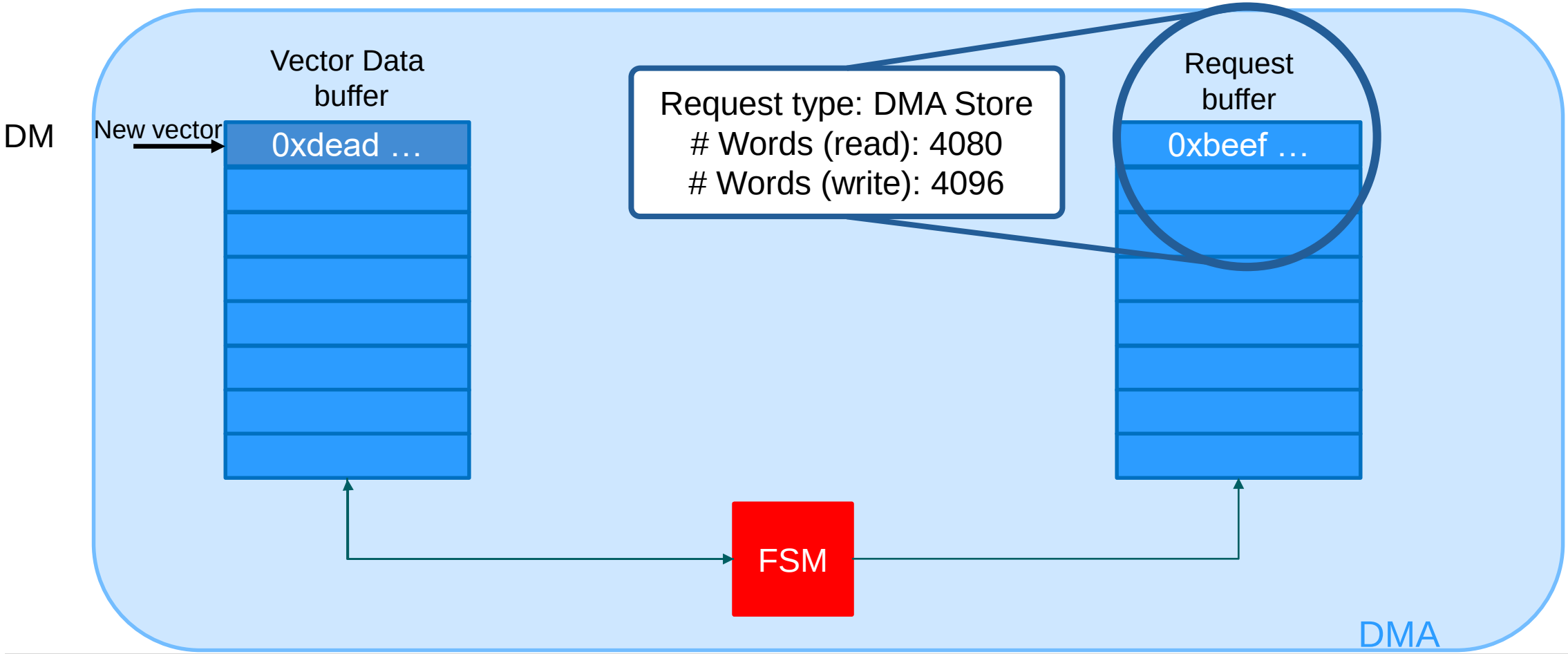


DMA Functionality

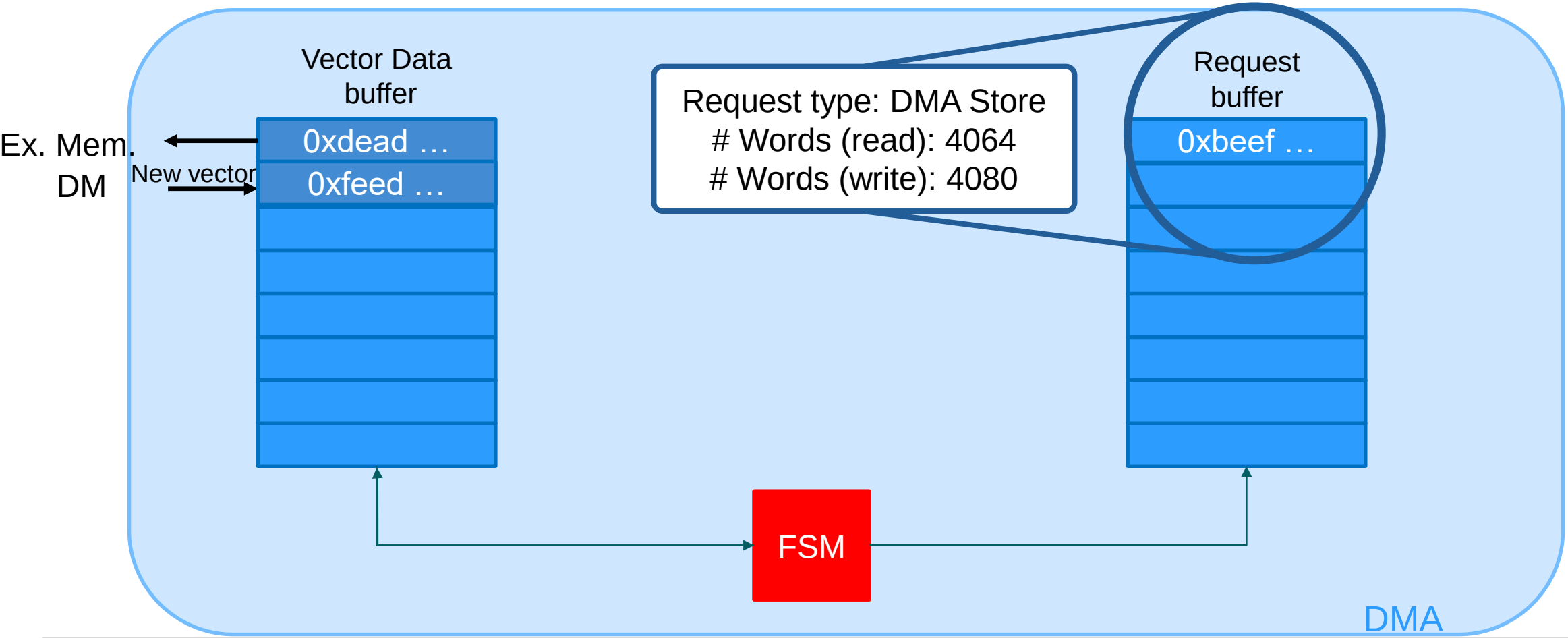
DMA Functionality



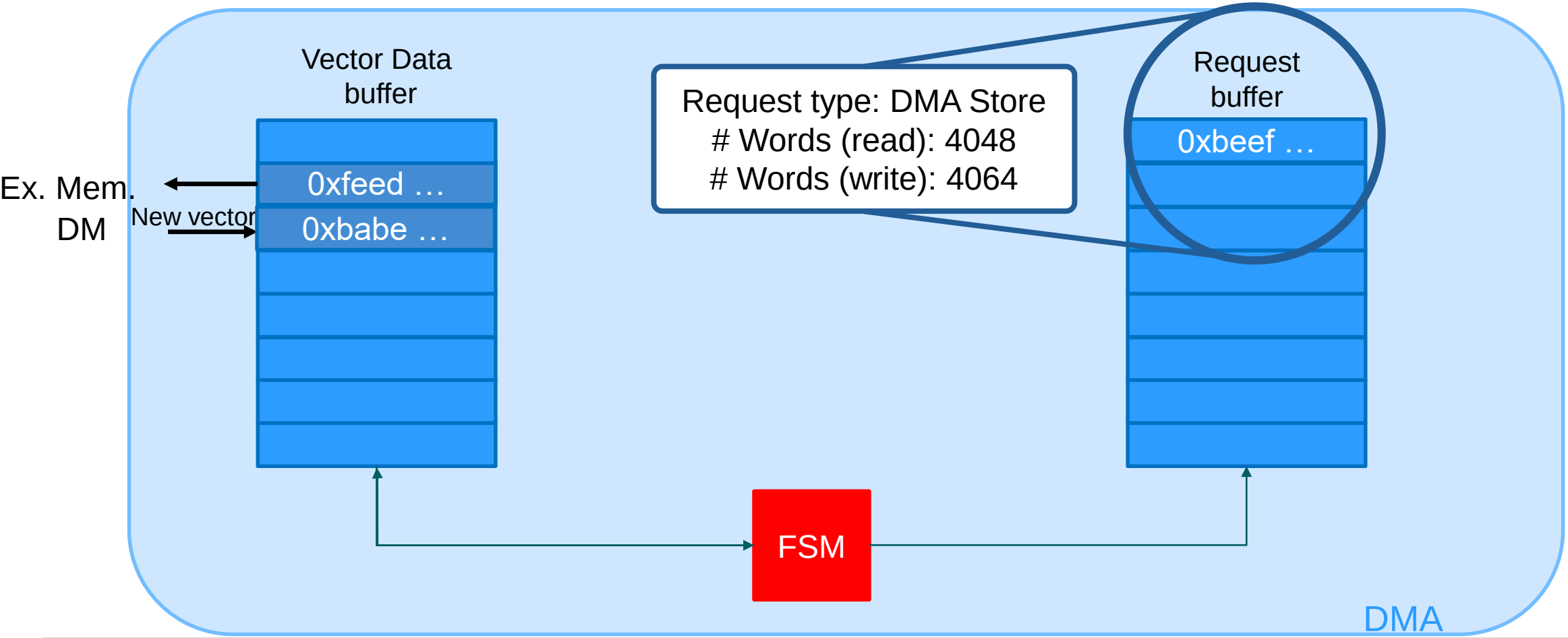
DMA Functionality



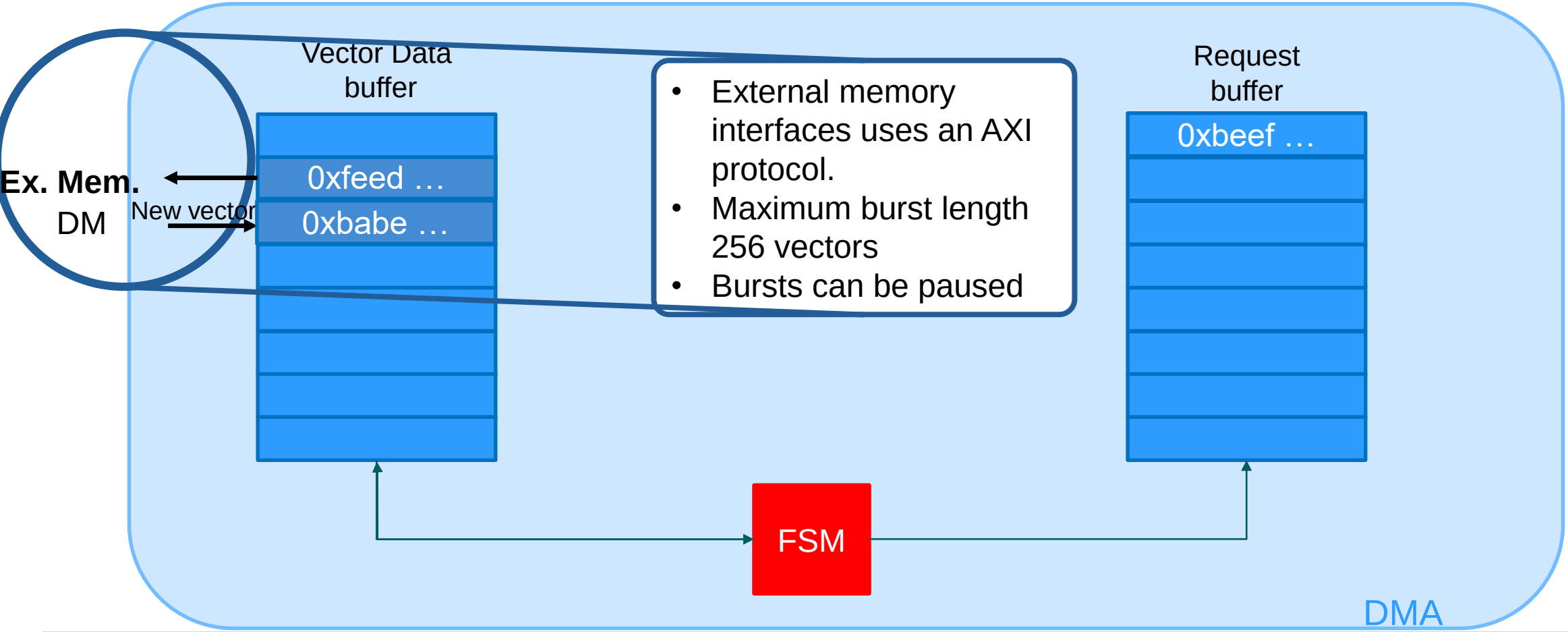
DMA Functionality

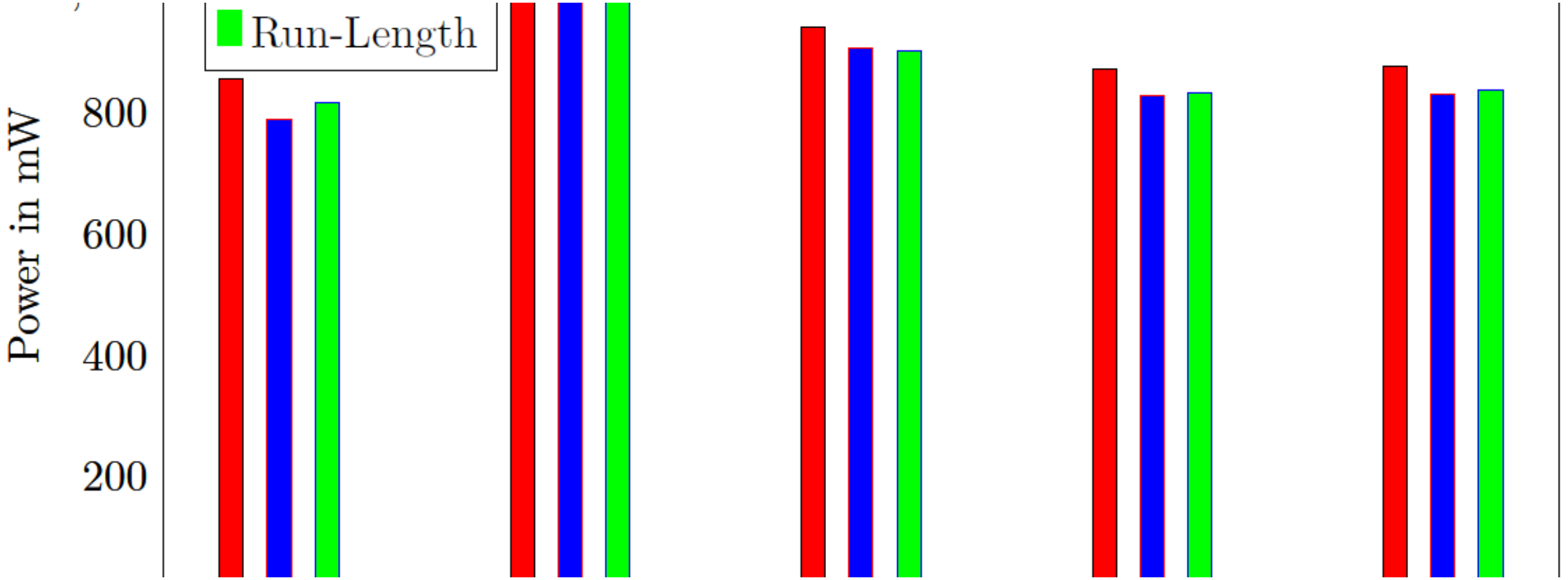


DMA Functionality



DMA Functionality




Run-Length
Power in mW
800
600
400
200