MX+: Pushing the Limits of Microscaling Formats for Efficient Large Language Model Serving

Jungi Lee Junyong Park Soohyun Cha
Jaehoon Cho Jaewoong Sim

Seoul National University

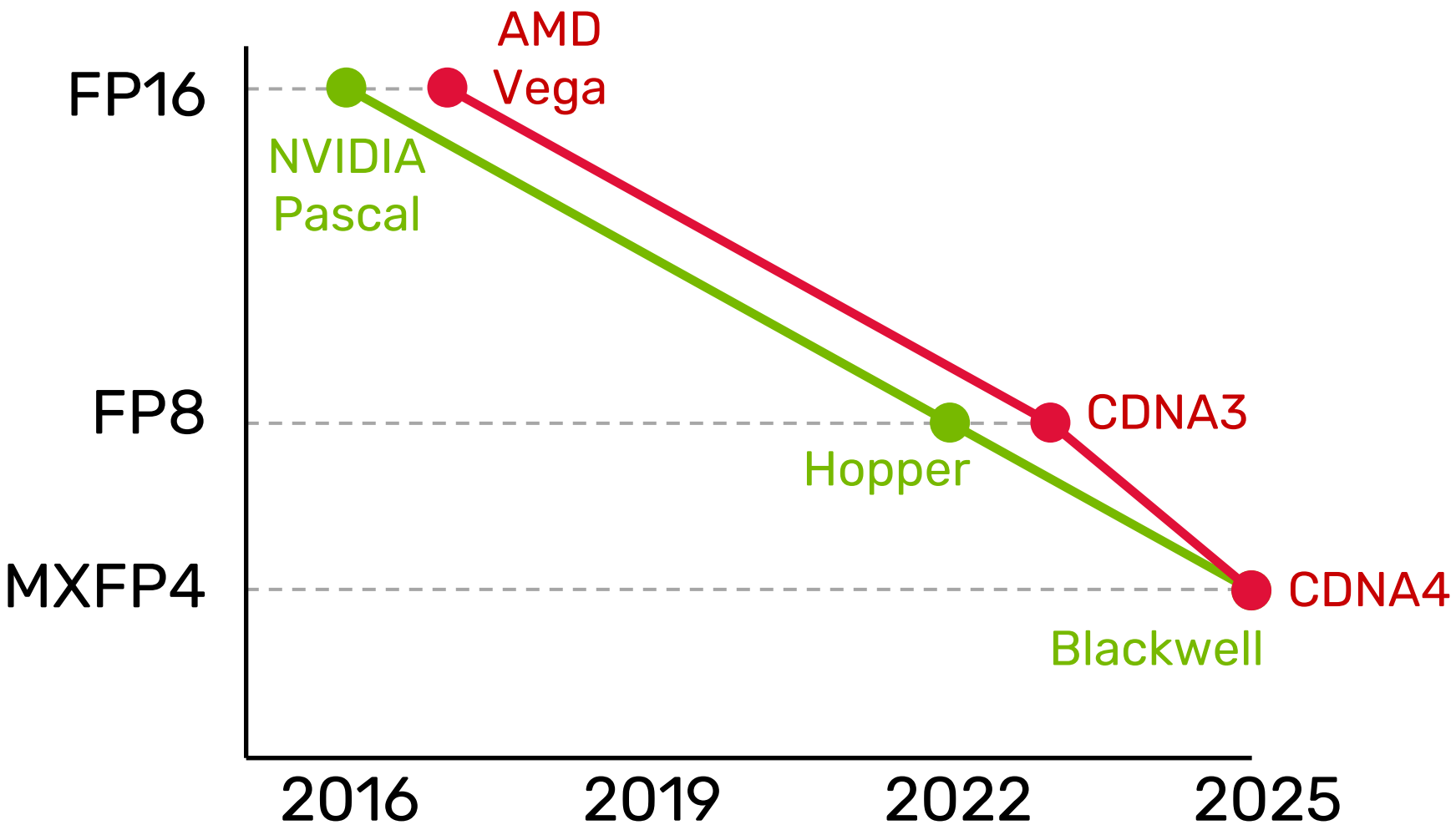
MICRO'25 | October 2025

Department of ECE
Seoul National University

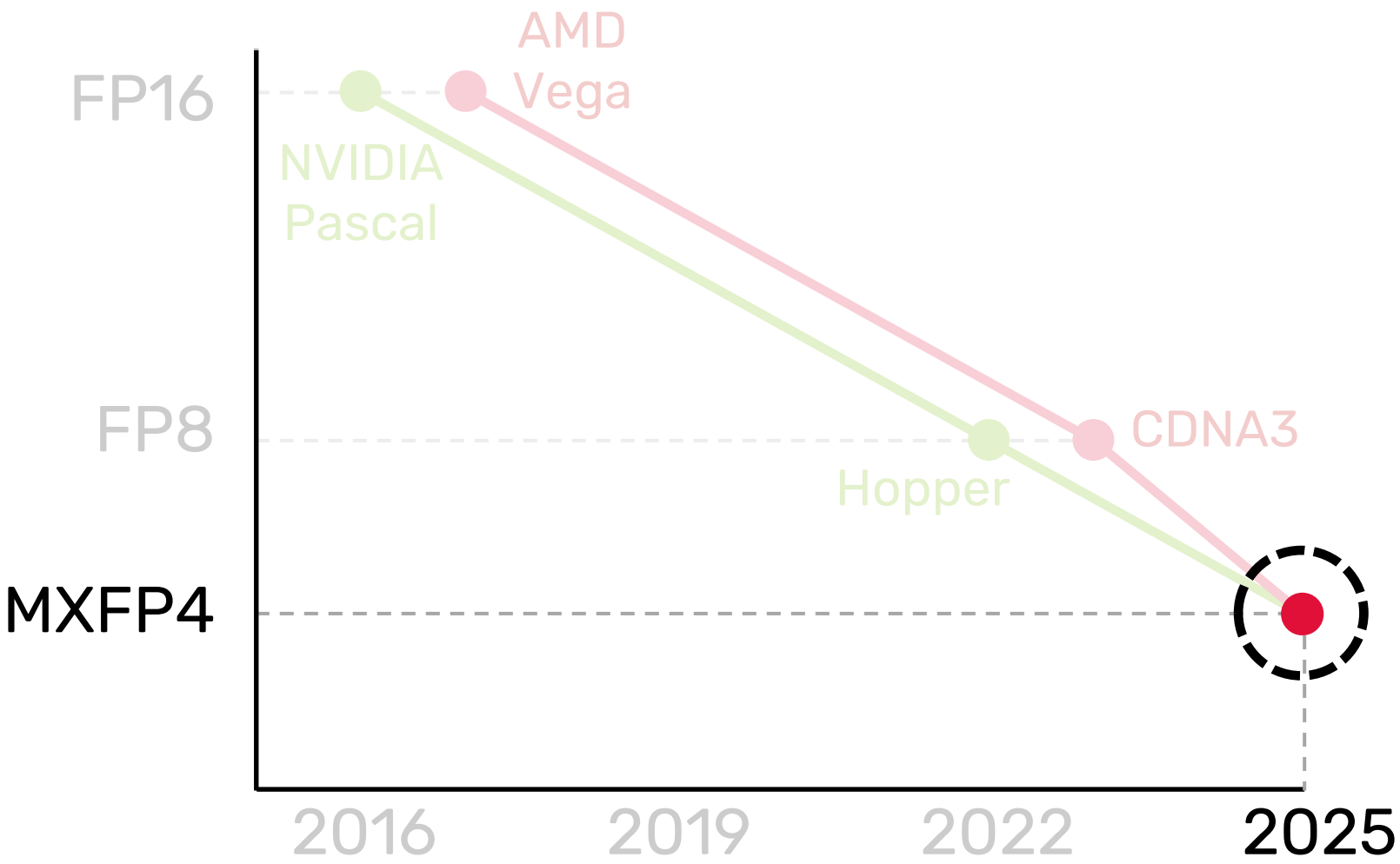
Outline

- **Introduction to MX Formats**
- **Motivation**
  - Implications for Low-bit LLM Serving
  - Analysis on Low-bit MX Formats
- **MX+: Cost-Effective and Non-intrusive Extension to MX Formats**
  - Key Insight
  - Software & Hardware Integration on GPUs
- **Evaluation**
- **Conclusion**

Increasing Support for MXFP4



Increasing Support for MXFP4



LLM

OpenAI gpt-oss

Software

ONNX, MLIR, etc.

GPU

NVIDIA Blackwell GPU

AMD CDNA4 GPU

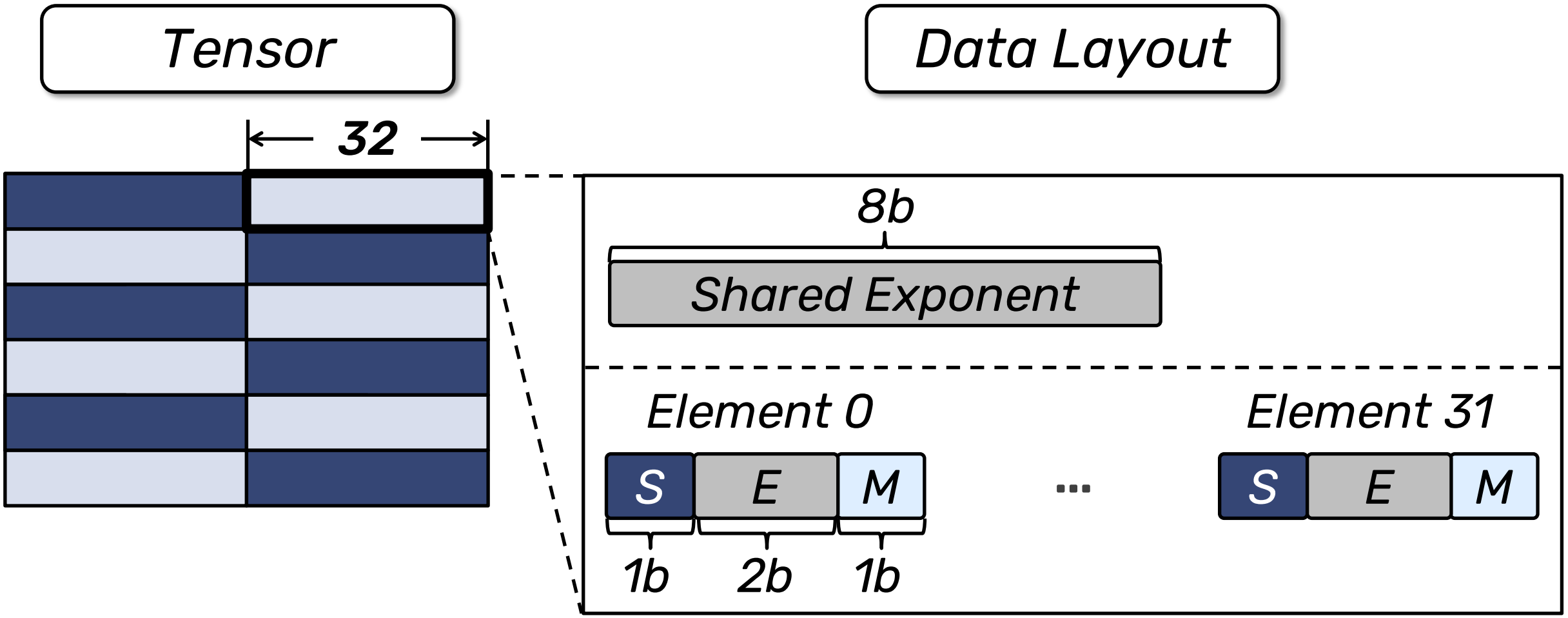
Microscaling (MX) Formats

Open and Interoperable Data Formats

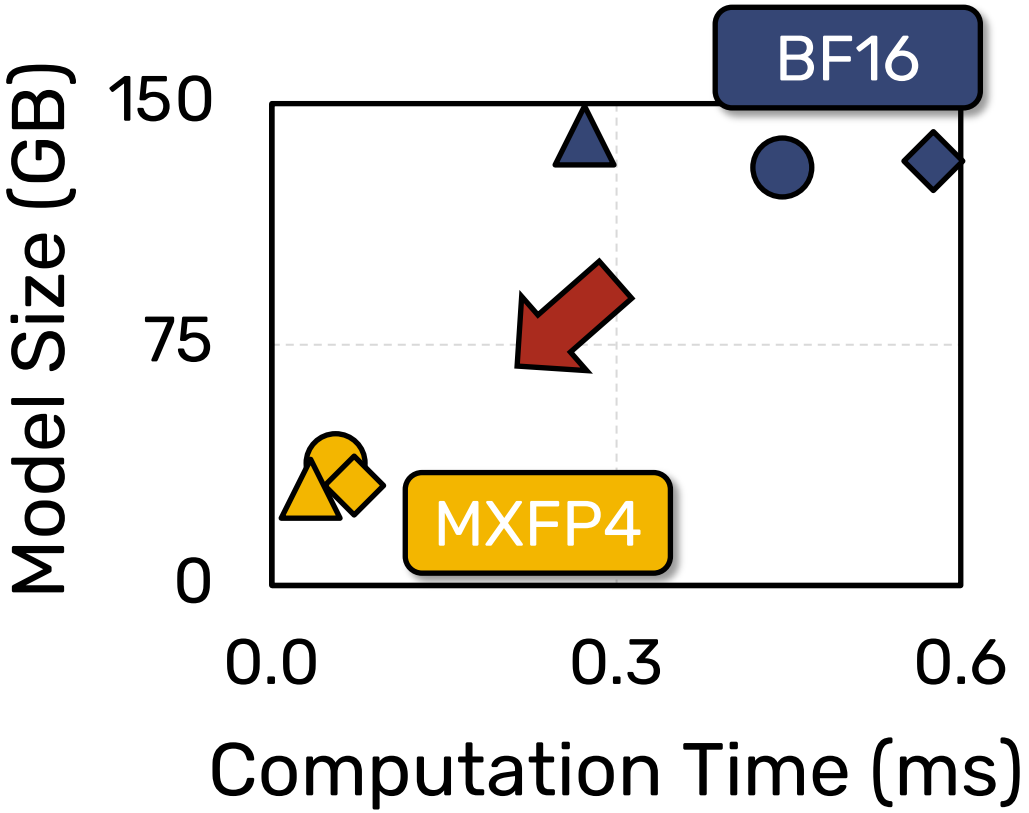
Standardized by the Industry *

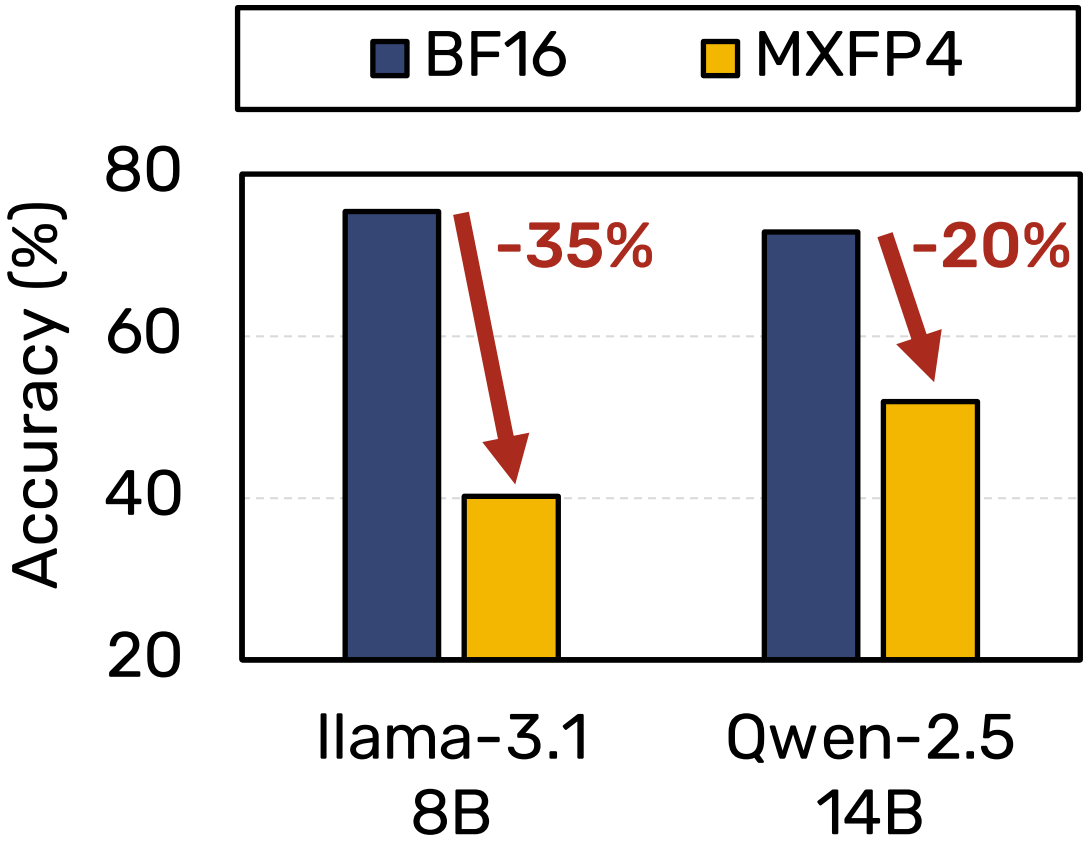
** AMD, Arm, Intel, Meta, Microsoft, NVIDIA, and Qualcomm*

Microscaling (MX) Formats



Implications on LLM Serving

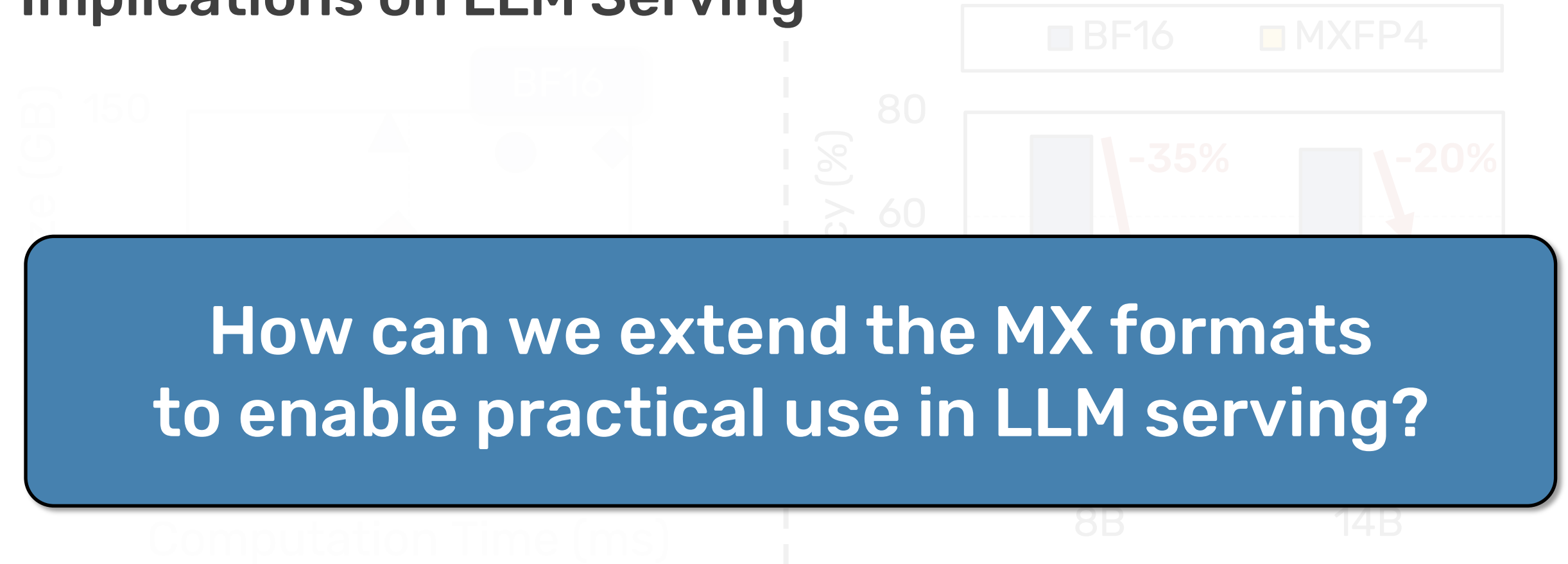




Reduce Compute & Memory Overheads ☺

Low Accuracy ☹

Implications on LLM Serving

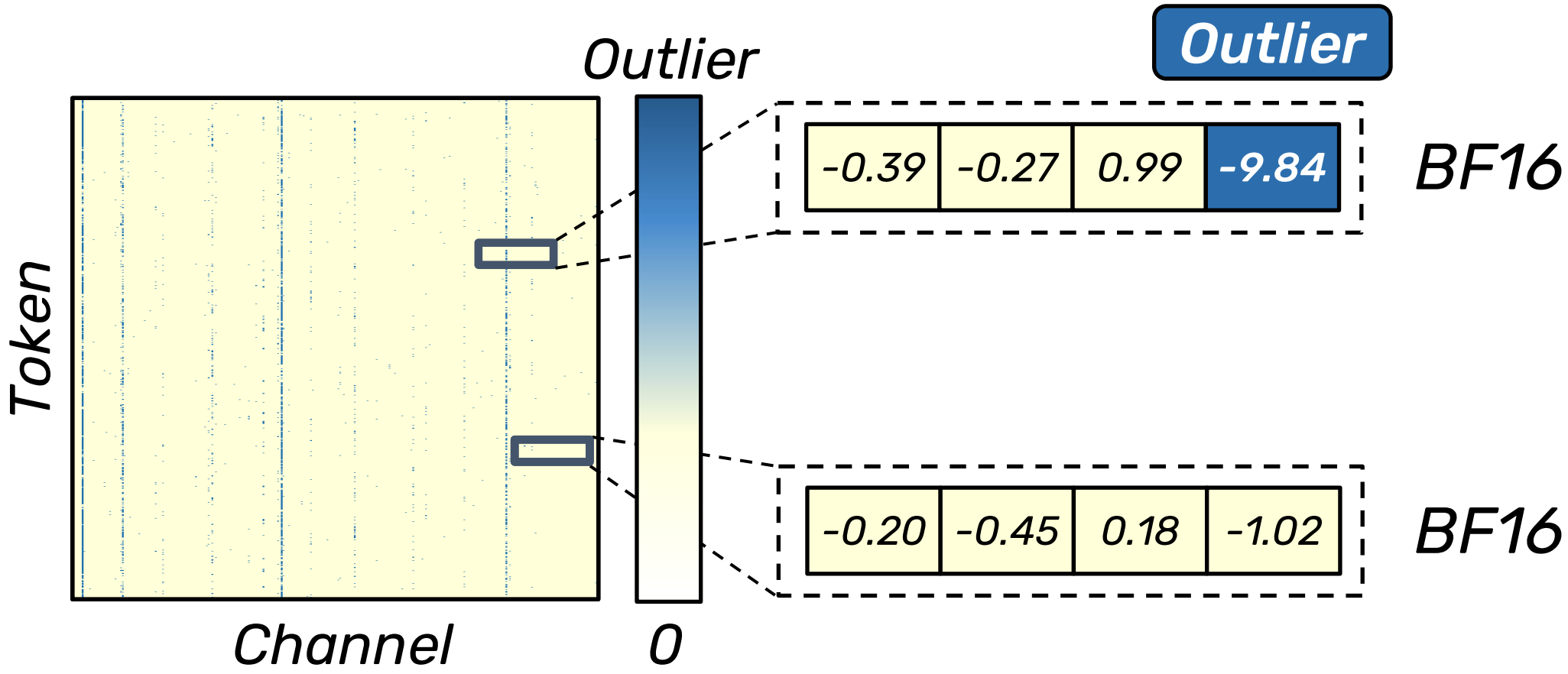


How can we extend the MX formats to enable practical use in LLM serving?

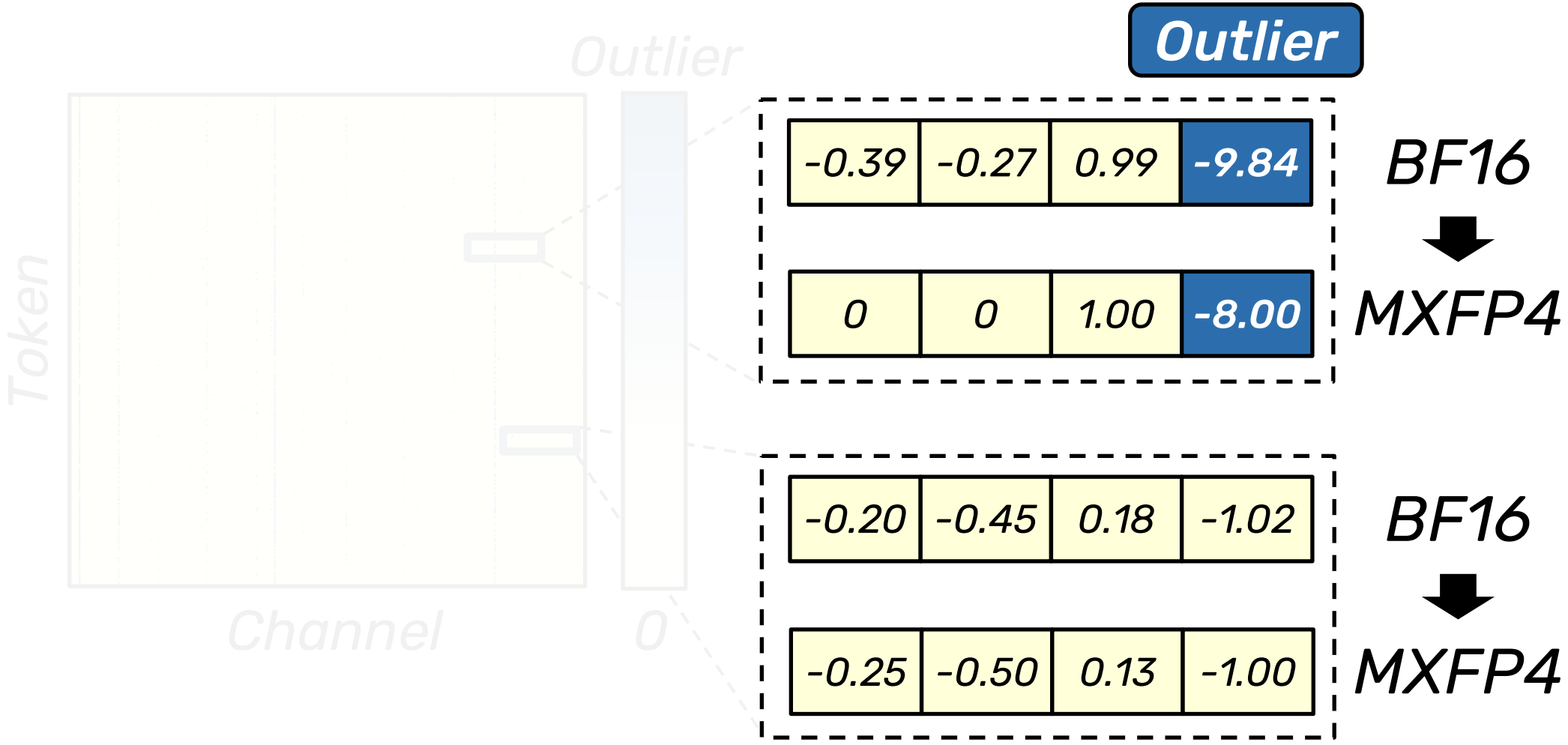
Reduce Compute & Memory Overheads ☺

Low Accuracy ☹

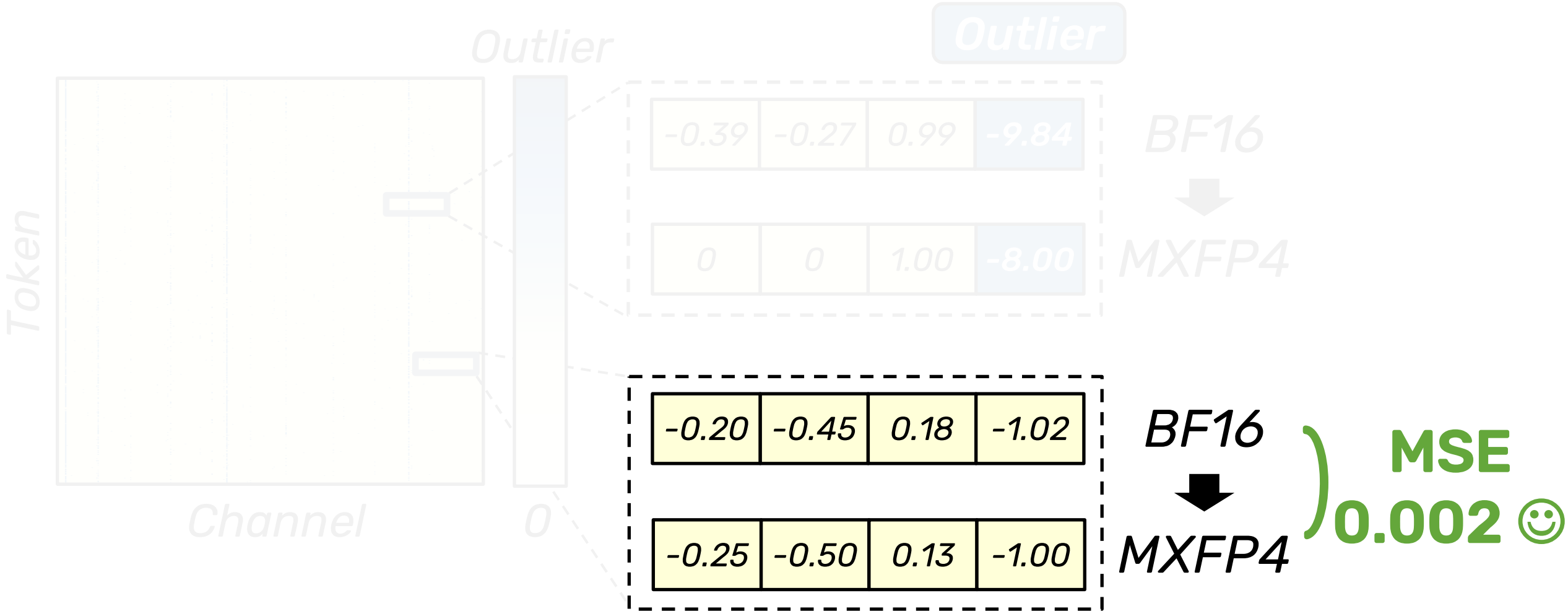
Analysis on Low-Bit MX Format



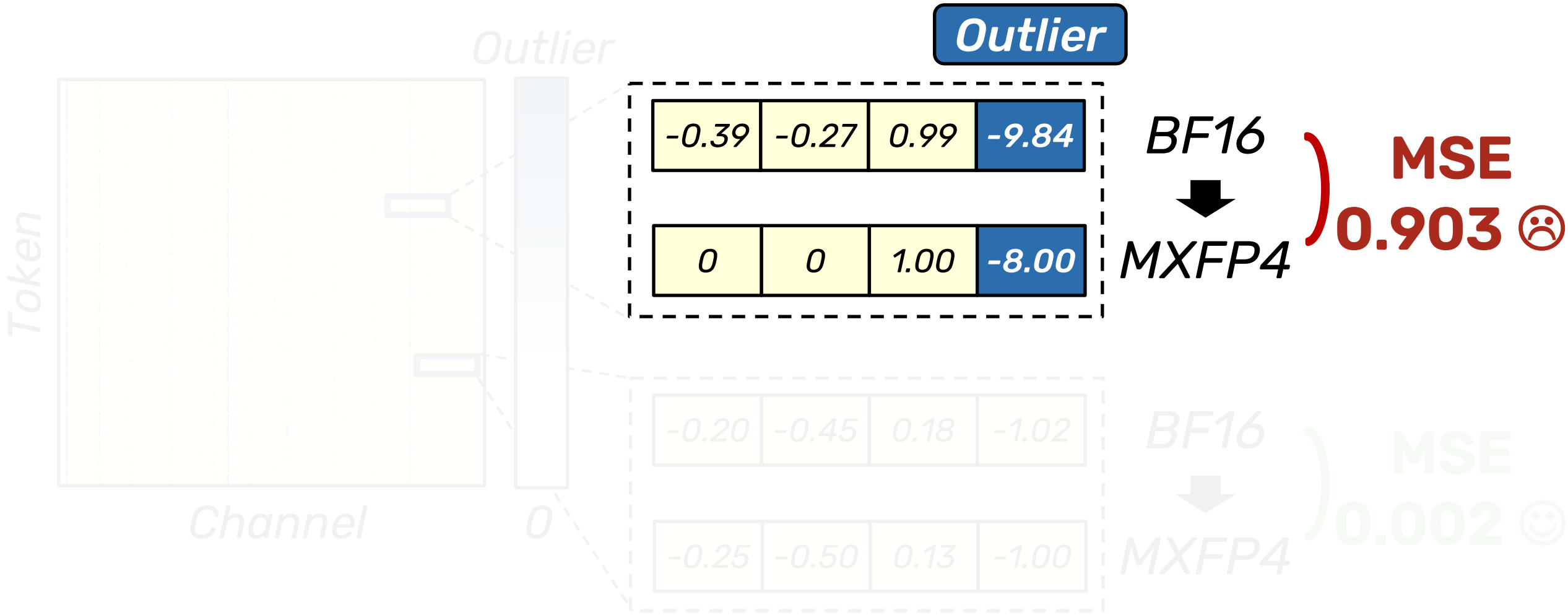
Analysis on Low-Bit MX Format



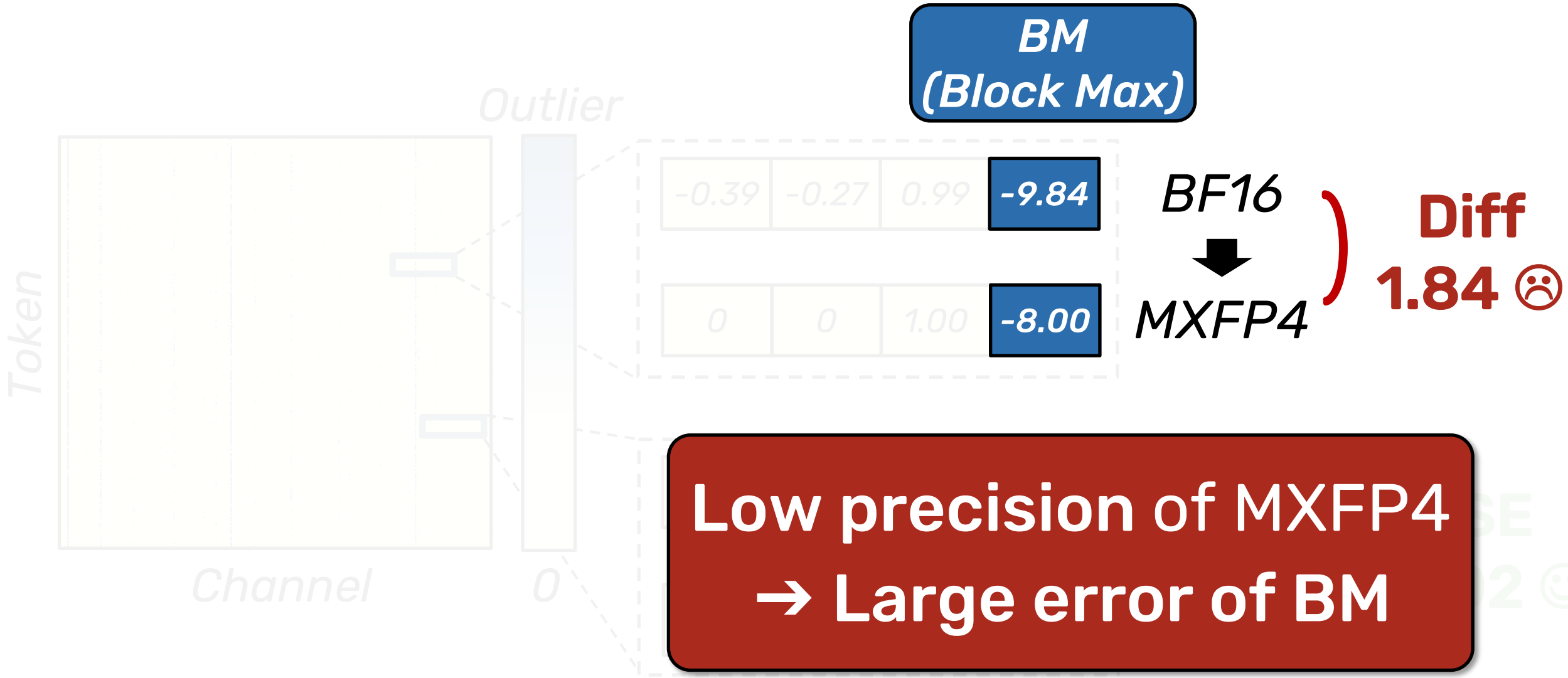
Analysis on Low-Bit MX Format



Analysis on Low-Bit MX Format

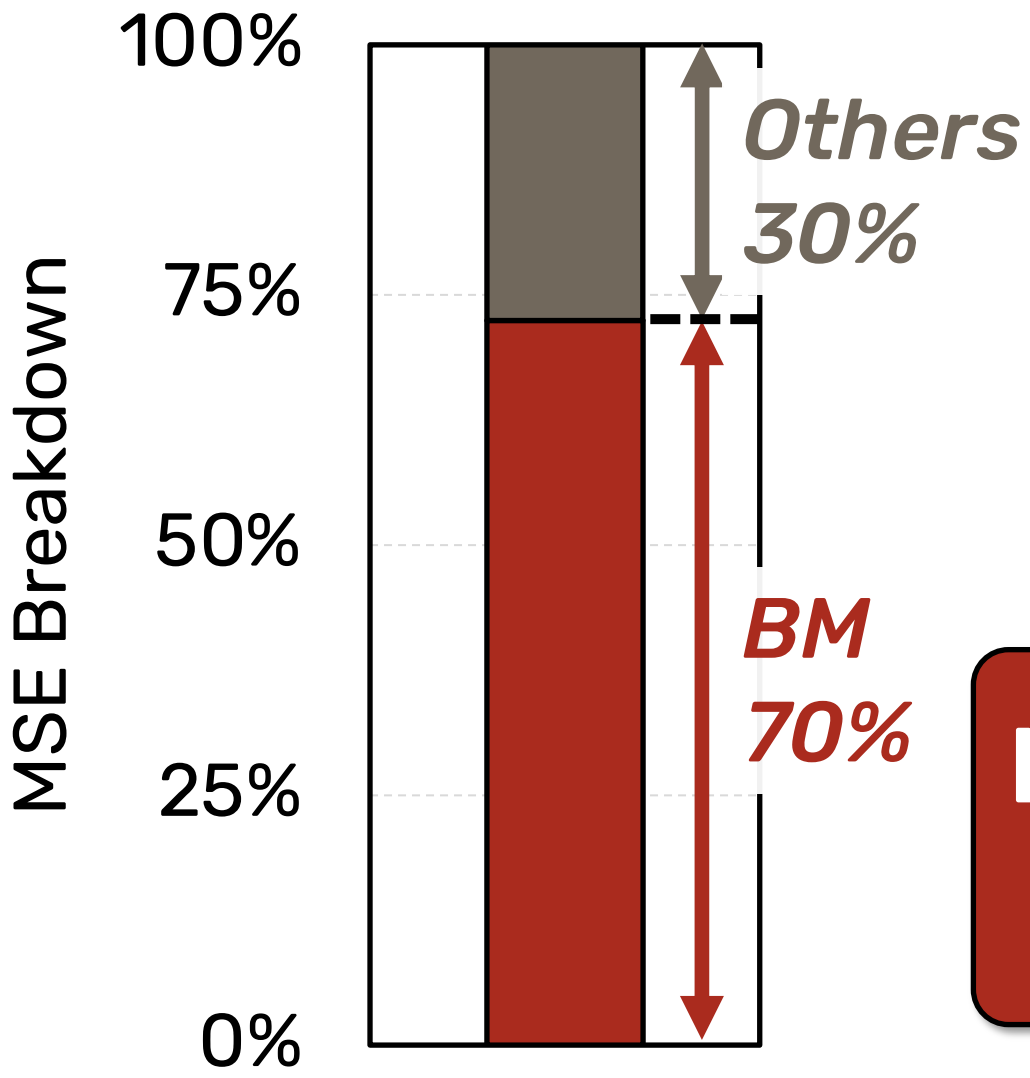


Analysis on Low-Bit MX Format



Low precision of MXFP4
→ **Large error of BM**

Analysis on Low-Bit MX Format



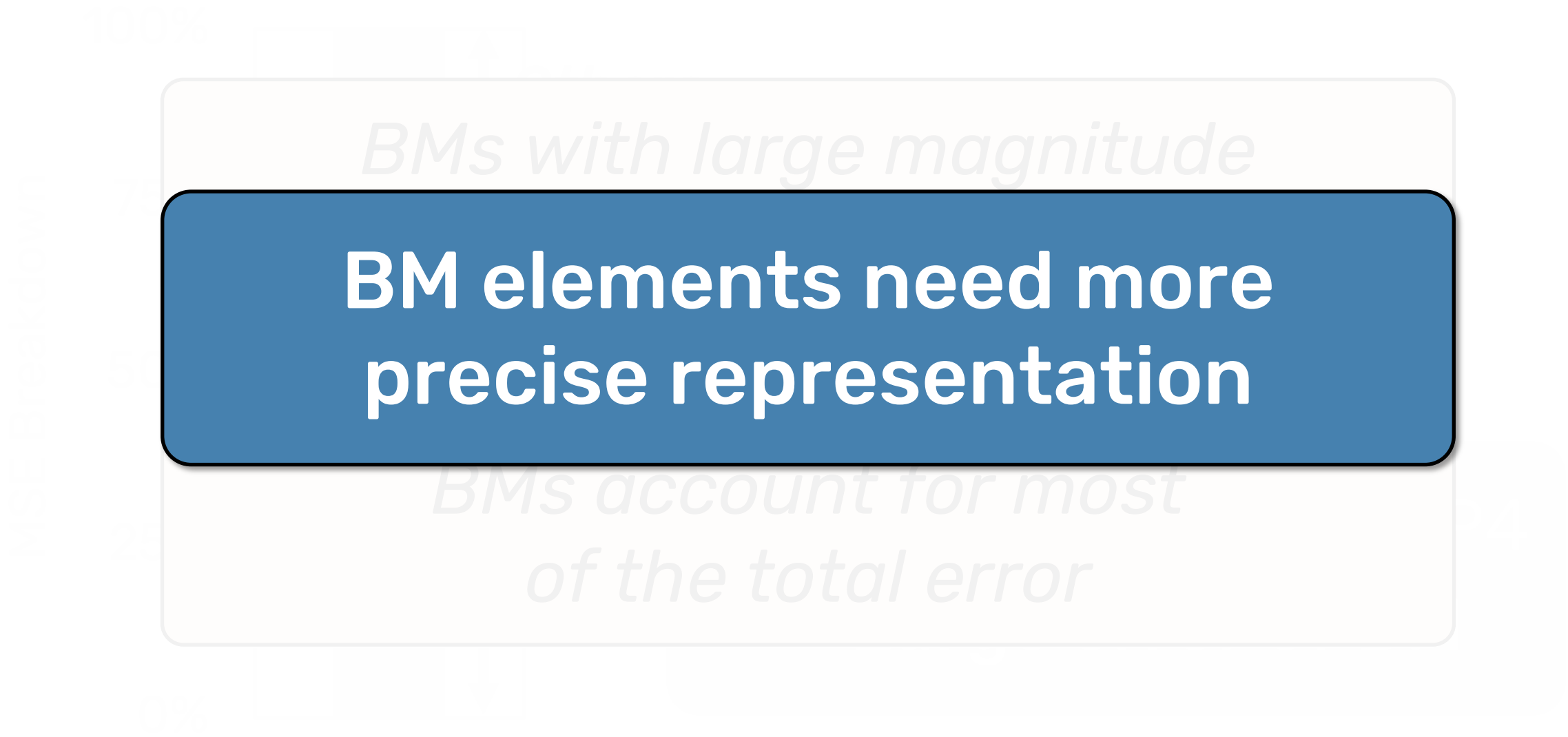
Low precision of MXFP4
→ **Large error of BM**

Analysis on Low-Bit MX Format

BMs with large magnitude show significant error

BMs account for most of the total error

Analysis on Low-Bit MX Format

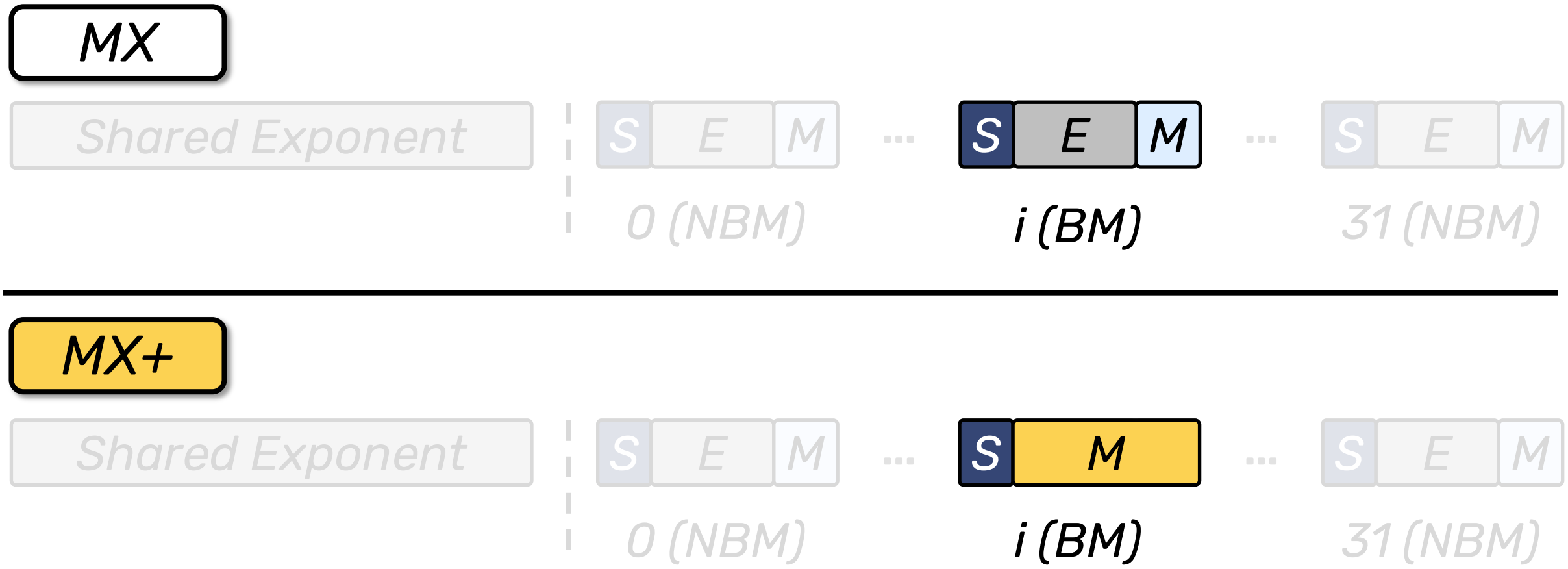


BM elements need more precise representation

Outline

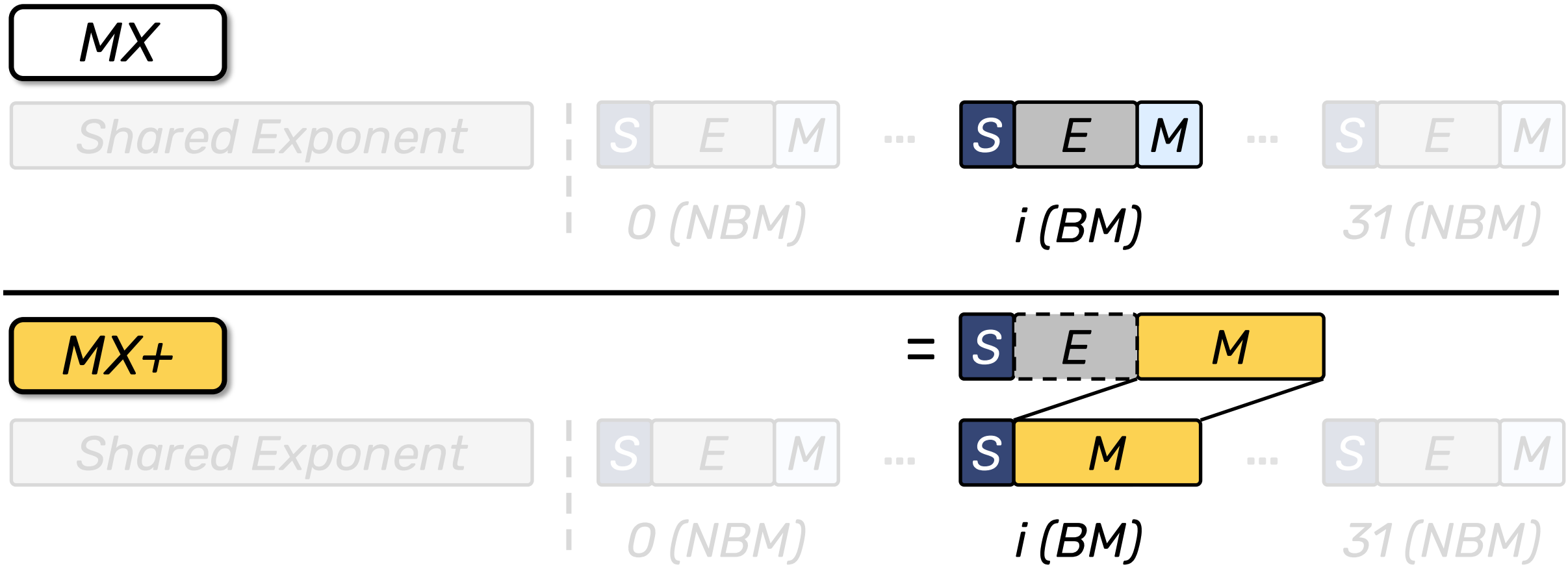
- **Introduction to MX Formats**
- **Motivation**
  - Implications for Low-bit LLM Serving
  - Analysis on Low-bit MX Formats
- ***MX+*: Cost-Effective and Non-intrusive Extension to MX Formats**
  - Key Insight
  - Software & Hardware Integration on GPUs
- **Evaluation**
- **Conclusion**

MX+ Overview



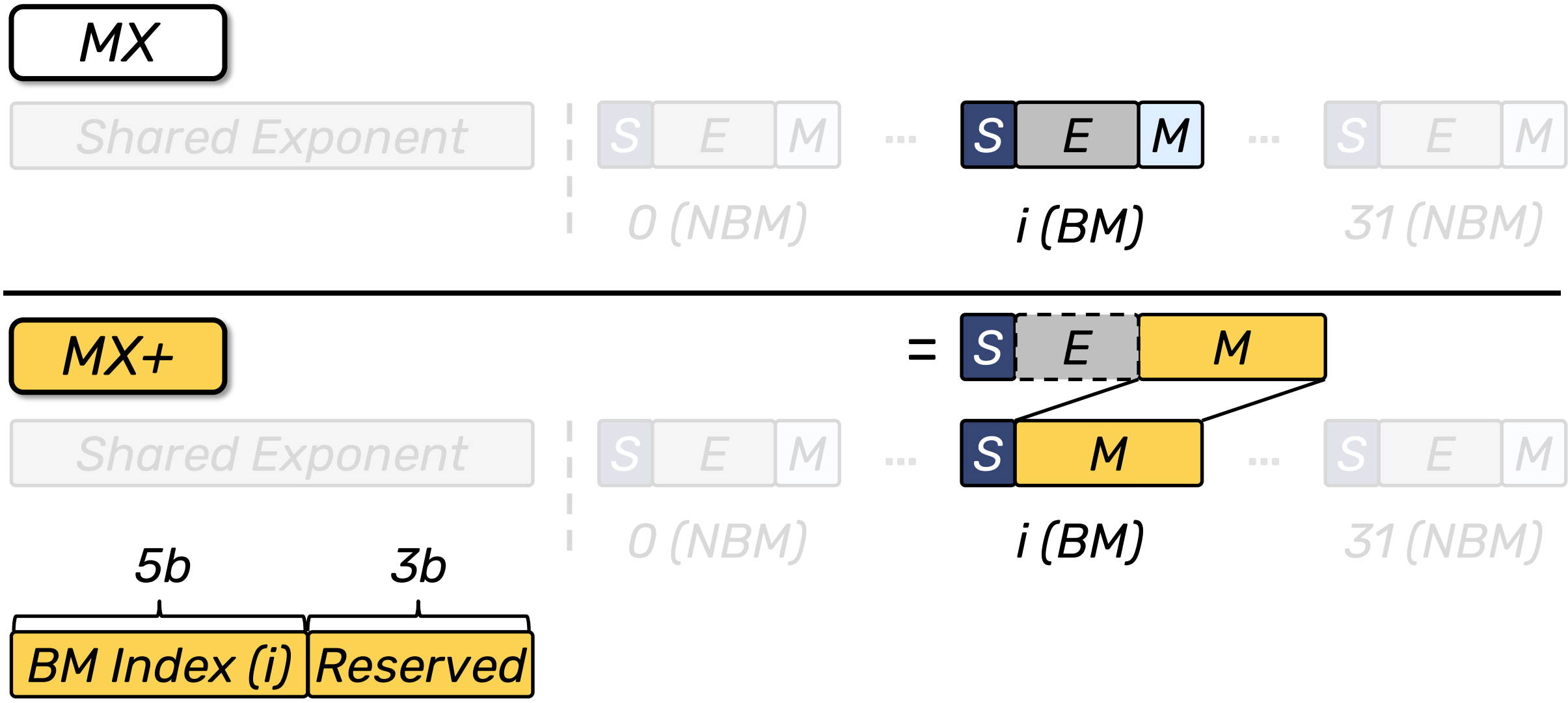
BM represented in high-precision ☺

MX+ Overview



BM represented in high-precision ☺

MX+ Overview

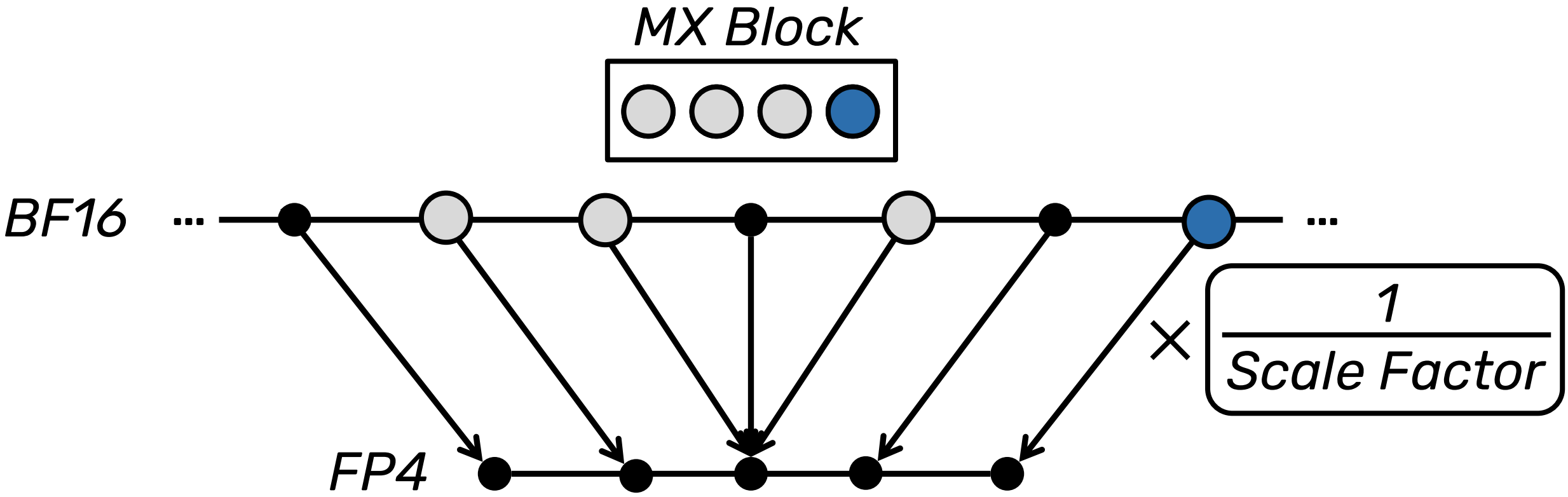

MX
Shared Exponent
S E M
...
S E M
...
S E M
0 (NBM)
i (BM)
31 (NBM)
MX+
=
S E M
S M
Shared Exponent
S E M
...
S M
...
S E M
0 (NBM)
i (BM)
31 (NBM)
5b
3b
BM Index (i)
Reserved

Key Insight

Exponent of BM is set to the max value

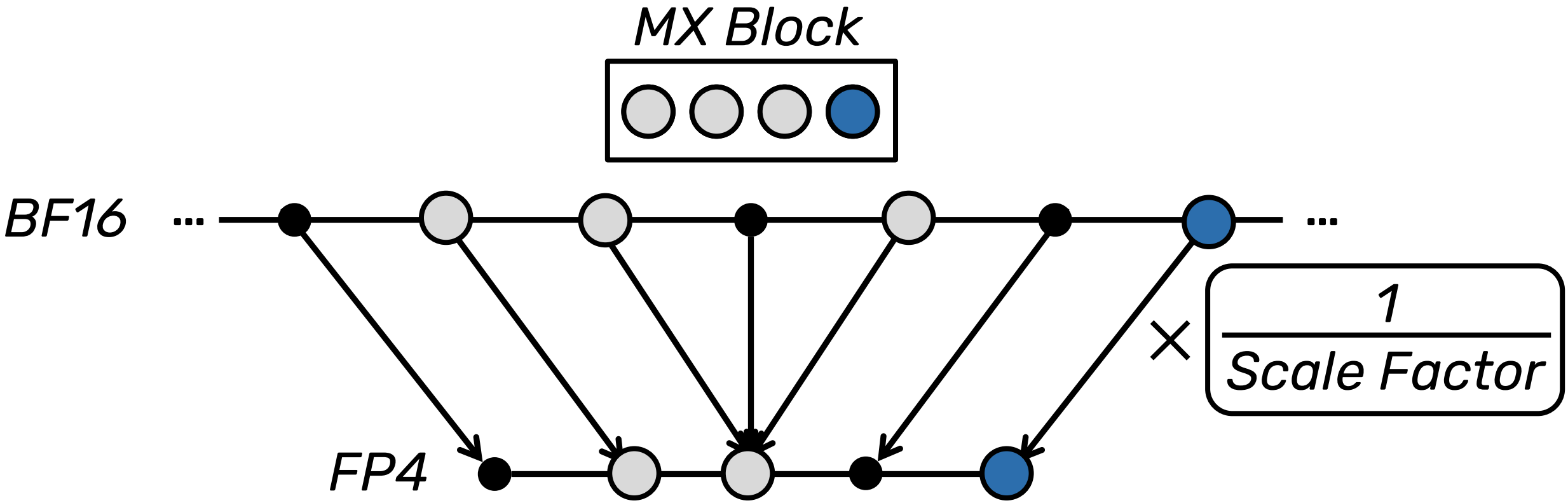
Key Insight

Exponent of BM is set to the max value



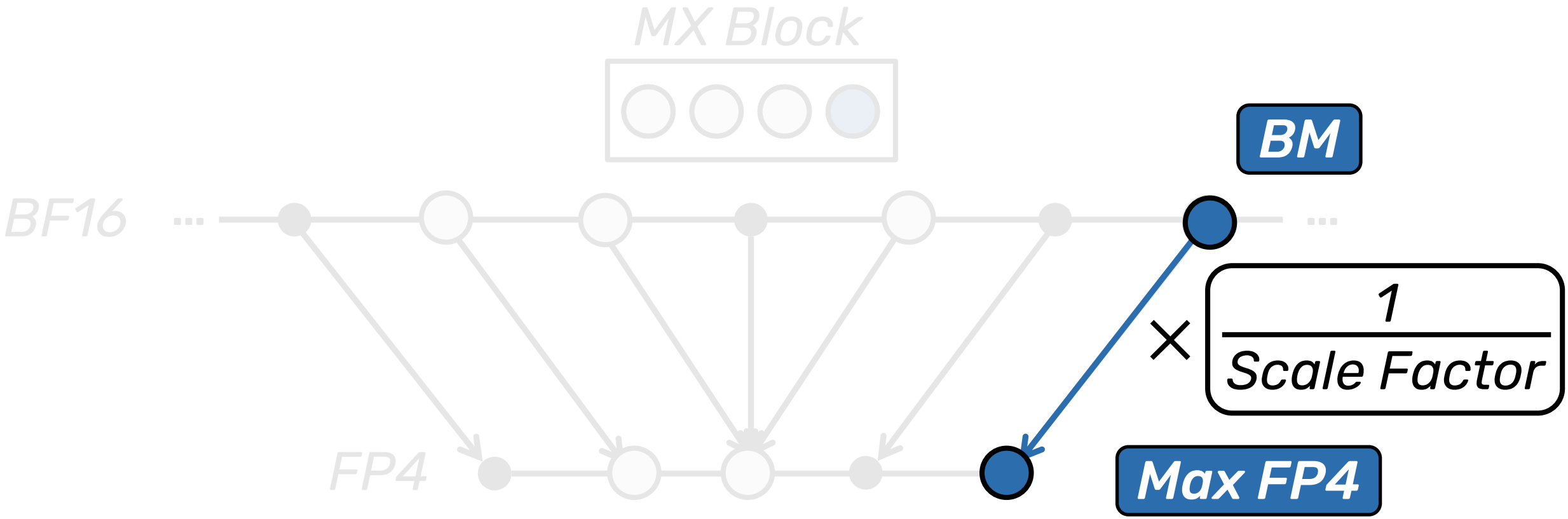
Key Insight

Exponent of BM is set to the max value



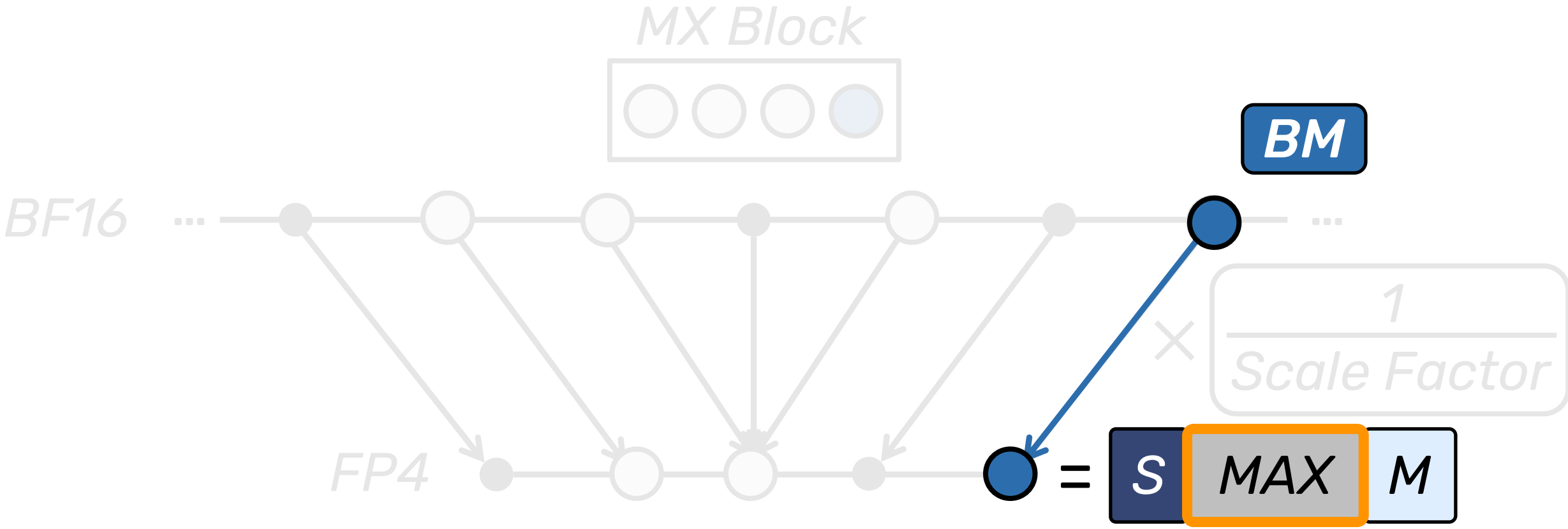
Key Insight

Exponent of BM is set to the max value

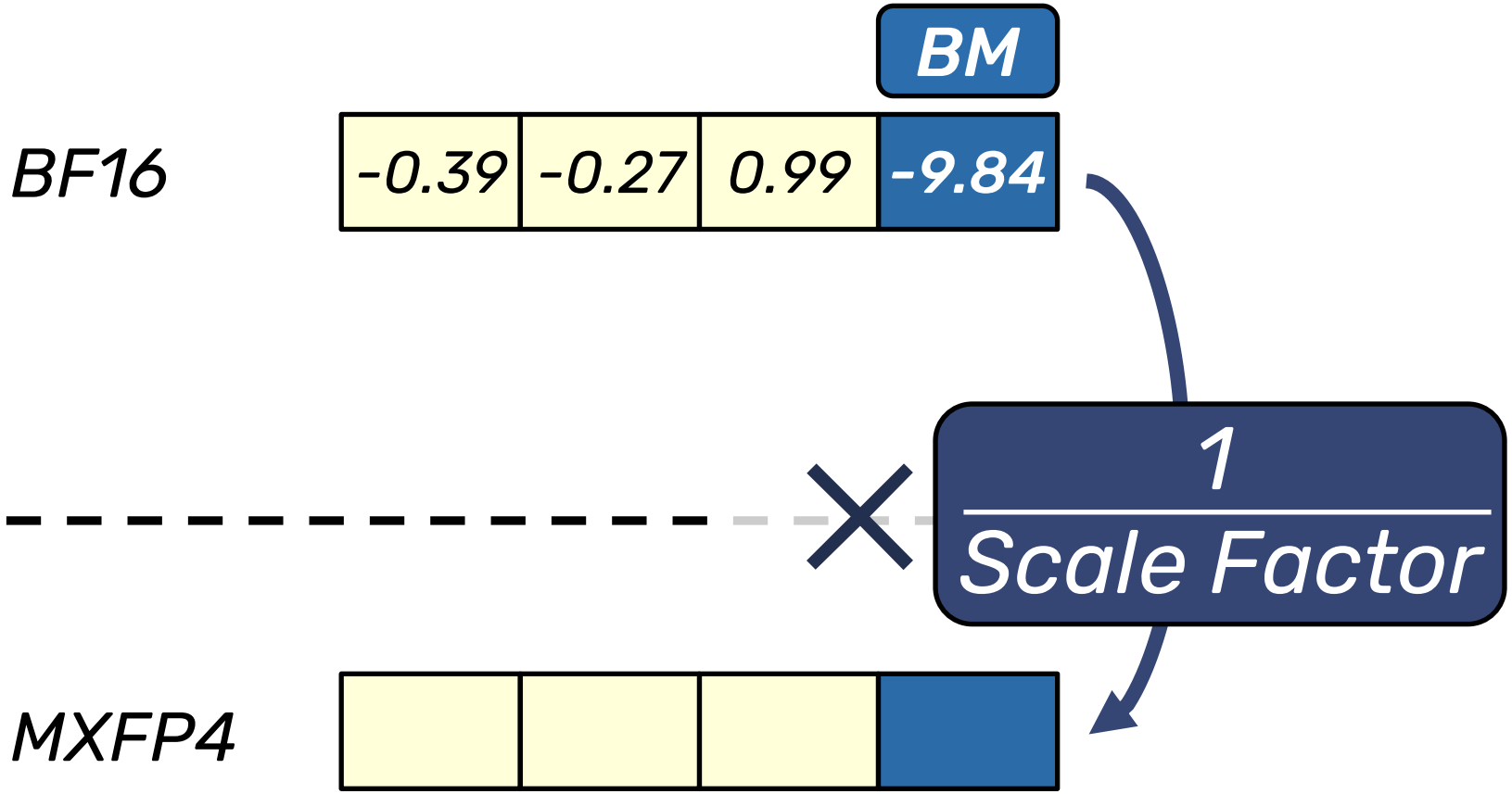


Key Insight

Exponent of BM is set to the max value



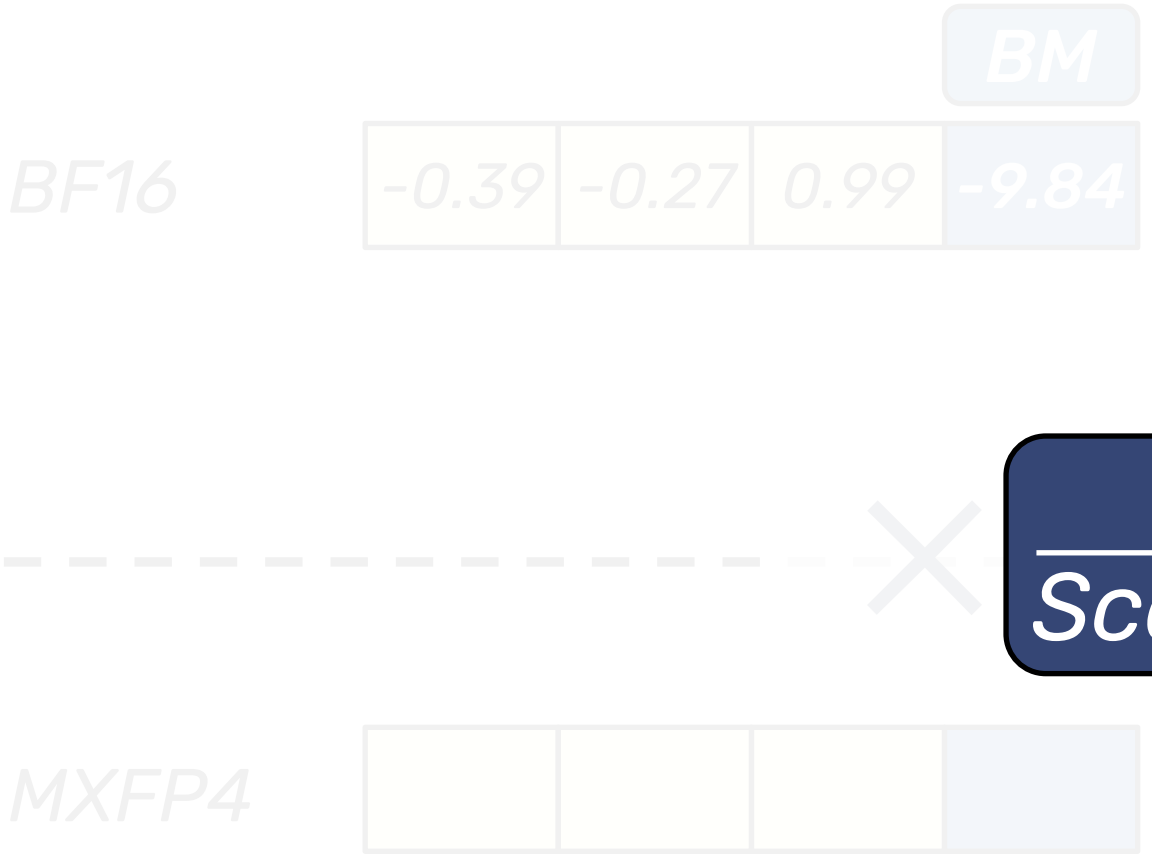
Conversion to MXFP4



Conversion to MXFP4

$$\frac{1}{\text{Scale Factor}} = \frac{2^{\text{Max Exp}}}{2^{\text{BM Exp}}}$$

Conversion to MXFP4



FP4 Max Exp = 2

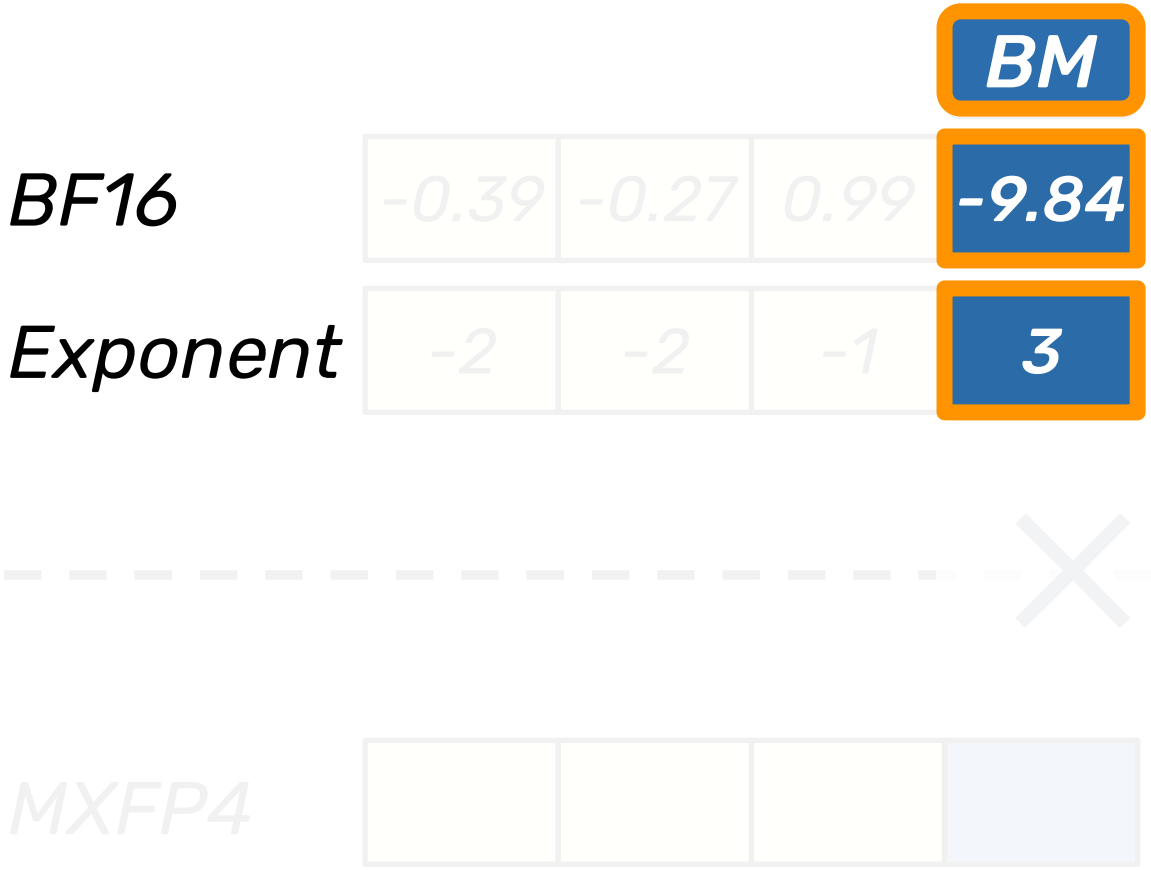
$$\frac{1}{\text{Scale Factor}} = \frac{2^{2}}{2^{\text{BM Exp}}}$$

Conversion to MXFP4

FP4 Max Exp = 2

$$\frac{1}{\text{Scale Factor}} = \frac{2^{2}}{2^{\text{BM Exp}}}$$

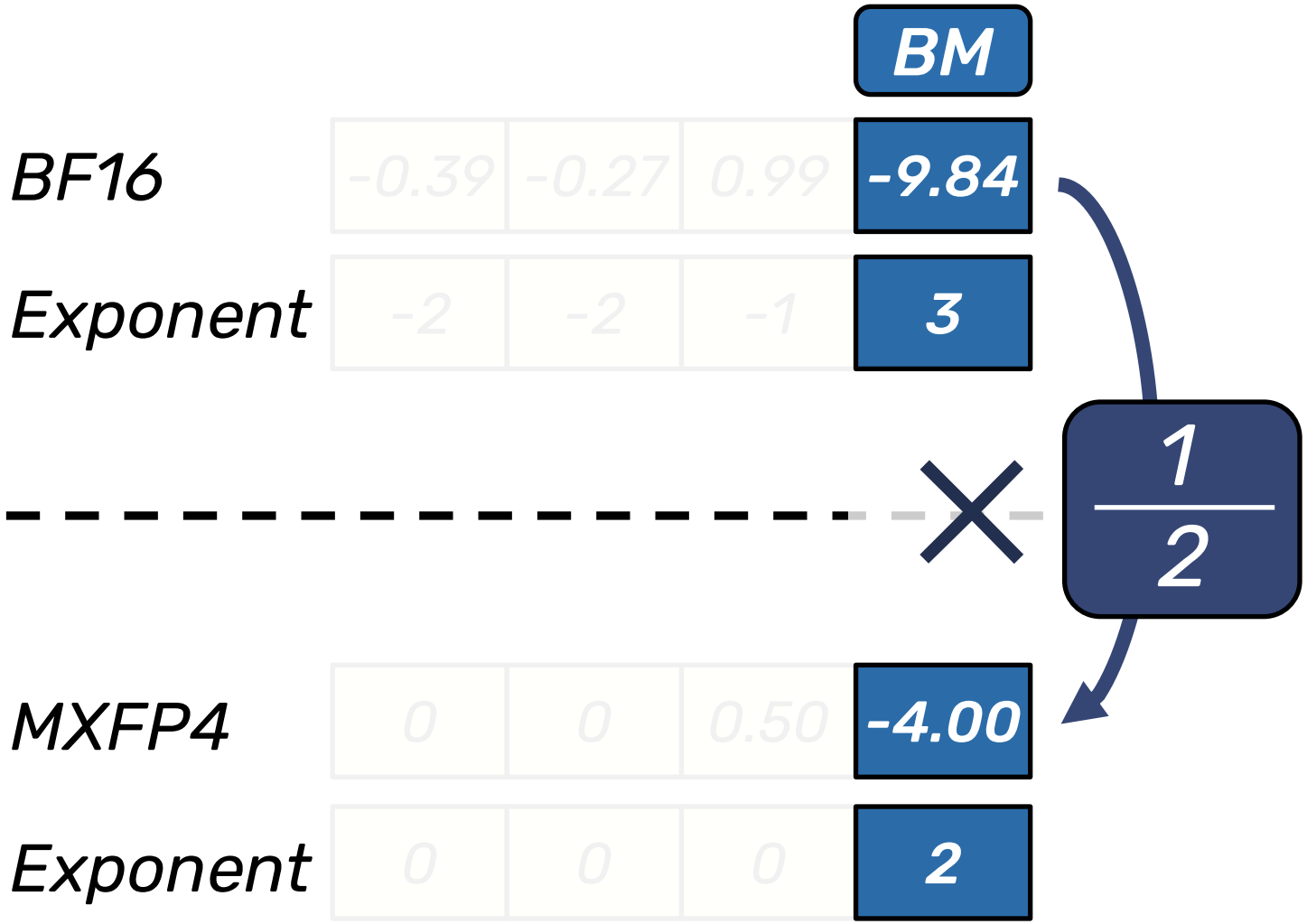
Conversion to MXFP4



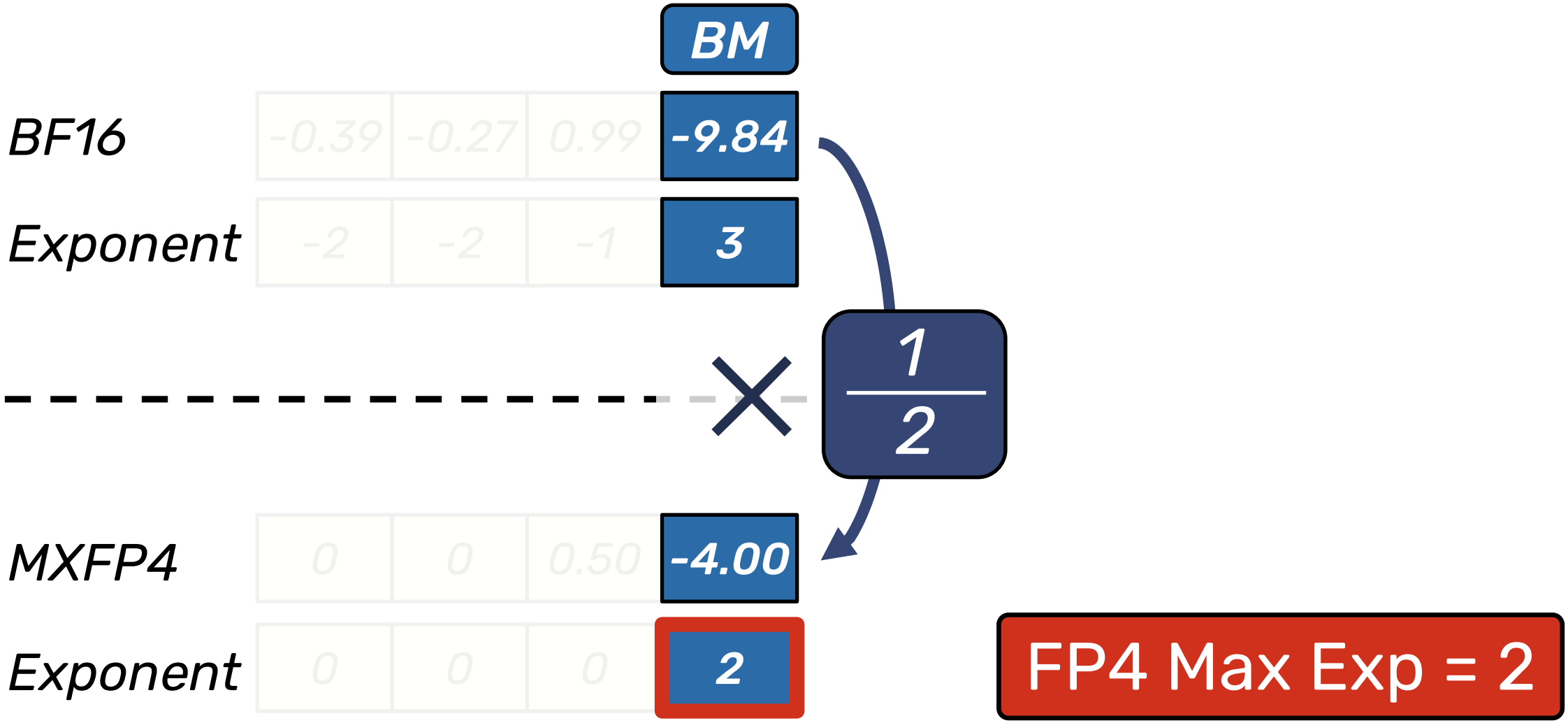
FP4 Max Exp = 2

$$= \frac{2^{2}}{2^{3}}$$

Conversion to MXFP4



Conversion to MXFP4



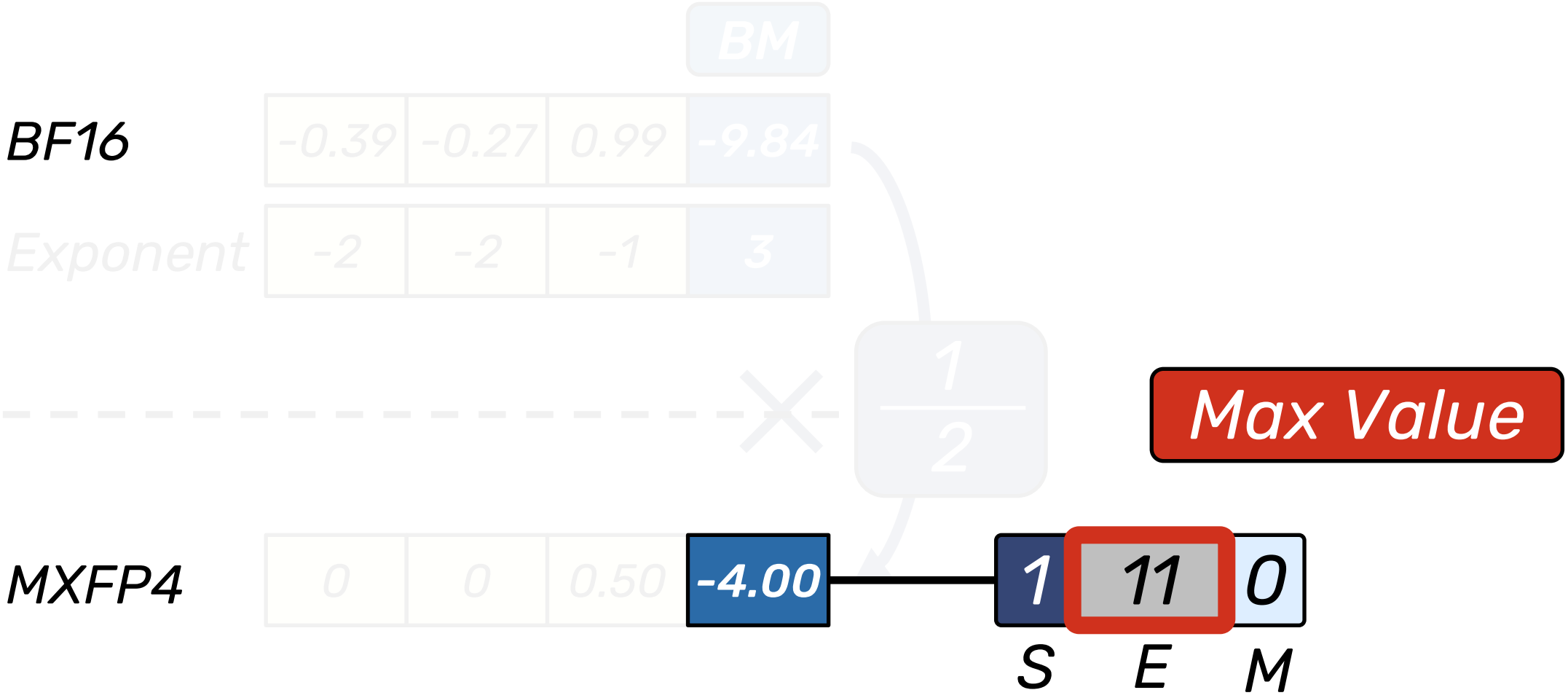
Conversion to MXFP4

				BM
BF16	-0.39	-0.27	0.99	-9.84
Exponent	-2	-2	-1	3
MXFP4	0	0	0.50	-4.00
Exponent	0	0	0	2

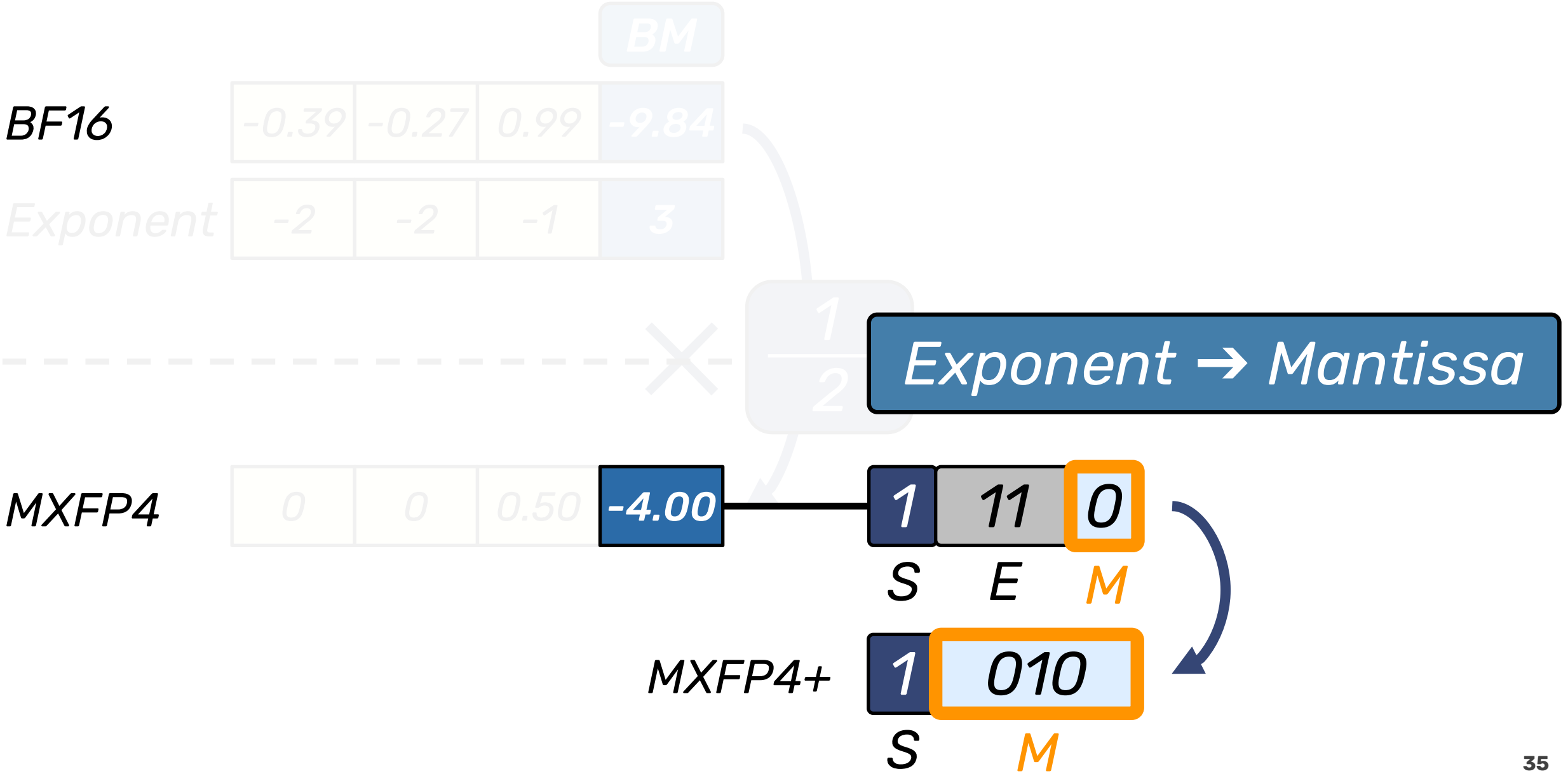
Store mantissa bits instead of exponent

FP4 Max Exp = 2

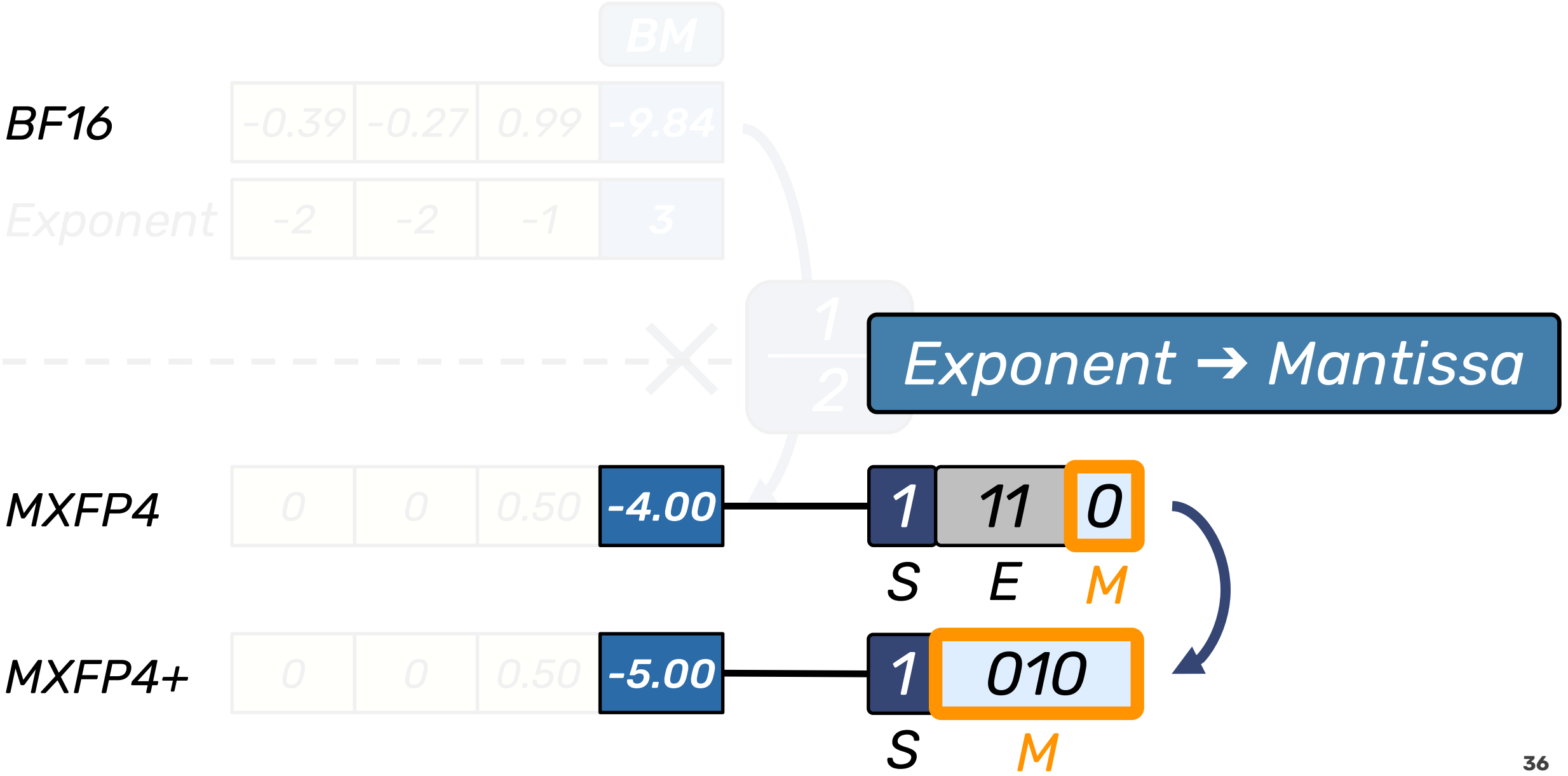
MX+ Extension



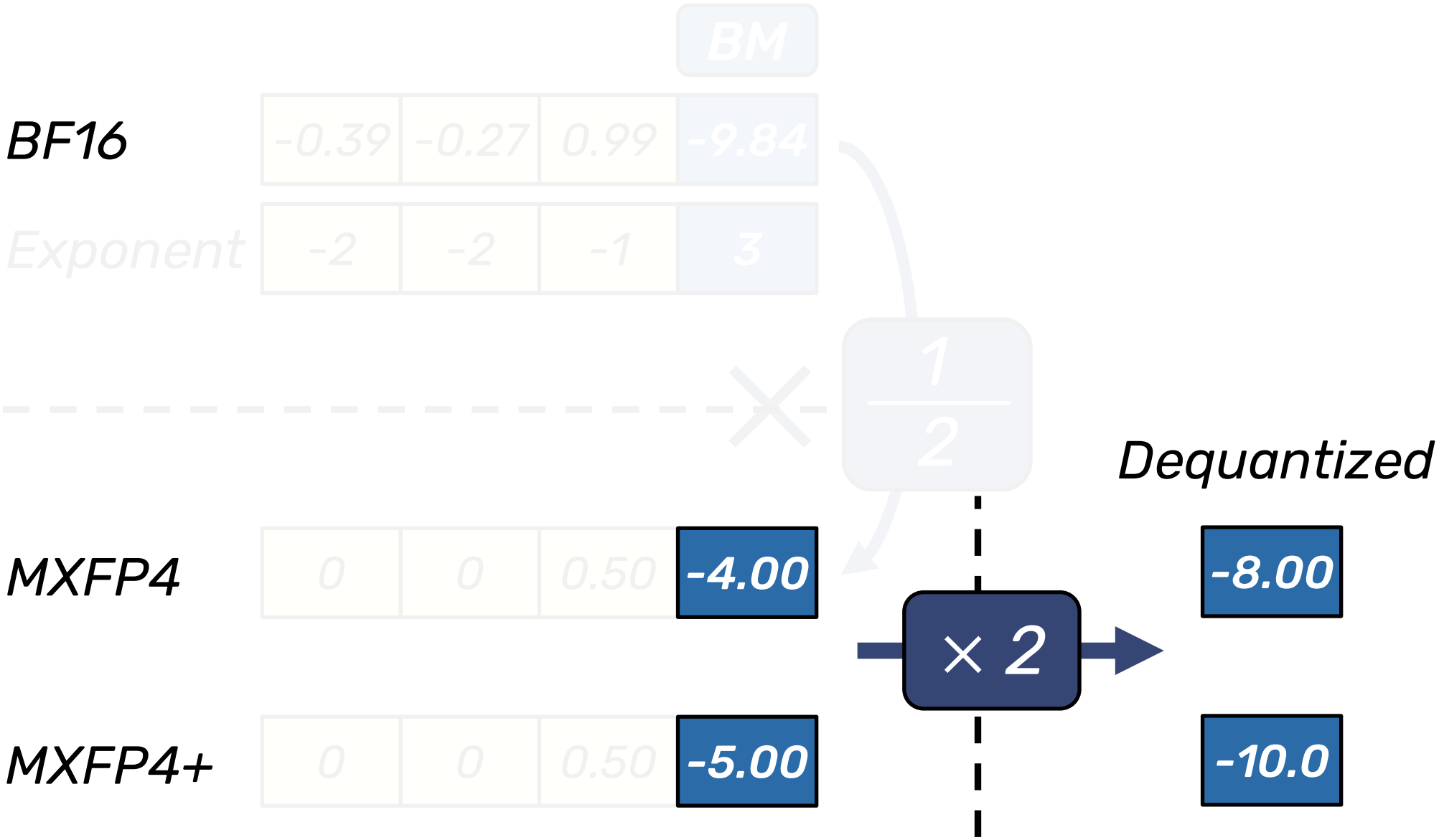
MX+ Extension



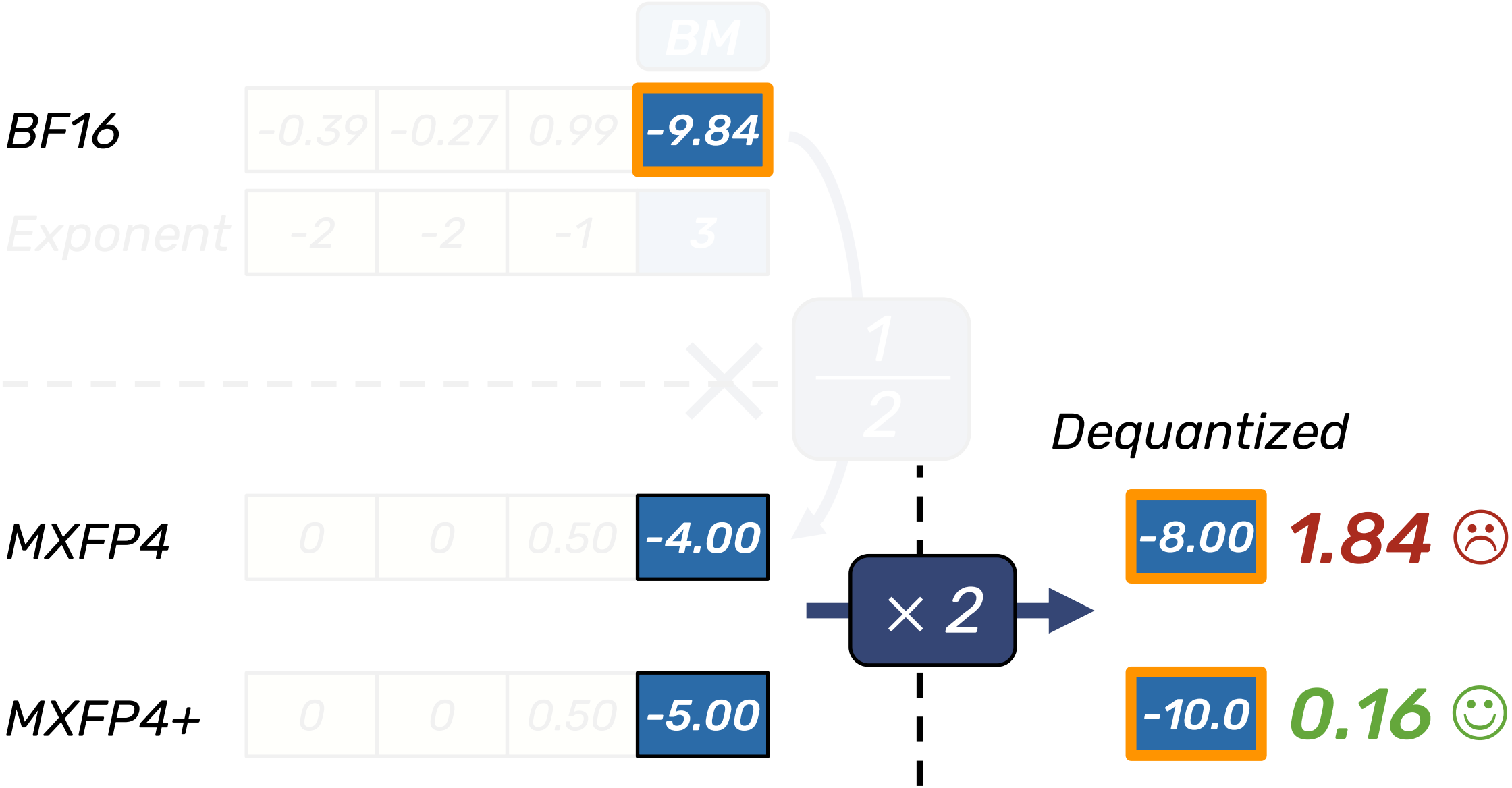
MX+ Extension



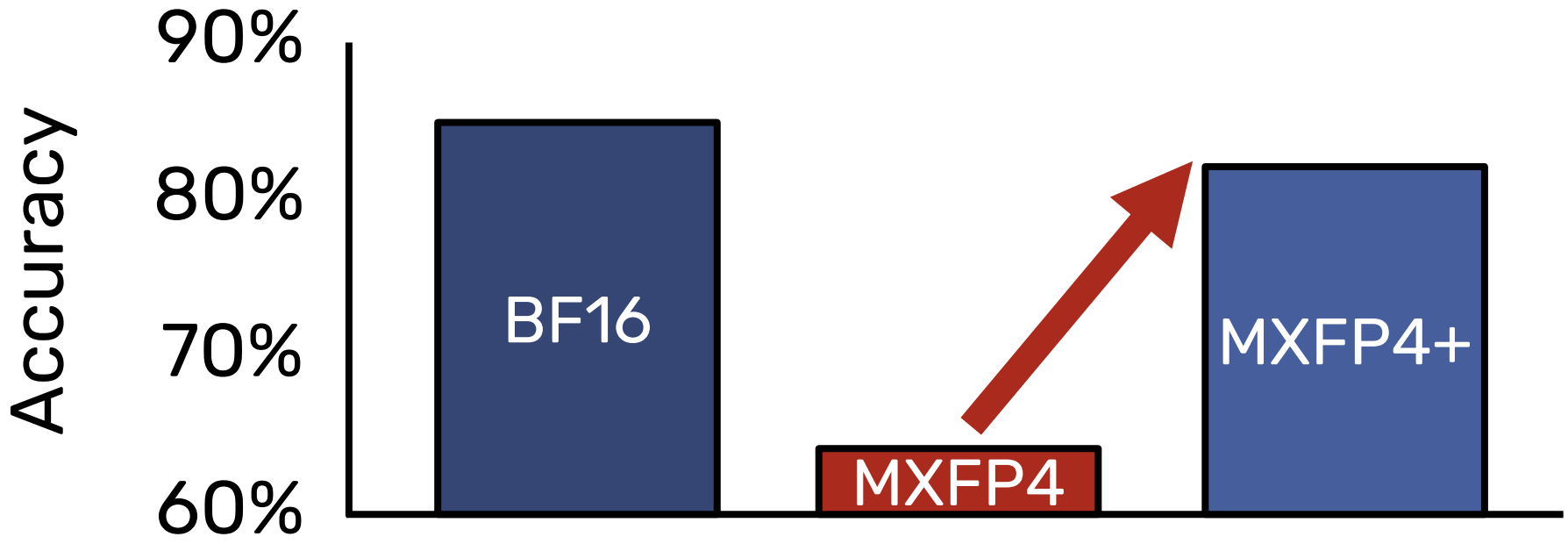
MX+ Extension



MX+ Extension



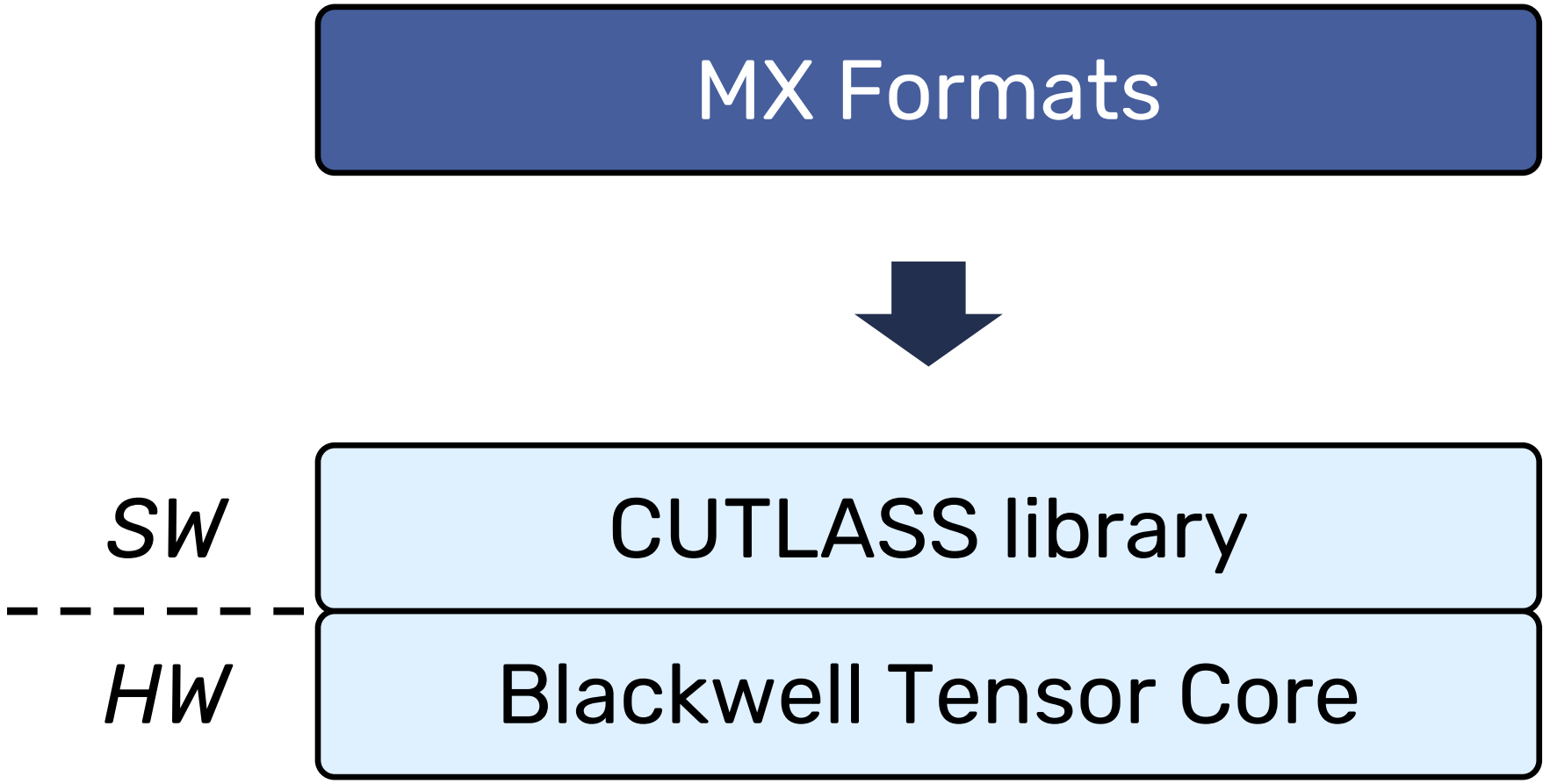
MX+ Extension



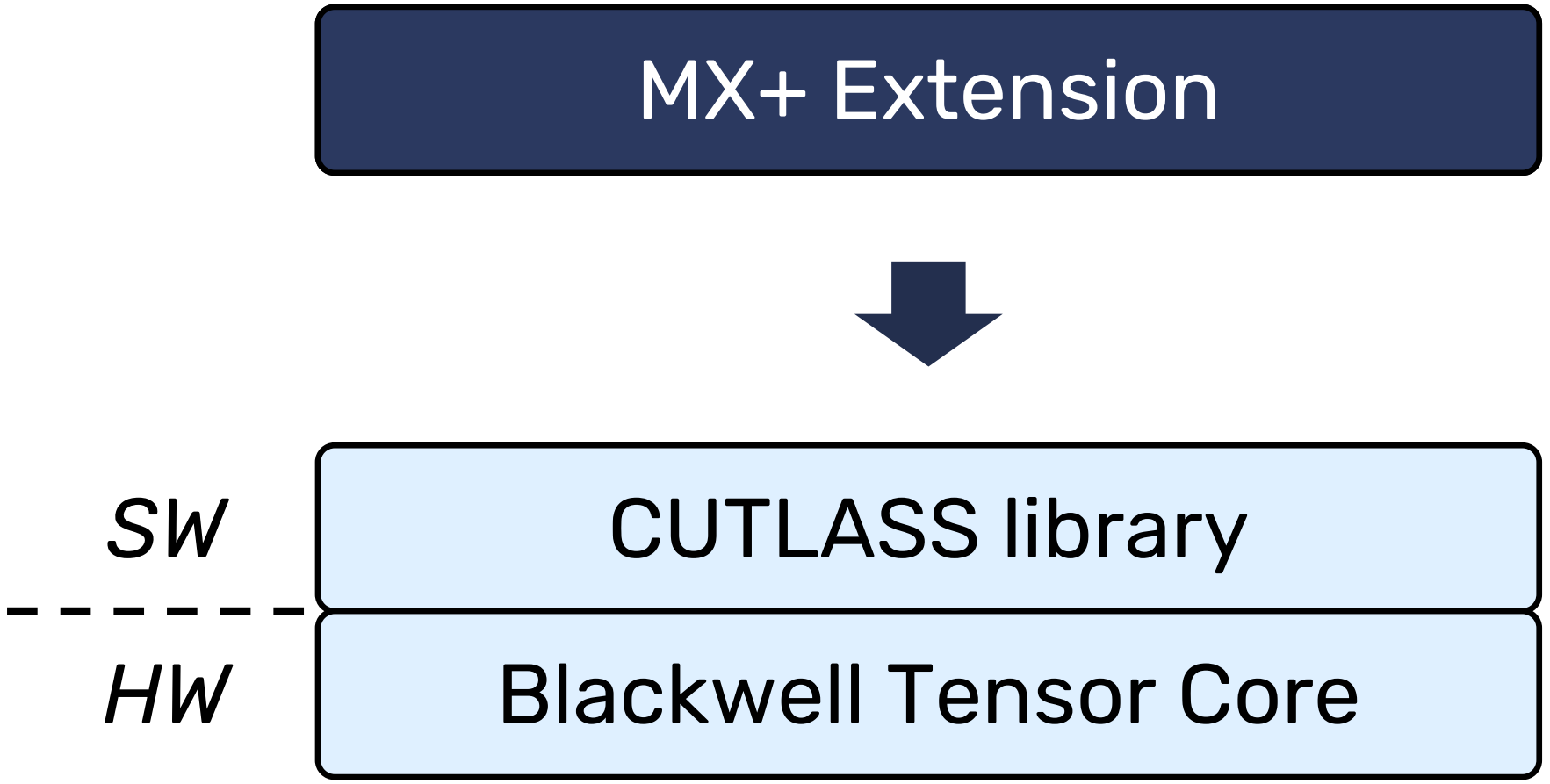
MXFP4 -8.00 1.84 ☹

MXFP4+ -10.0 0.16 ☺

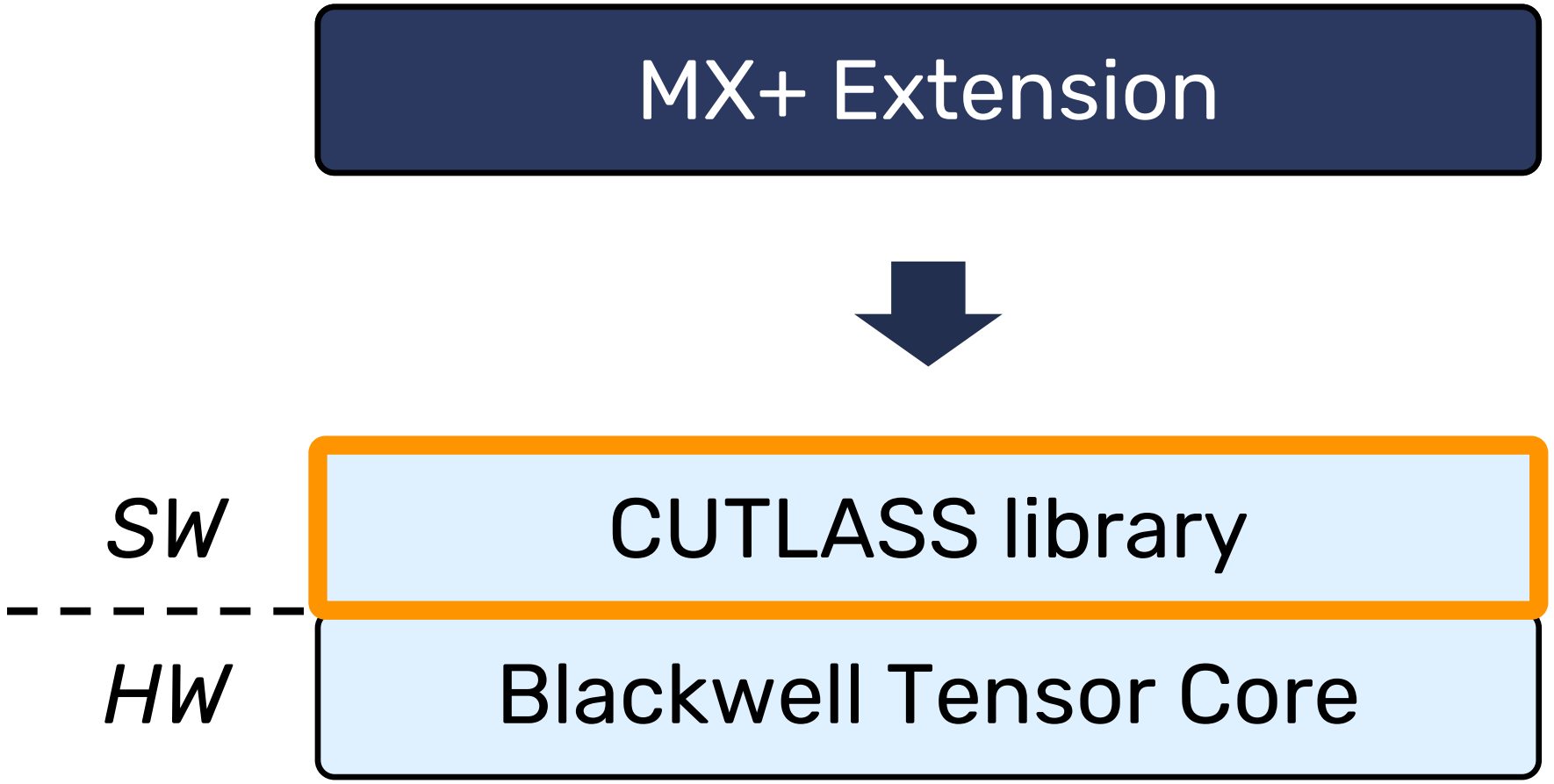
Integration of MX+ on GPUs



Integration of MX+ on GPUs



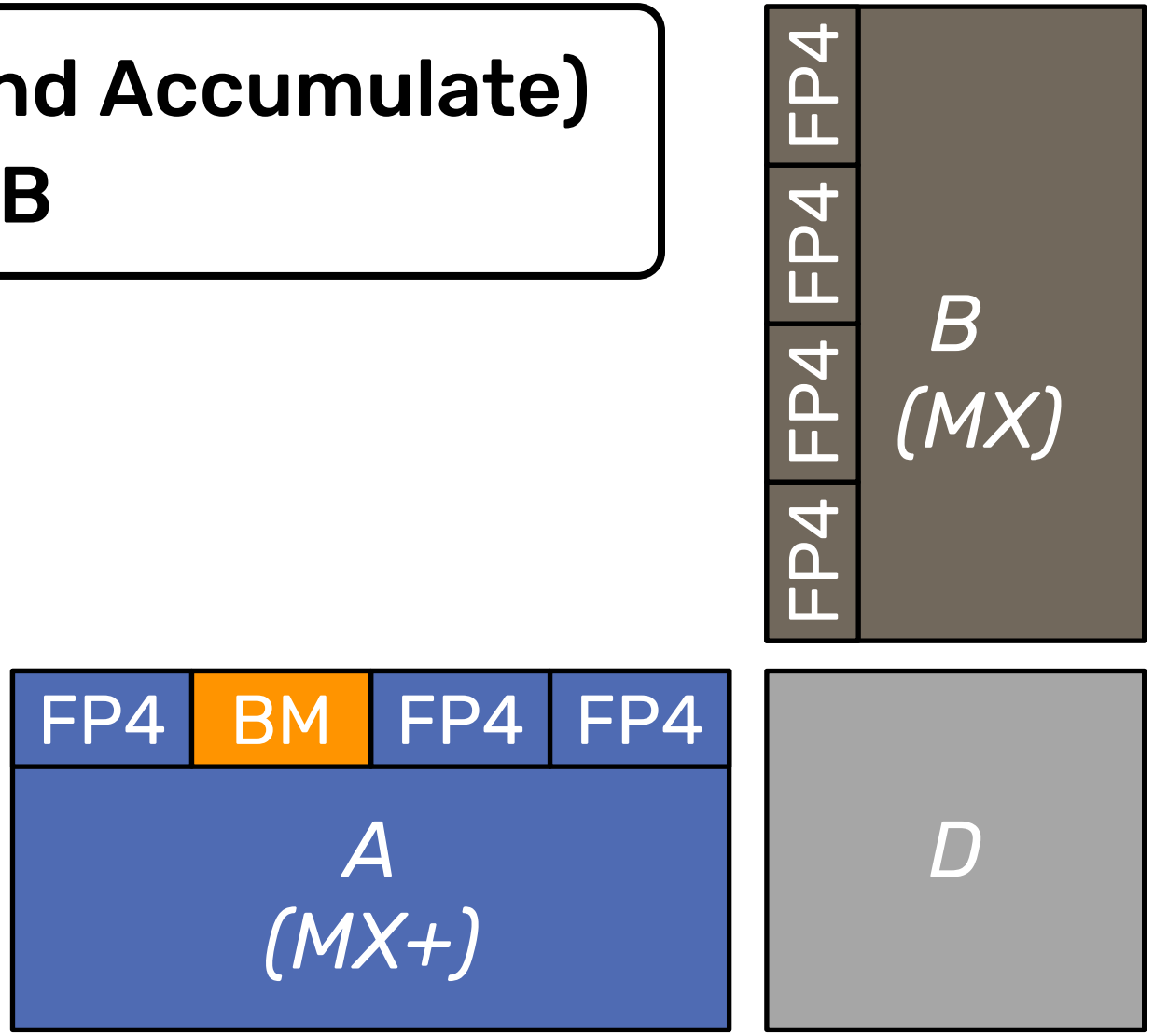
Integration of MX+ on GPUs



Software Integration of MX+ on GPUs

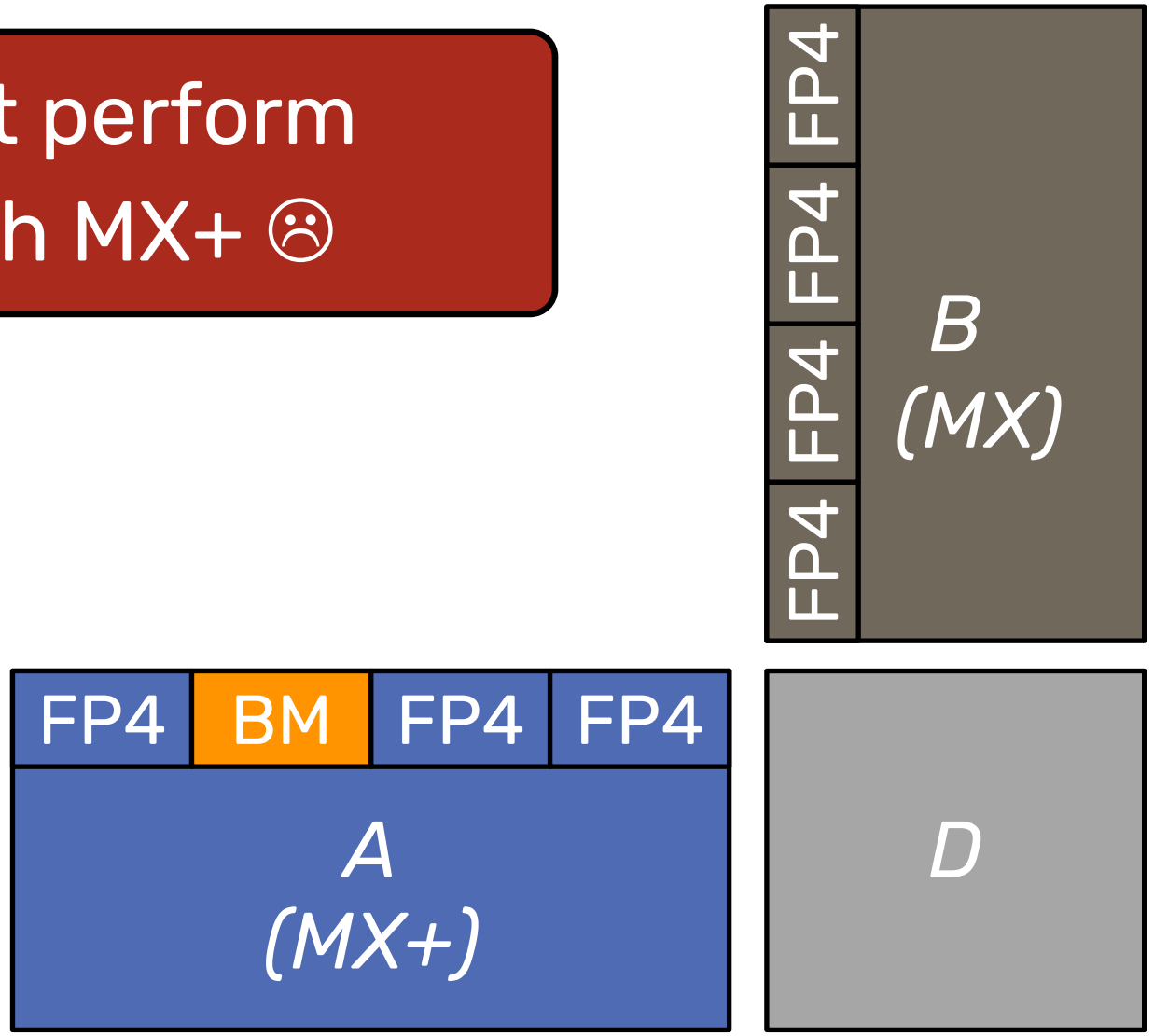
MMA (Matrix Multiply and Accumulate)

$$D \mathrel{+}= A \times B$$



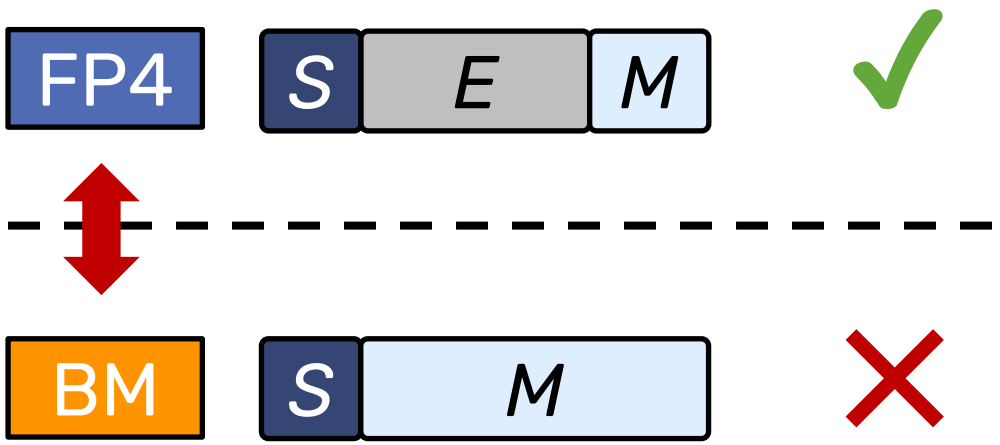
Software Integration of MX+ on GPUs

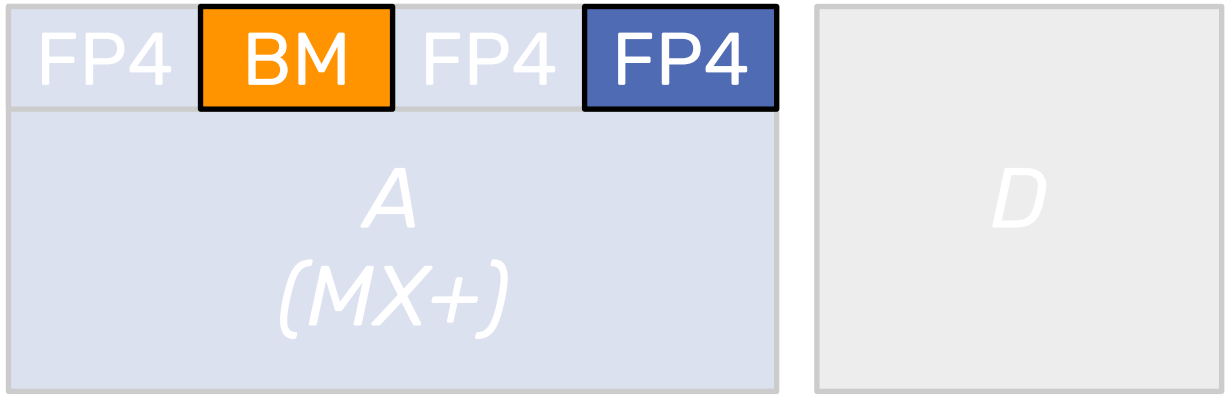
Tensor Core cannot perform MMA operation with MX+ ☹



Software Integration of MX+ on GPUs

Tensor Core cannot perform MMA operation with MX+ ☹





Software Integration of MX+ on GPUs

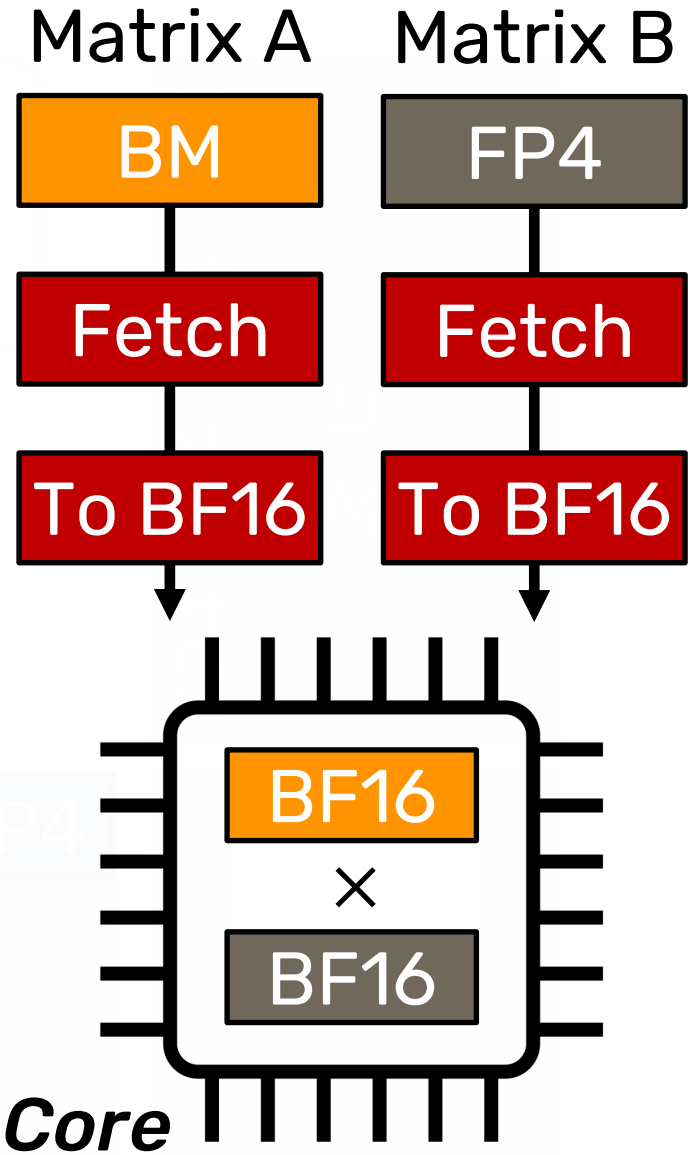
Tensor Core cannot perform MMA operation with MX+ ☹

➔ Compute BMs in CUDA Core?

Software Integration of MX+ on GPUs

Tensor Core cannot perform MMA operation with MX+ ☹

➔ Compute BMs in CUDA Core? ✕



Software Integration of MX+ on GPUs

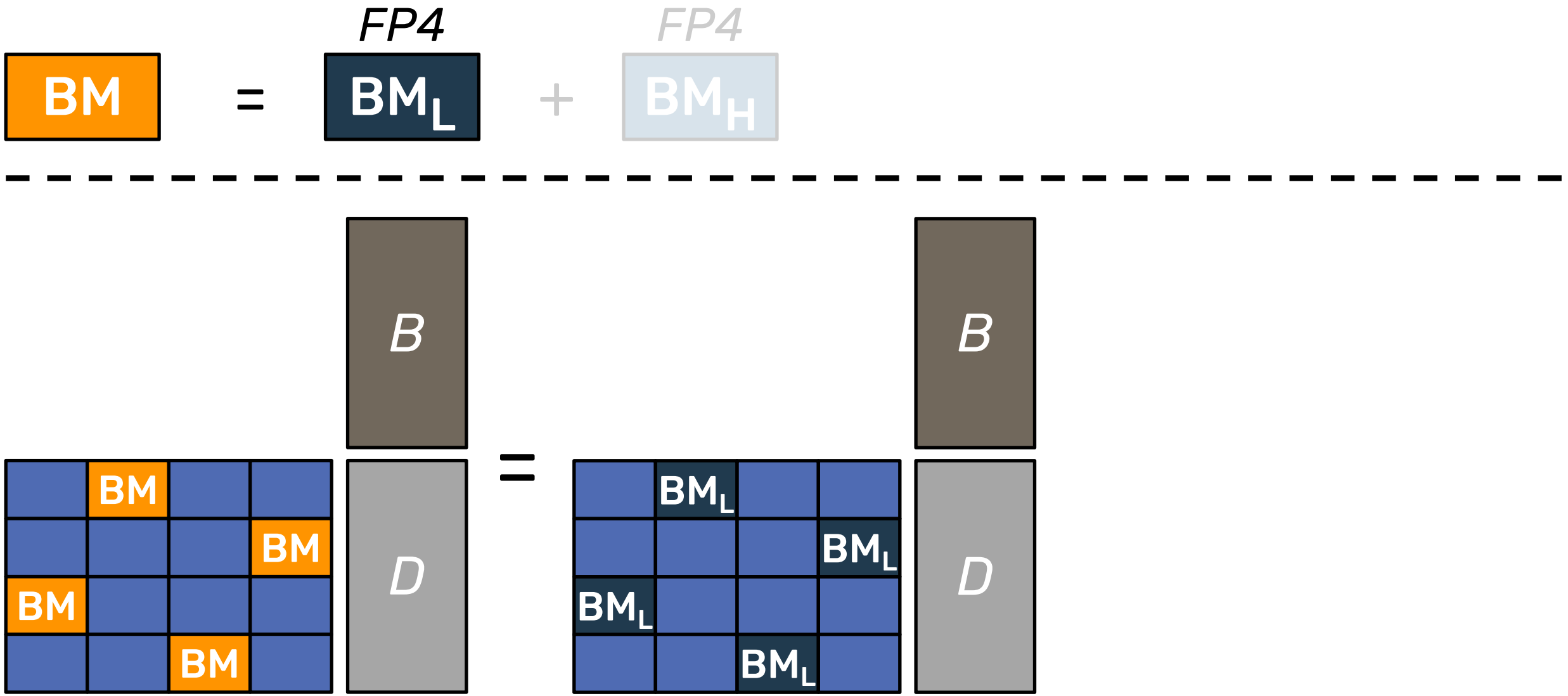
Tensor Core cannot perform MMA operation with MX+ ☹

➔ Compute BMs in Tensor Core ✓

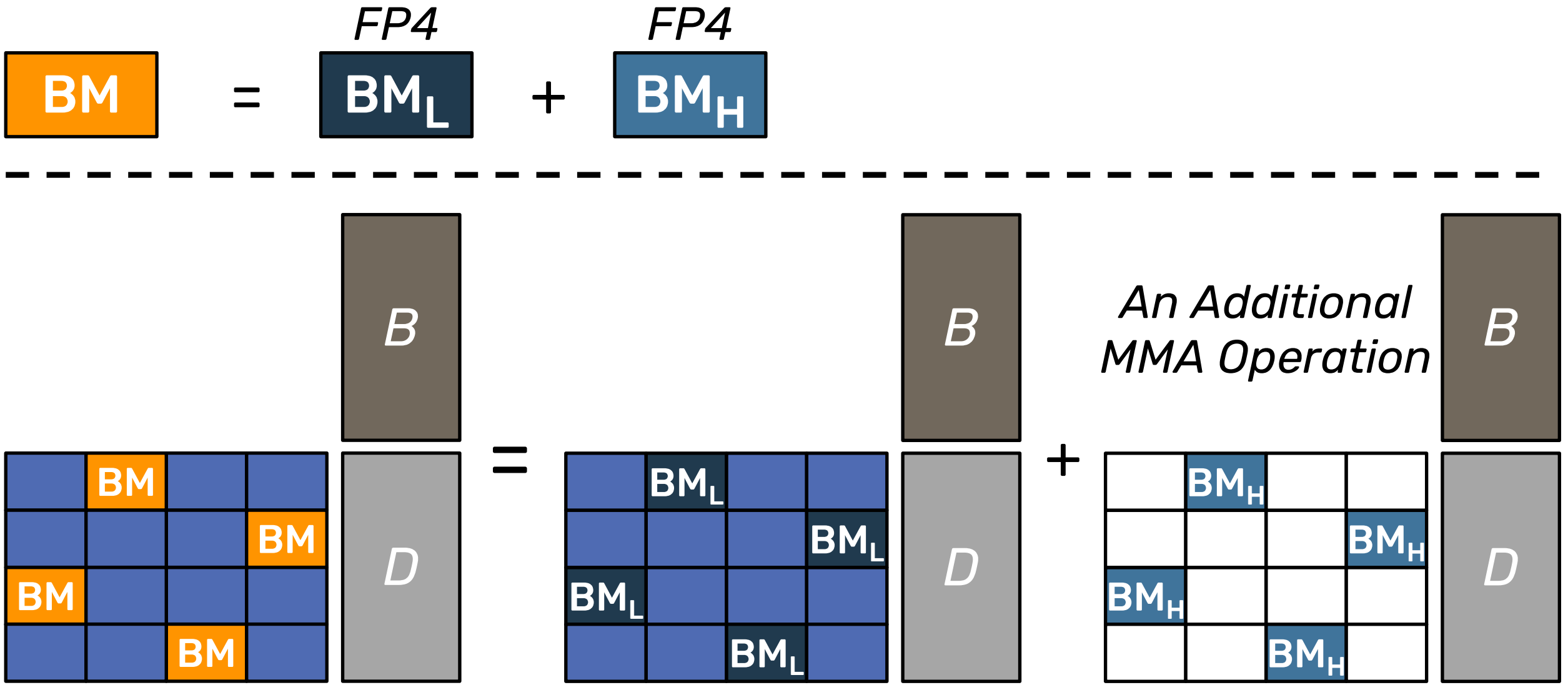
Maintains Performance in the Decode Stage ☺

Software Integration of MX+ on GPUs

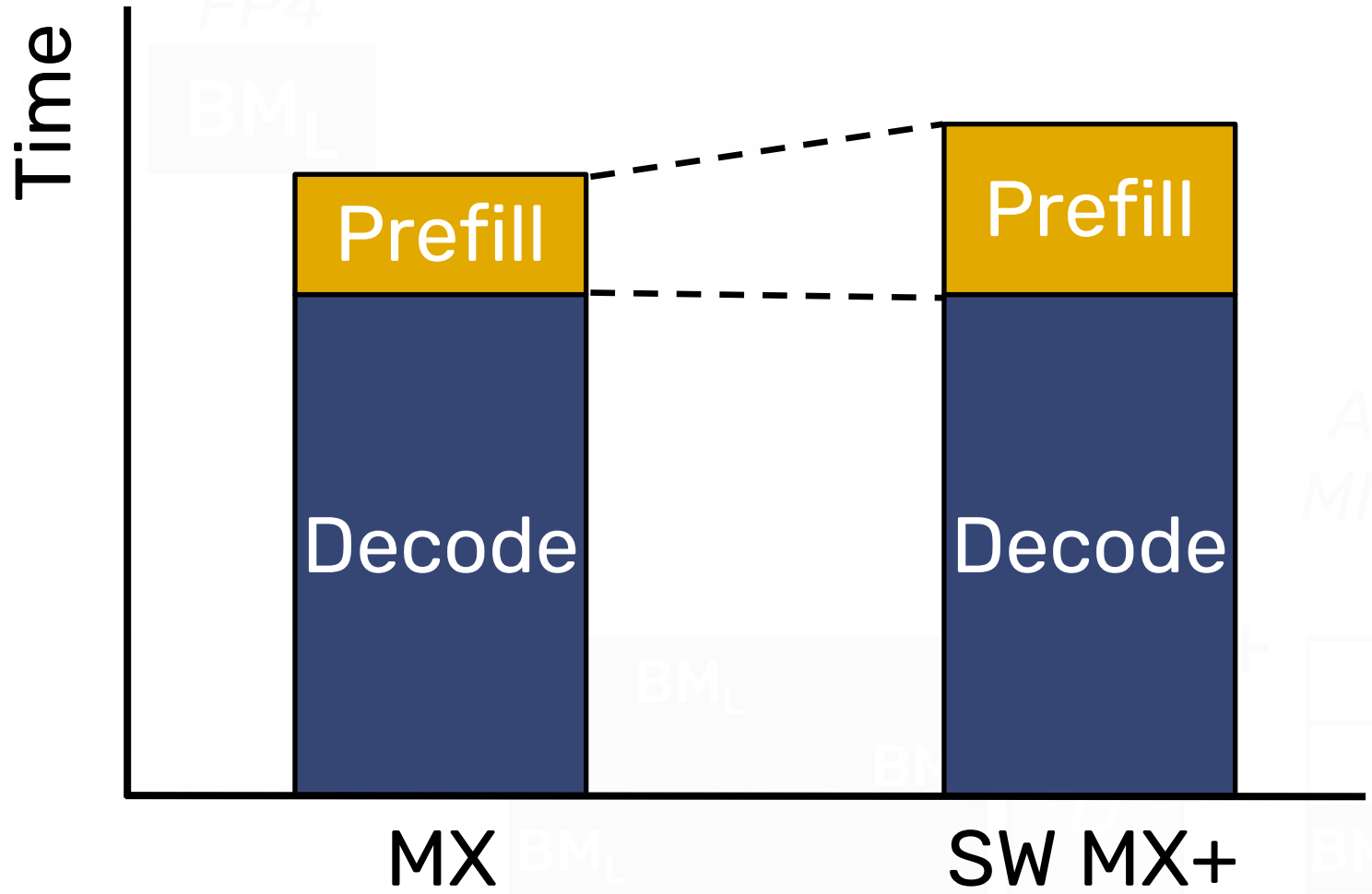
Software Integration of MX+ on GPUs



Software Integration of MX+ on GPUs



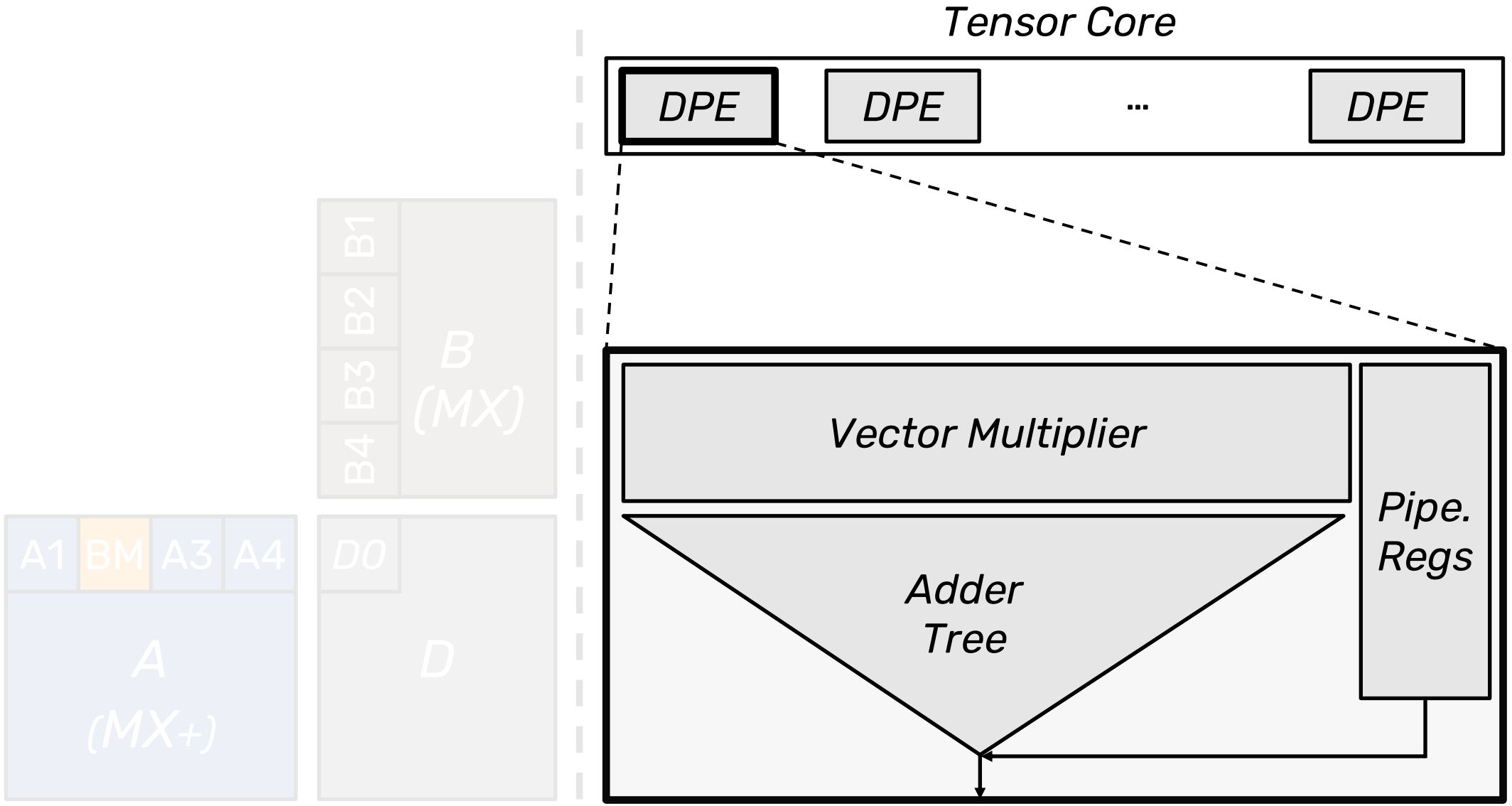
Software Integration of MX+ on GPUs



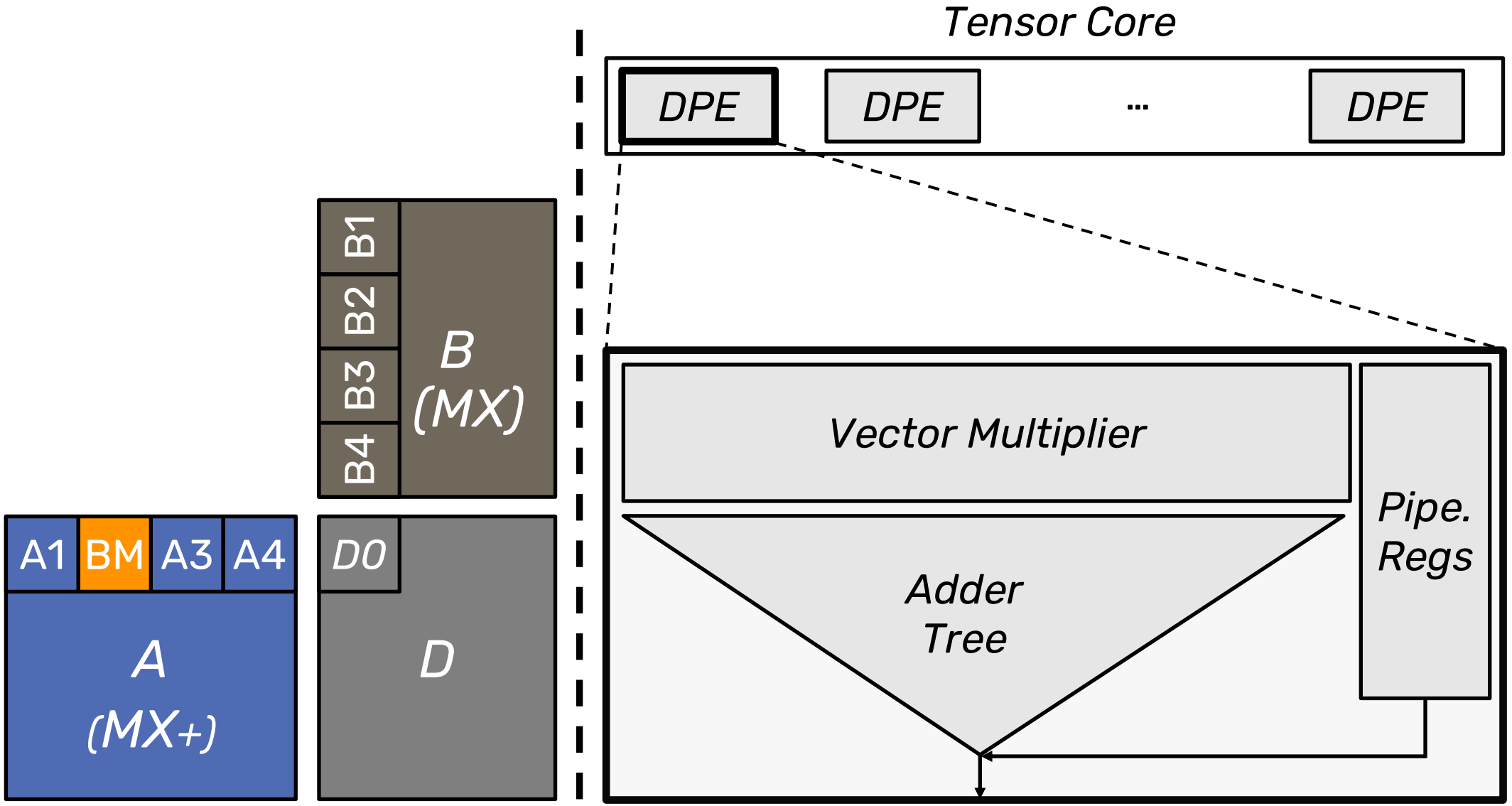
Moderate Overhead

Maintains Performance

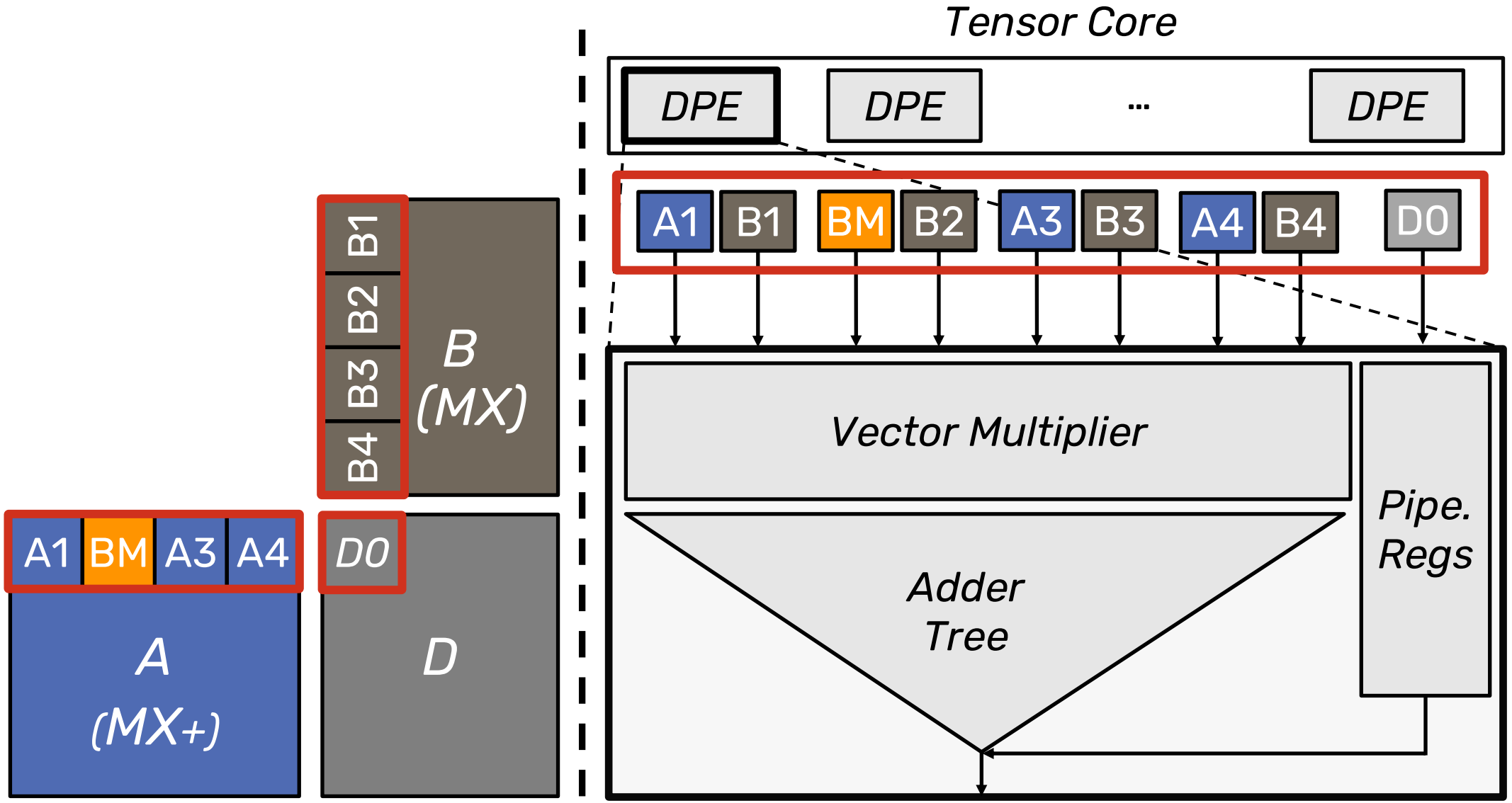
Architectural Support for MX+



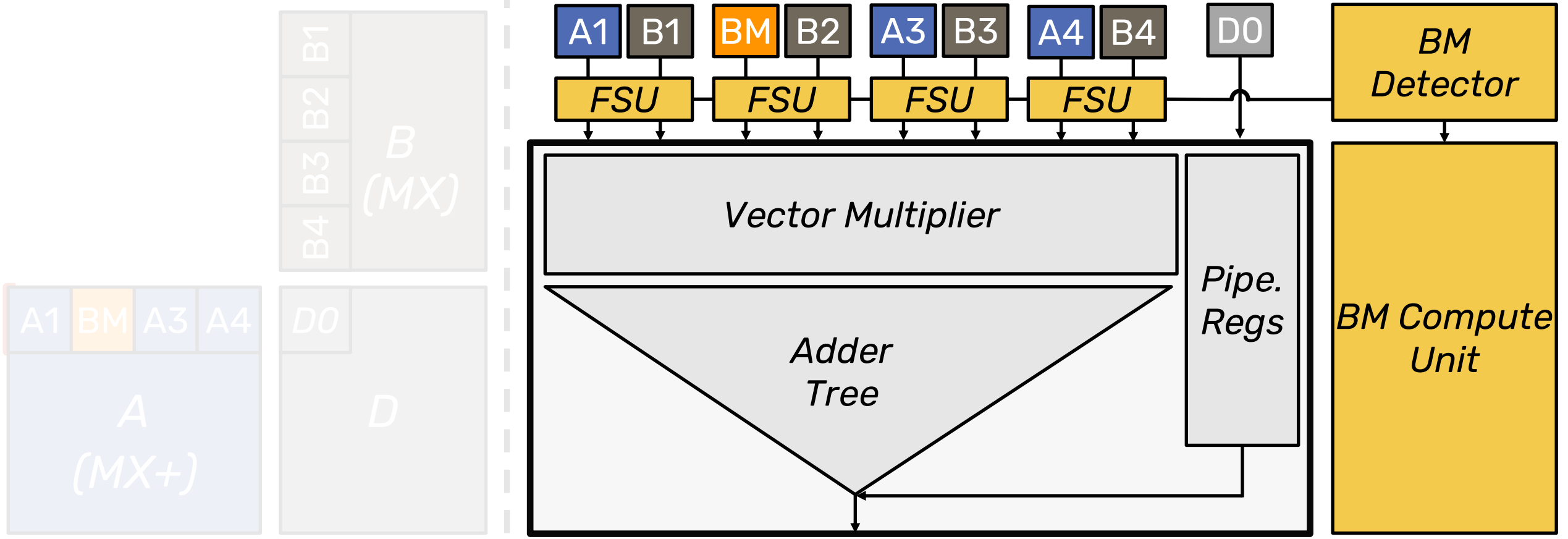
Architectural Support for MX+



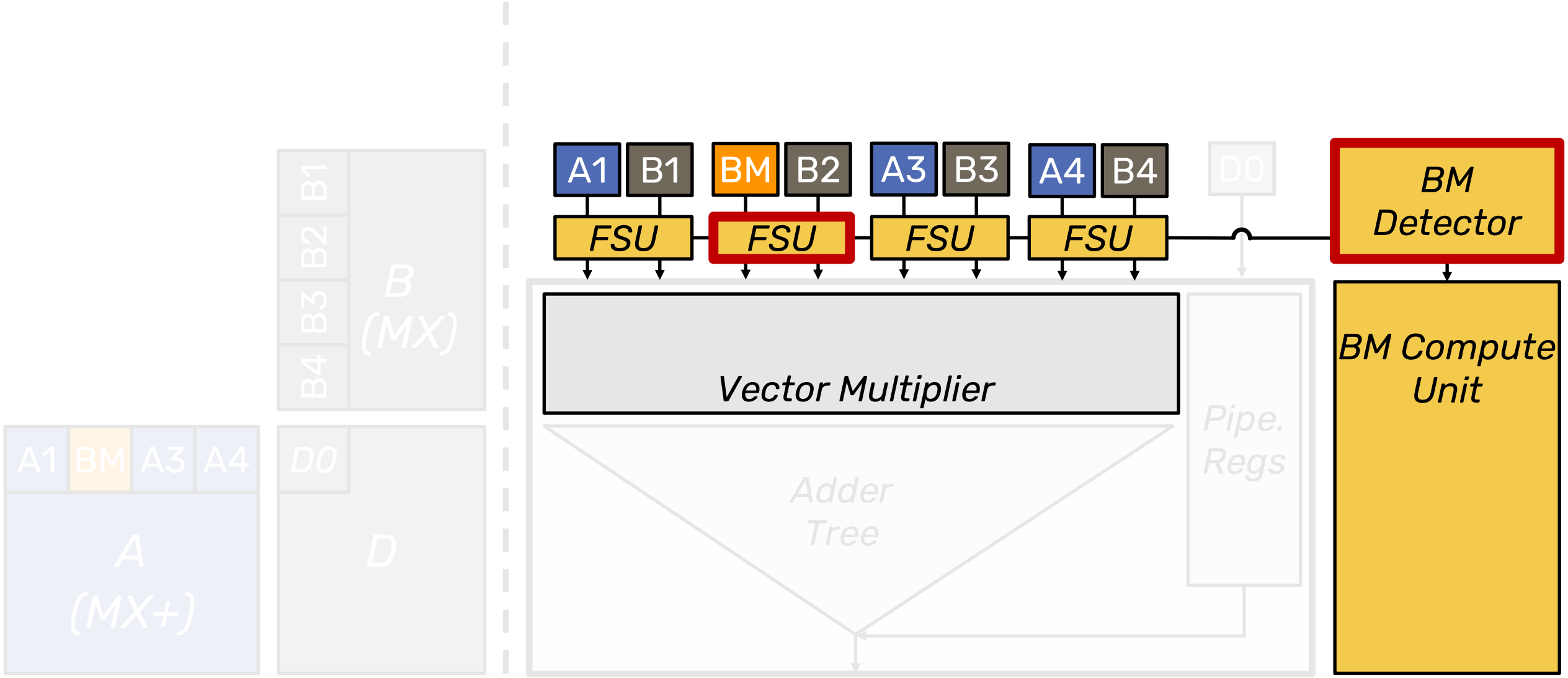
Architectural Support for MX+



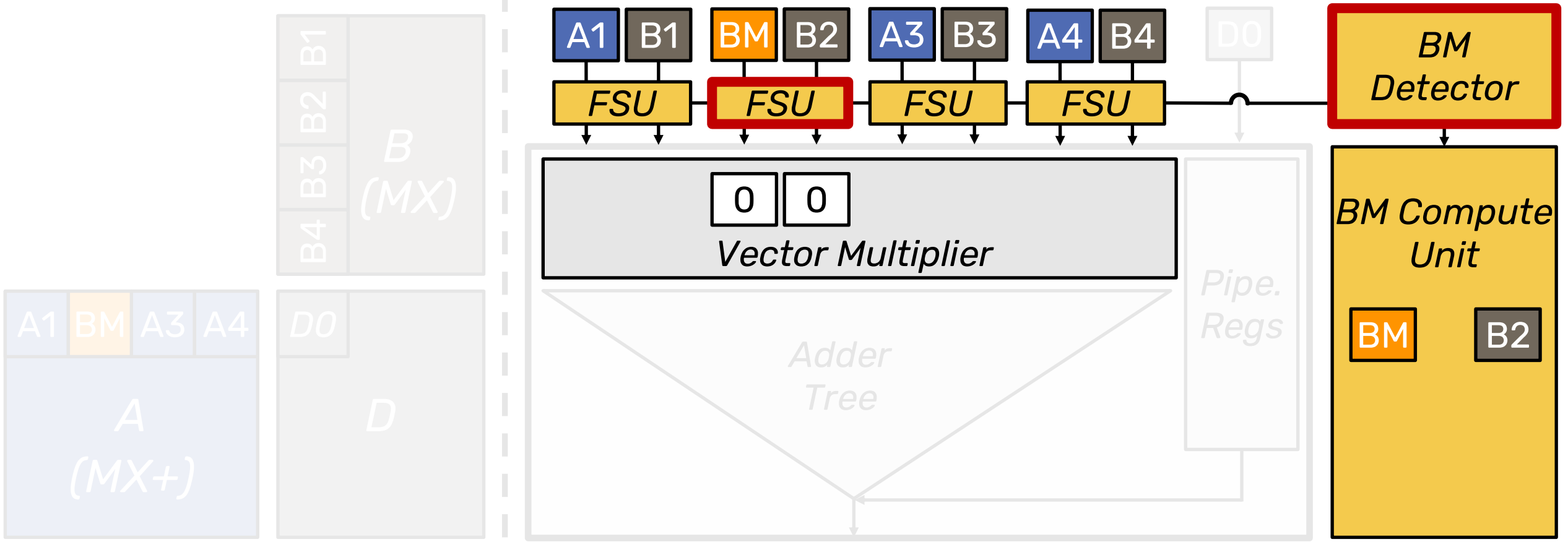
Architectural Support for MX+



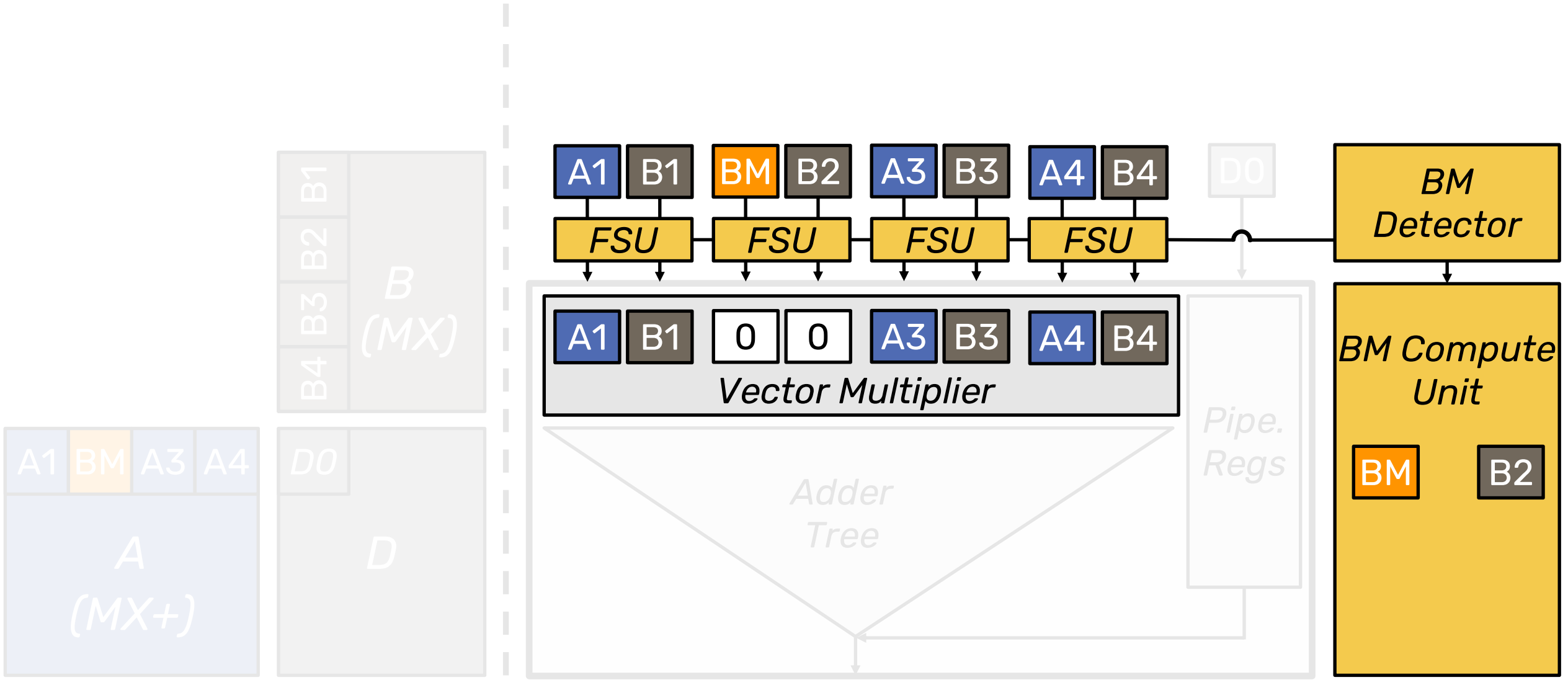
Architectural Support for MX+



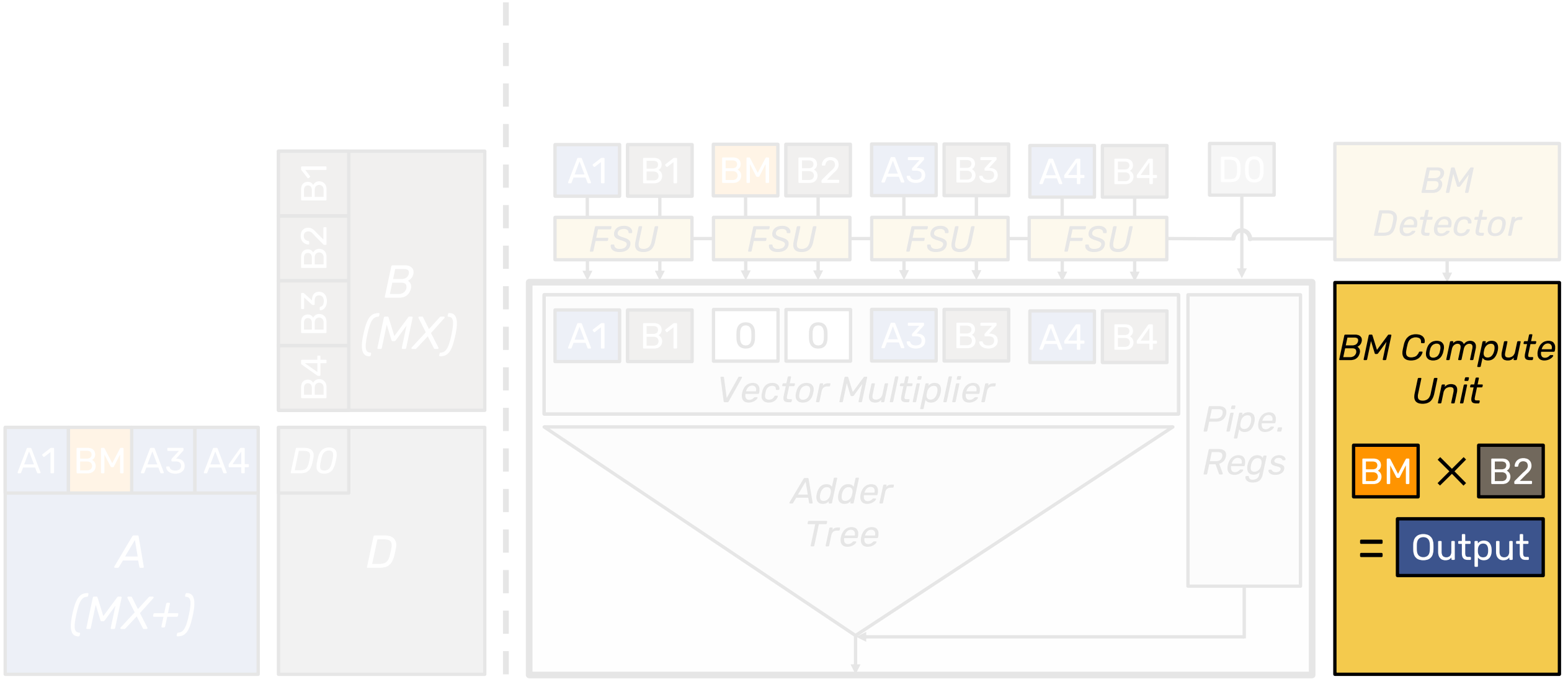
Architectural Support for MX+



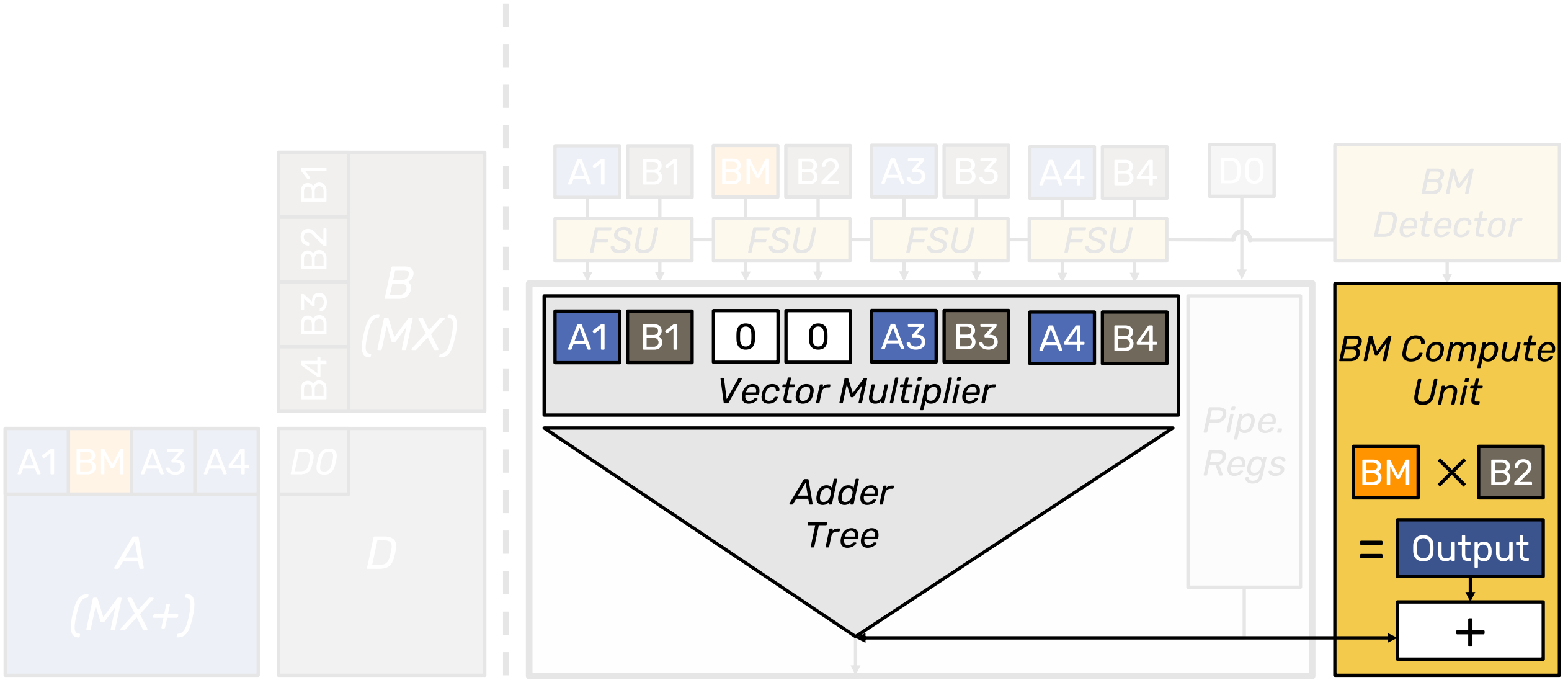
Architectural Support for MX+



Architectural Support for MX+



Architectural Support for MX+



Outline

- **Introduction to MX Formats**
- **Motivation**
  - Implications for Low-bit LLM Serving
  - Analysis on Low-bit MX Formats
- ***MX+*: Cost-Effective and Non-intrusive Extension to MX Formats**
  - Key Insight
  - Software & Hardware Integration on GPUs
- **Evaluation**
- **Conclusion**

Experimental Setup

Models

- OPT, Llama-2 & 3.1, Mistral, etc.

Accuracy

- MX PyTorch emulation library

Performance

- Software: CUTLASS library
- Hardware: NVIDIA RTX 5090 GPU

Workloads

- lm-evaluation-harness

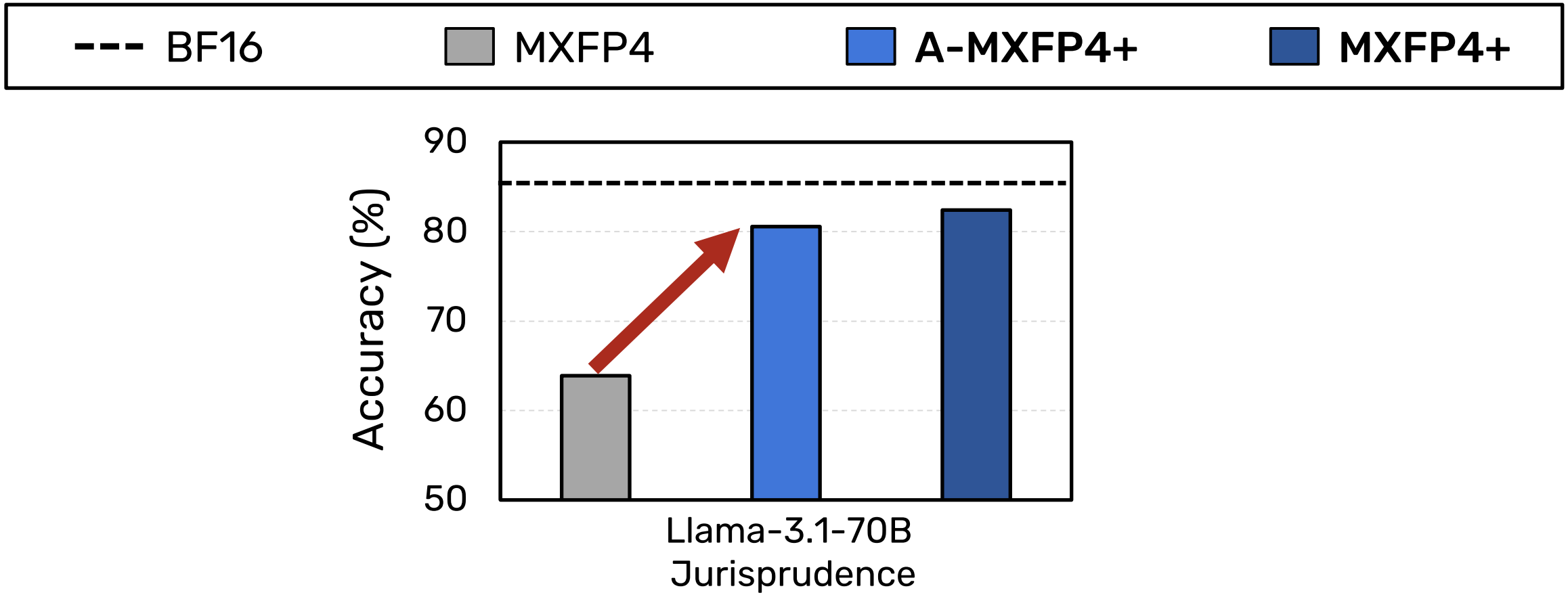
MX+ Schemes

Scheme	Activation	Weight
A-MXFP4+	MXFP4+	MXFP4
MXFP4+	MXFP4+	MXFP4+

MX Schemes

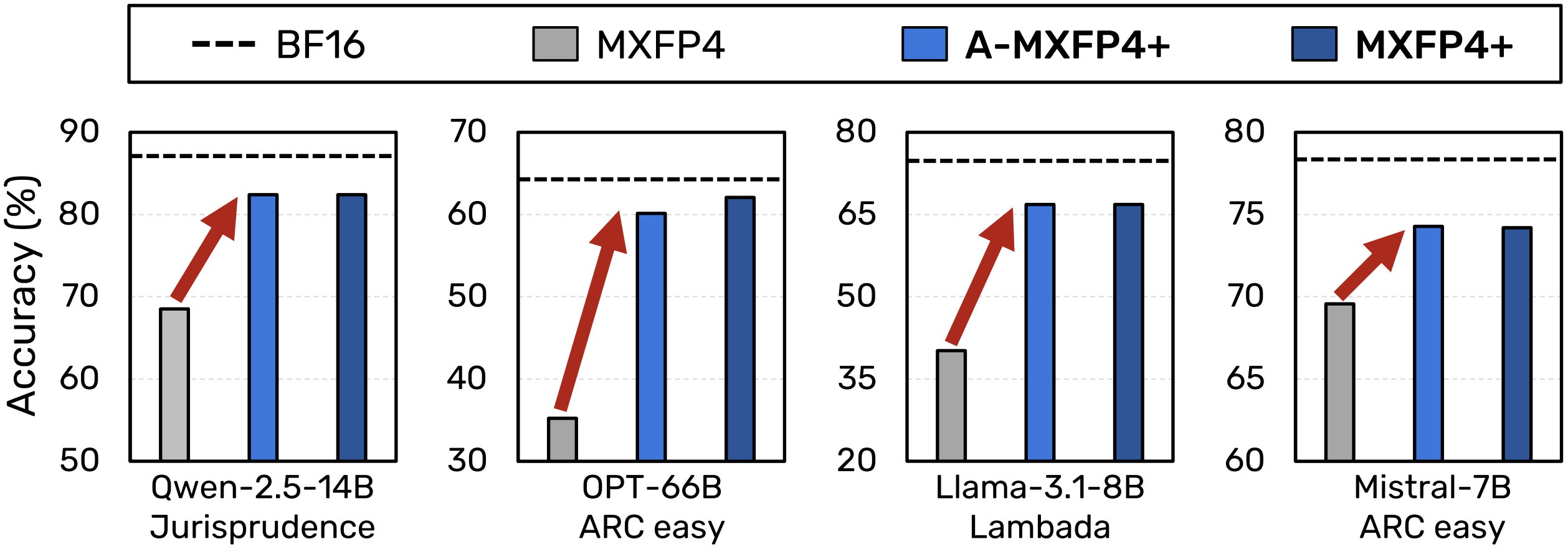
Scheme	Activation	Weight
MXFP4	MXFP4	MXFP4

Accuracy



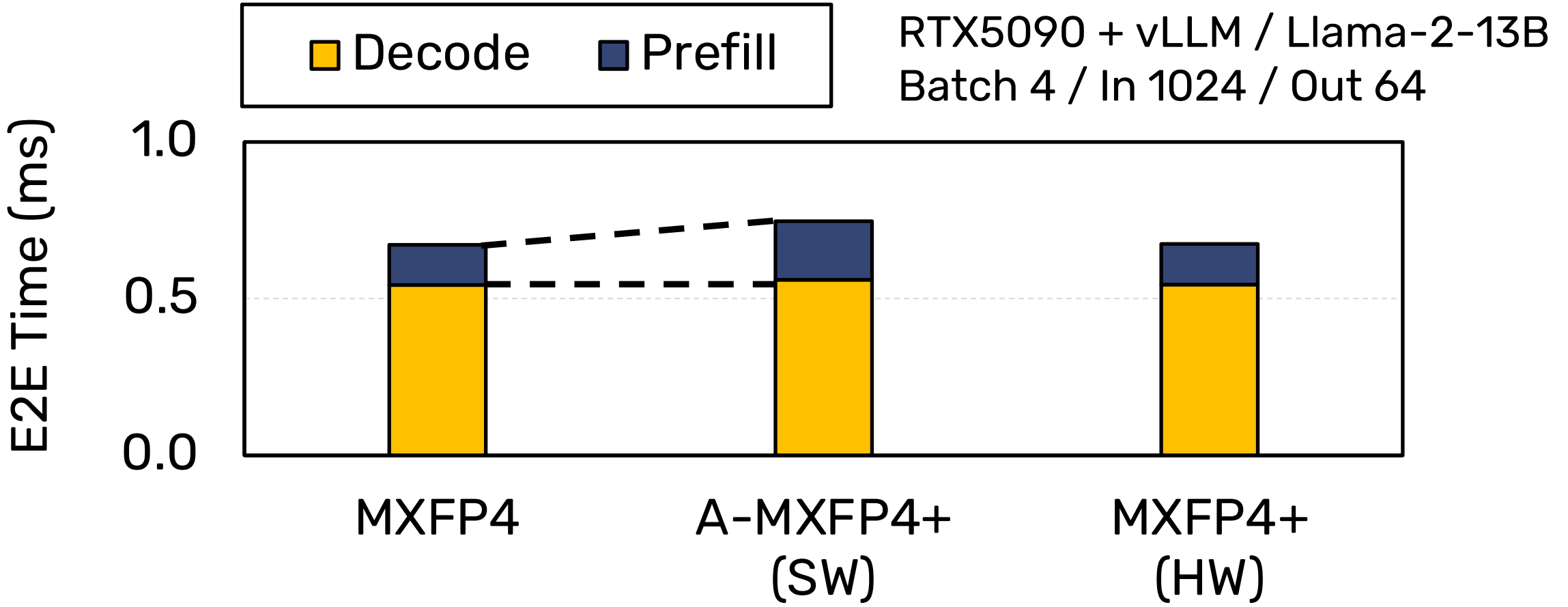
MX+ greatly improves accuracy over MX!

Accuracy



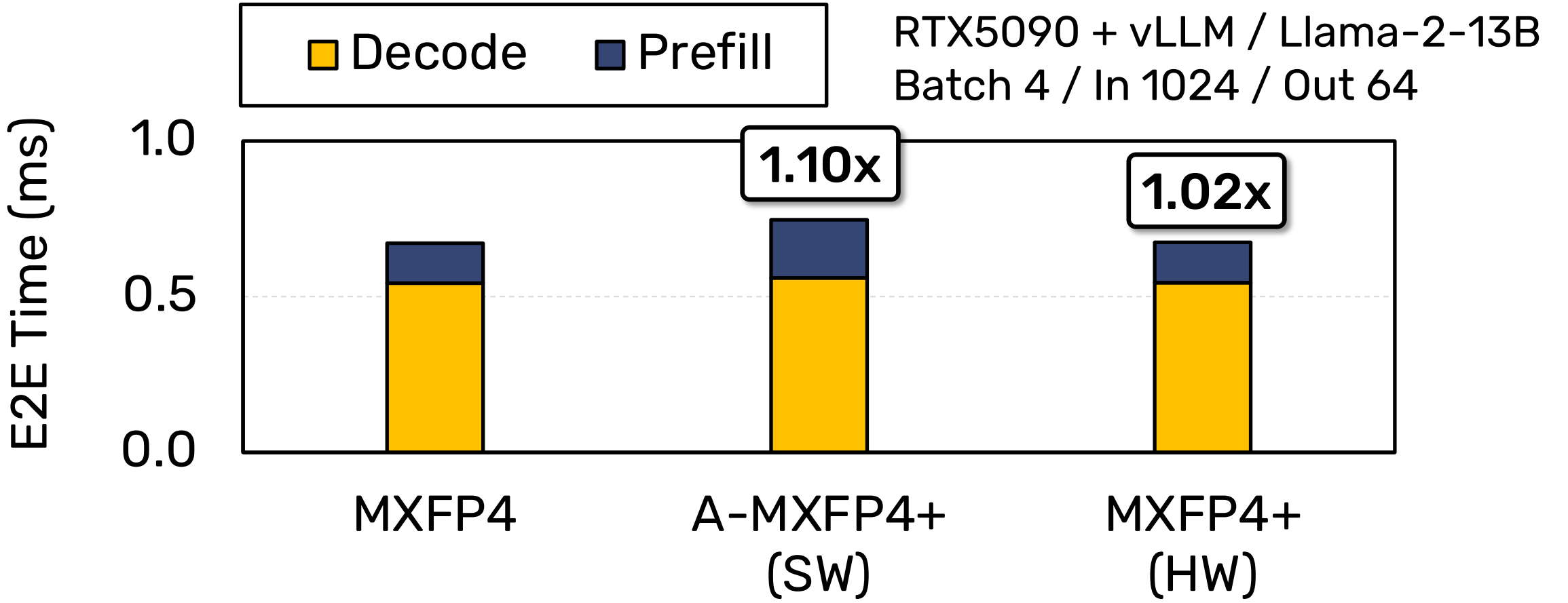
MX+ greatly improves accuracy over MX!

E2E Performance



MX+ maintains comparable performance to MX

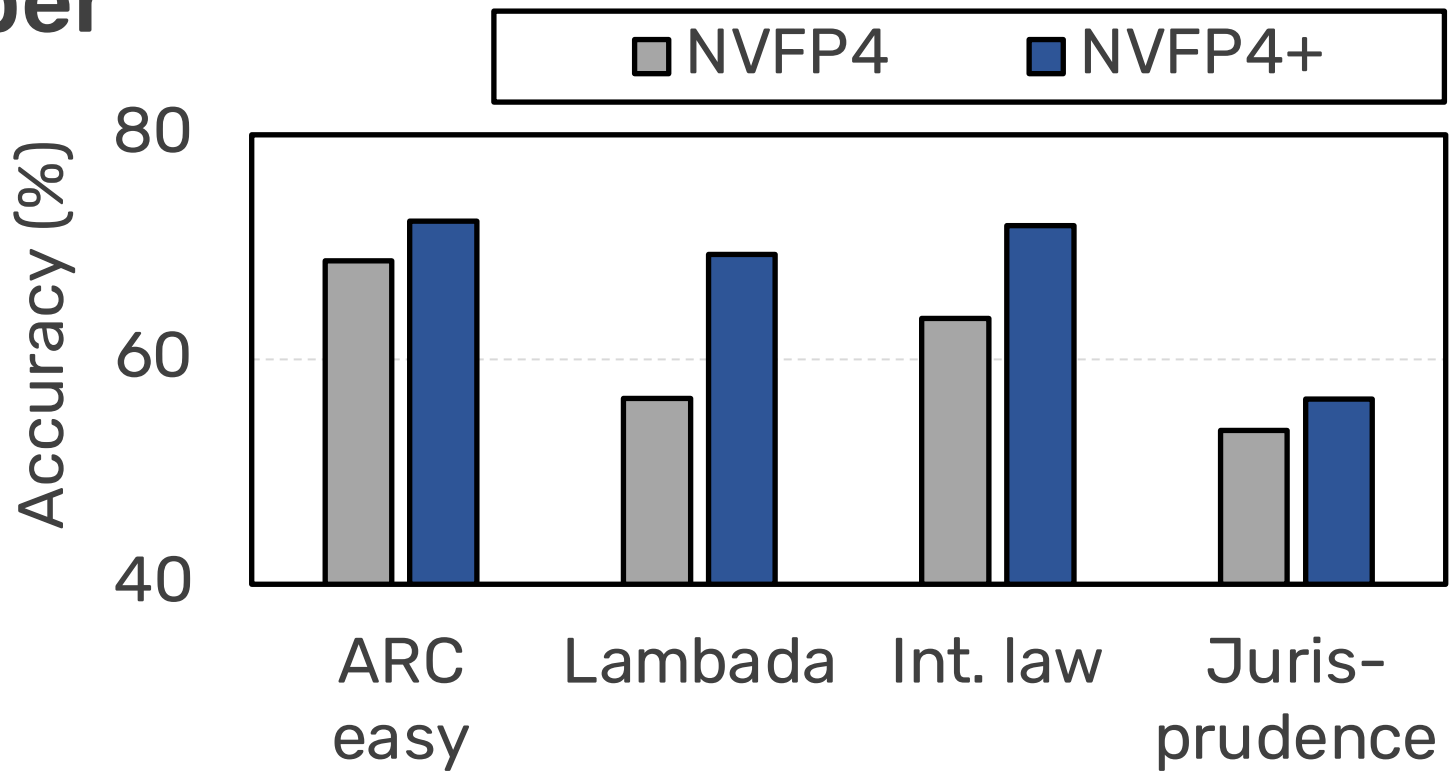
E2E Performance



MX+ maintains comparable performance to MX

More Details in Our Paper

- Application of MX+ for NVFP4
  - And comparison with MXFP4+



- Comparison with A8W4 and other quantization works
- Handling multiple outliers in a block
- Use of reserved bits
- MX+ for other DNN workloads (e.g., ViT and CNN)

Conclusion

Problem

- Low-Bit MX Format shows subpar accuracy

Solution: *MX+*, a cost-effective and non-intrusive extension to MX

- Higher precision for BMs by repurposing exponent bits as extended mantissa
- Integration to existing software and hardware system

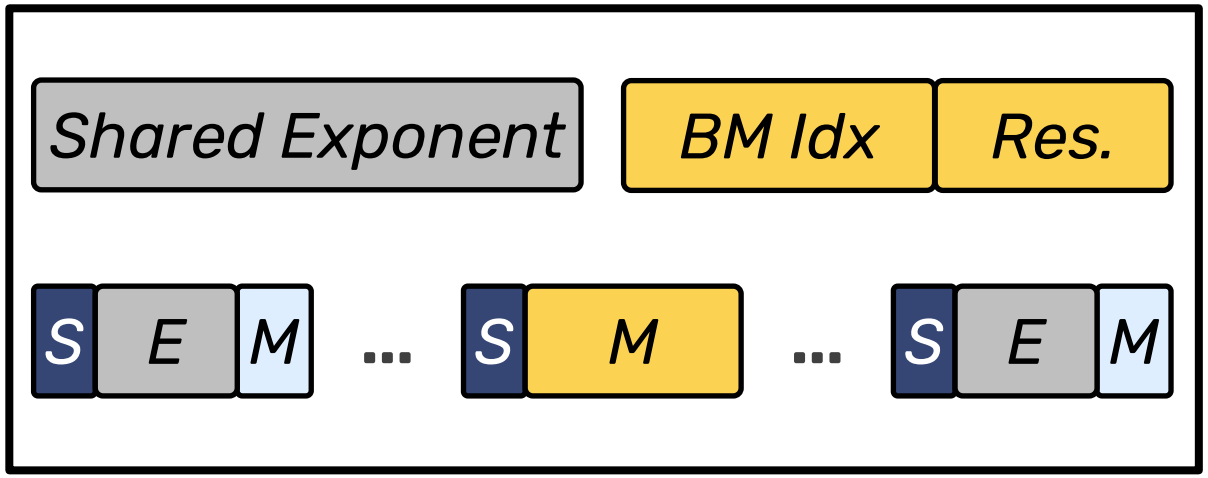
Result

- MX+ improves **accuracy by up to +42.15%** compared to the MX format
- **Comparable performance** under software and hardware integration

Thank You!

MX+

Pushing the Limits of
Microscaling Formats for Efficient
Large Language Model Serving



Jungi Lee (jungi.lee@snu.ac.kr)

MICRO'25 | October 2025