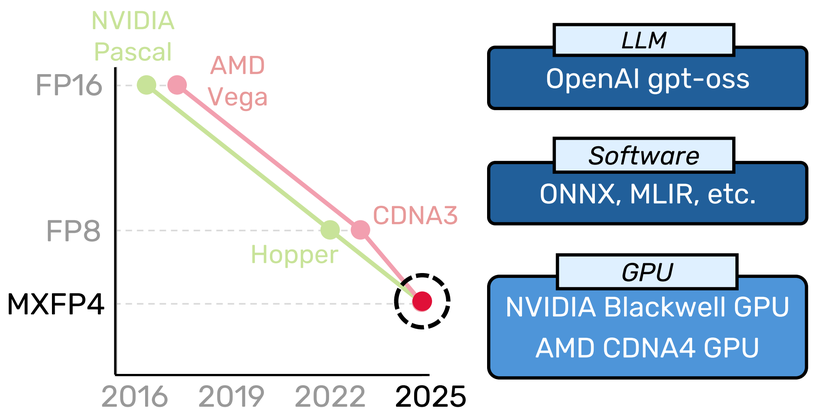
MX+: Pushing the Limits of Microscaling Formats for Efficient Large Language Model Serving

Jungi Lee, Junyong Park, Soohyun Cha, Jaehoon Cho, Jaewoong Sim
Seoul National University

Background

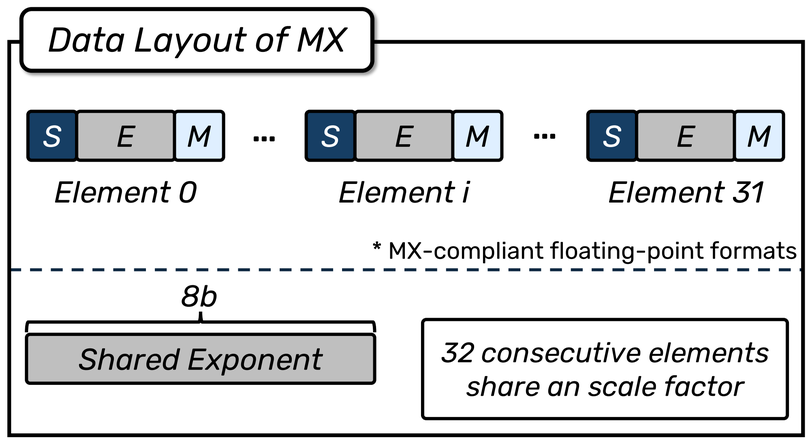
Increasing Support for MX Formats

MXFP4 is supported across the latest GPUs and used in various software and LLMs.



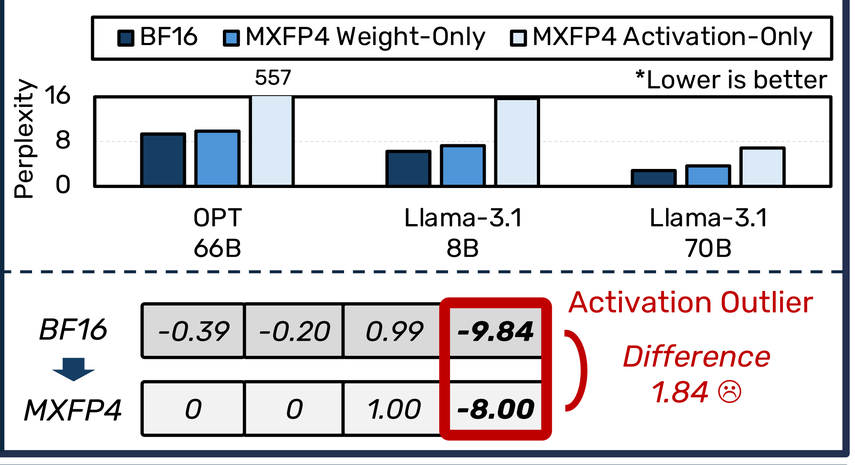
MX Formats

MXFP4 is a data type in **MX formats**, which are **open and interoperable** data formats standardized by the **industry.**



Implications on LLM Serving

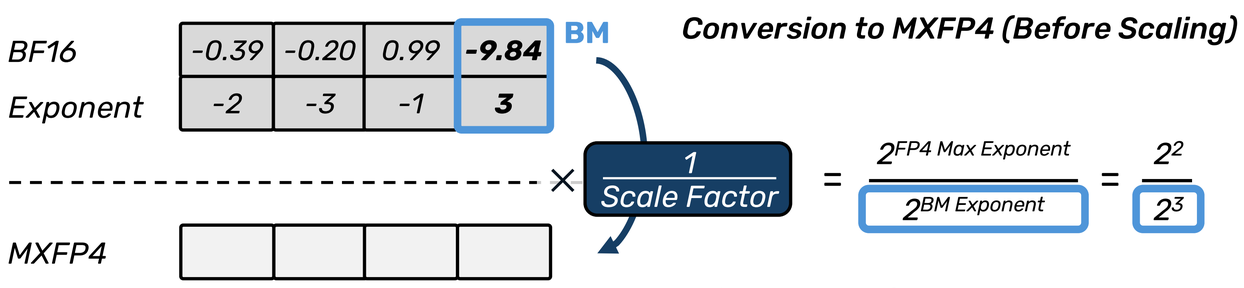
Using MXFP4 for **LLM serving is not practical**, as its **low precision** incurs a large error for **activation outliers.**



Key Insights

Insight 1

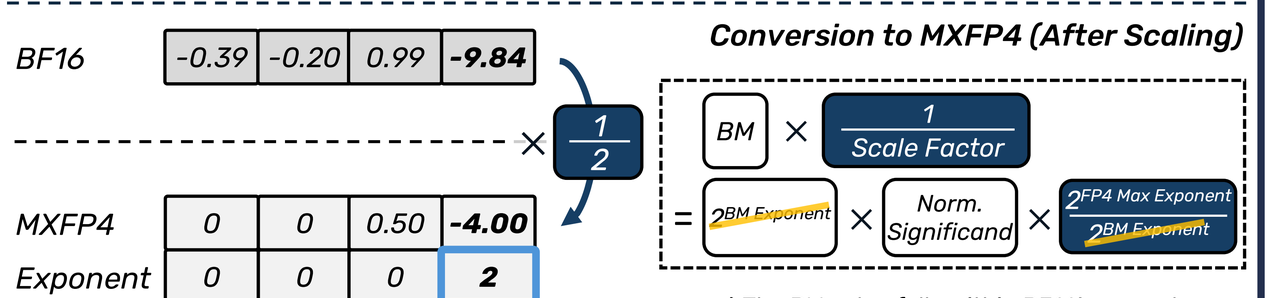
The absolute maximum value in an MX block (Block Max or BM) is **naturally identified during conversion** from high precision to MX.



* The BM value falls within BF16's normal range.

Insight 2

The BM element **does not need to store its own exponent** as it is **set to the maximum value** after scaling.

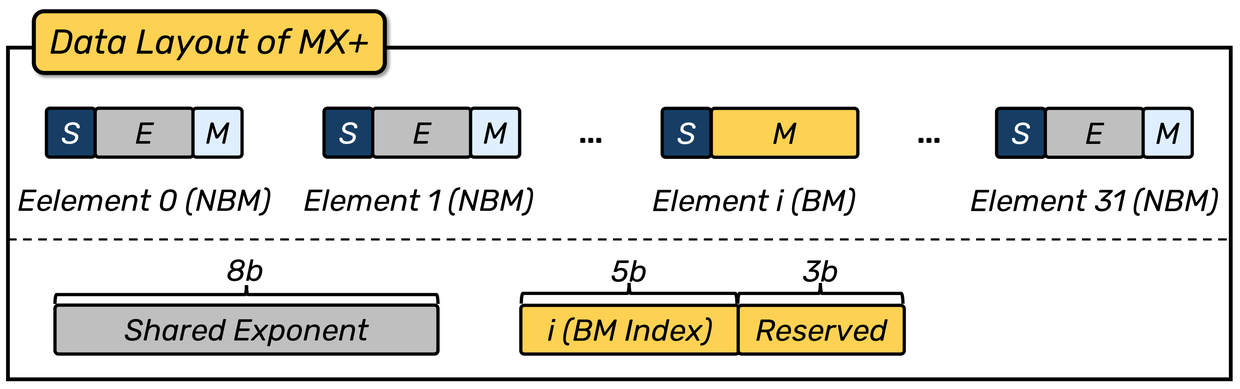


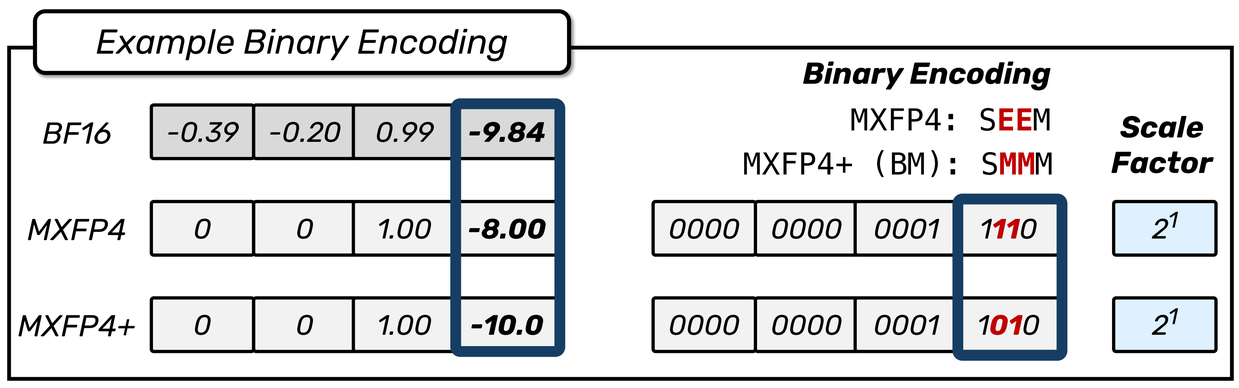
* The BM value falls within BF16's normal range.

MX+

MX+ Design

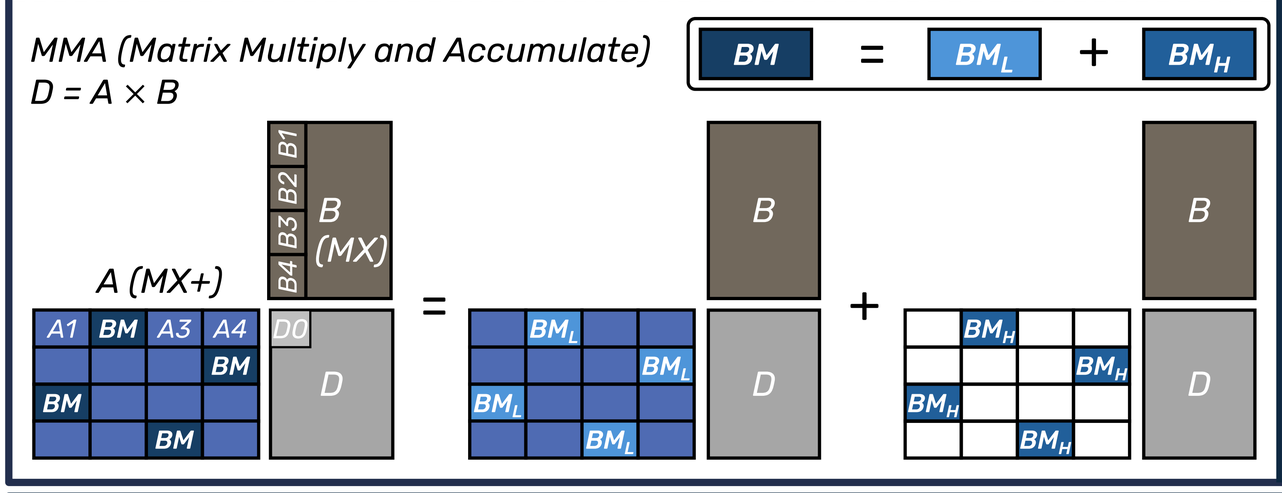
Key Direction: **Repurpose the exponent field** of BM as an **extended mantissa** to more precisely represent the BM value.





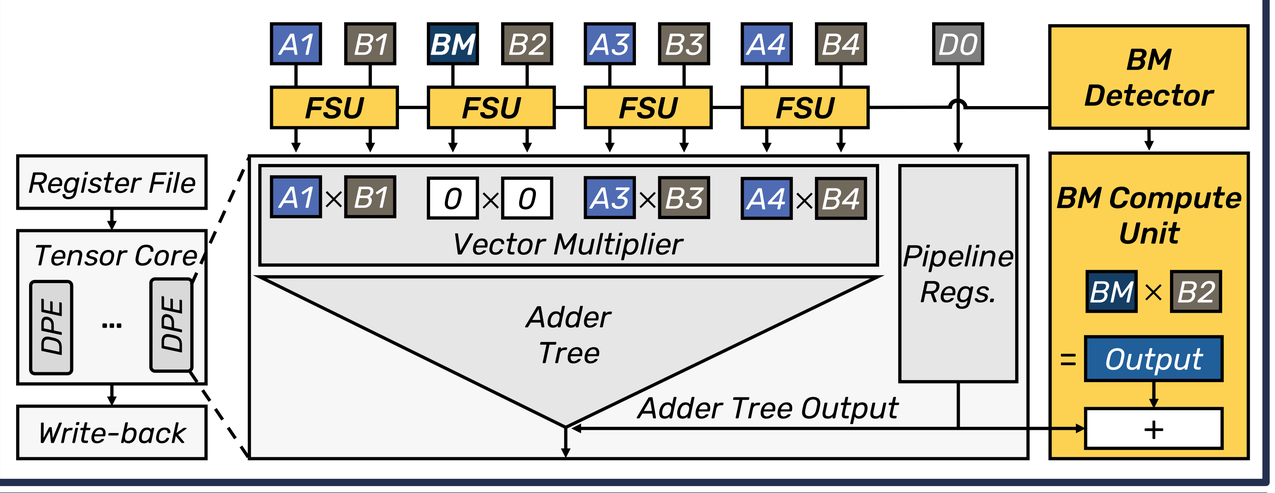
Software Integration on GPUs

Performs an additional MMA operation and **maintains performance in the decode stage**



Architectural Support on GPUs

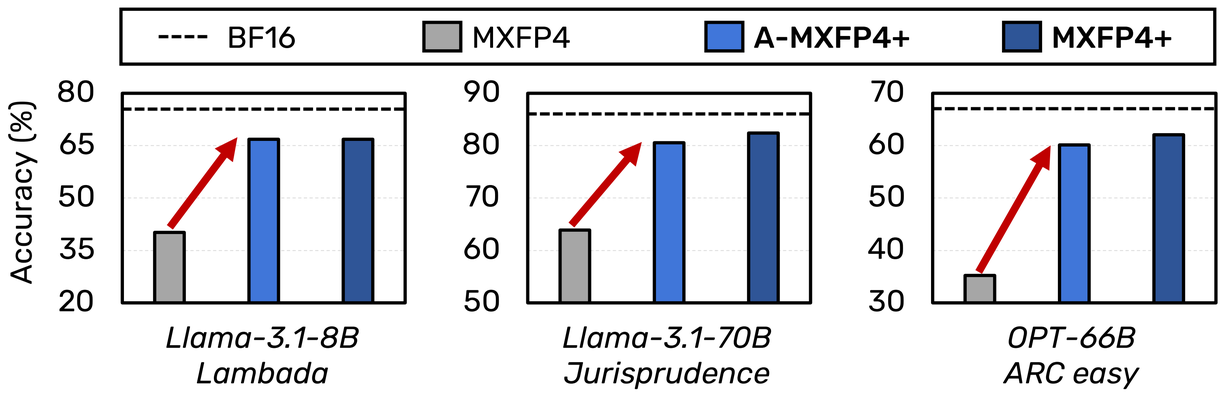
Removes the additional MMA operation and **maintains performance in both stages.**



Results

Accuracy

MX+ greatly improves accuracy over MX.



End-to-End Performance

MX+ shows comparable performance to MX.