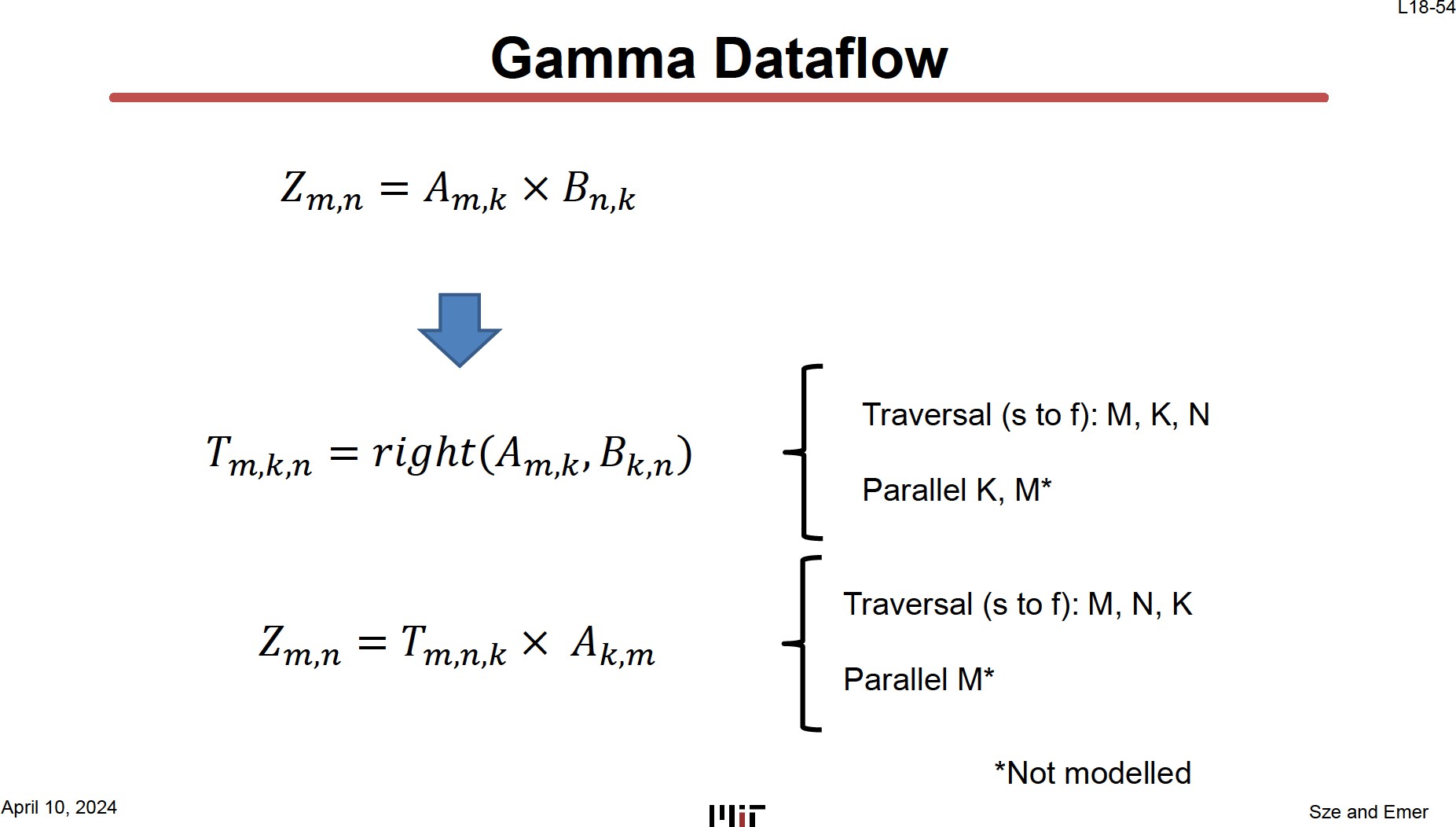
Gamma



A screenshot of a computer program

AI-generated content may be incorrect.

# A screenshot of a computer AI-generated content may be incorrect.Gamma - Step 1

𝑇𝑚,𝑘,𝑛 = 𝑟𝑖𝑔ℎ𝑡(𝐴𝑚,𝑘, 𝐵𝑘,𝑛)

Traversal (s to f): M, K, N Parallel K, M\*

\*Not modelled

# Gamma – Step 1 - Observations

* **There is a single concordant traversal of A**
* **The same B\_n fibers are fetched multiple times.**
* **For each specific M, the processing is parallel across K**
  + **And the T\_n fibers below are created concordently**
  + **Thus, creating T in a manner that allows for it to be rank swizzled**

T[M,K,N]

# Gamma - Rank Swizzled T

A diagram of a diagram

AI-generated content may be incorrect.

T’[M,N,K]

A computer screen shot of a diagram

AI-generated content may be incorrect.

Since elements of each K fiber in T[M,K,N] are processed in parallel and elements in

N fibers are created concordantly, the head elements needed for the swizzle are available!

# Gamma – Step 2

A screenshot of a computer

AI-generated content may be incorrect.𝑍𝑚,𝑛 = 𝑇𝑚,𝑛,𝑘 × 𝐴𝑘,𝑚

Traversal (s to f): M, N, K Parallel M\*

\*Not modelled

# Gamma – Step 2 - Observations

* **Exactly one concordant traversal of (swizzled) T tensor**
* **Concordant traversal of (swizzled) T that means it can be created in pipeline and consumed immediately without being held in its entirety in a buffer.**
* **Note that A\_k fibers are re-read repeatedly but are small since they are post-intersection.**
* **Output Z is created concordantly**

A diagram of a block diagram

AI-generated content may be incorrect.

A diagram of a memory

AI-generated content may be incorrect.

A diagram of a computer program

AI-generated content may be incorrect.

FiberCache decouples read roundtrips and memory latencies

L18-64



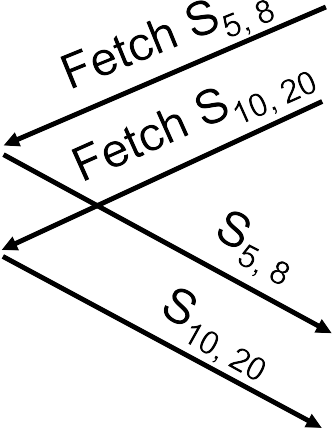
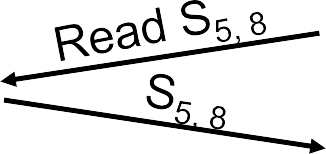
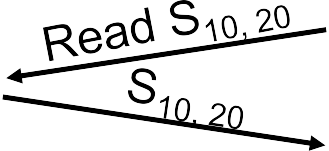
Memory

FiberCache



PE

**Time**

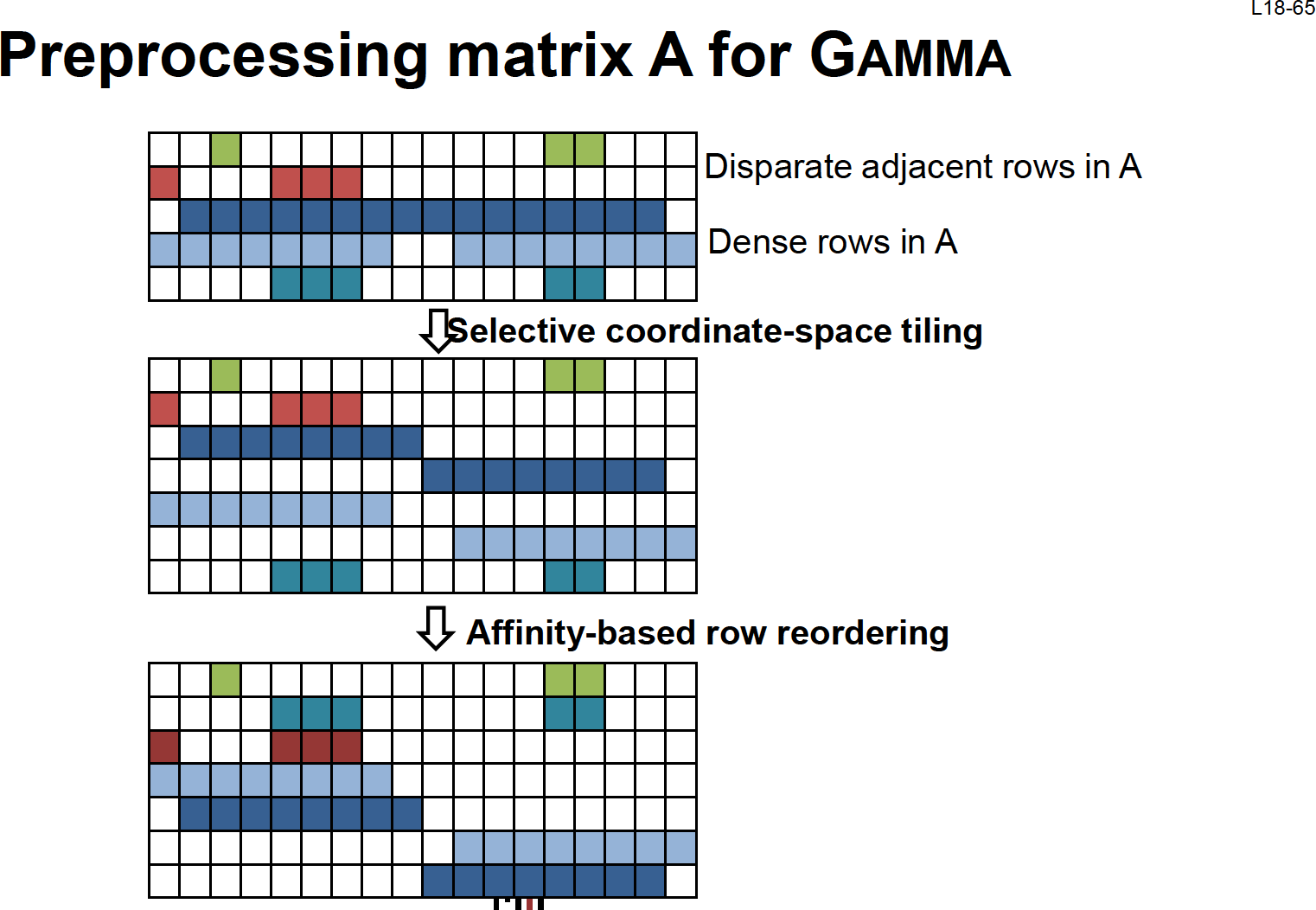


…

…

**Low roundtrip latency easy to cover with small buffers**





# Gamma Concepts

* **Pipeline computations with small intermediate storage**
* **Use parallelism/merger to do pipelined rank swizzle**
* **Decoupled/implicit fibercache to hold B fibers that might be reused**
* **Reorder A to maximize effectiveness of fibercache**

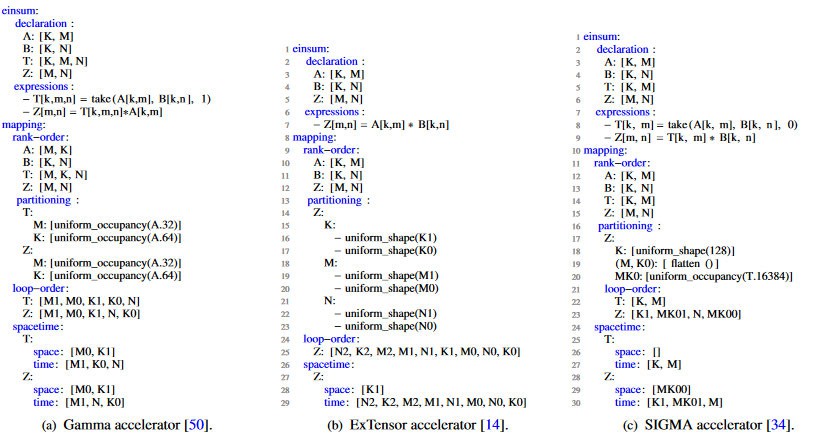
TeAAL – Modeling Sparse Dataflows

A diagram of a circuit

AI-generated content may be incorrect.

# TeAAL - Matrix Multiplication Designs

L18-69



[Nayak, TeAAL, MICRO2023]

A diagram of a software model

AI-generated content may be incorrect.

Summary

# spMspM dataflows

AMxK BKxN ZMxN

x =

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

M M

K K N

N

𝑍𝑚,𝑛 = � 𝐴𝑚,𝑘𝐵𝑘,𝑛

𝑘

Output Stationary Inner-product

**for** m in [0, M)

**for** n in [0, N)

**for k in [0, K)**

A-stationary – Column major Outer-product

**for k in [0, K)**

**for** m in [0, M)

**for** n in [0, N)

A-stationary – Row major Gustavson

**for** m in [0, M)

**for k in [0, K)**

**for** n in [0, N)

Z[m,n] += A[m,k] \* B[k,n]

Z[m,n] += A[m,k] \* B[k,n]

Z[m,n] += A[m,k] \* B[k,n]

𝑍𝑚 = � 𝑎𝑚,𝑘𝐵𝑘

𝑘

A diagram of data flow

AI-generated content may be incorrect.

## Speedups over Intel MKL on common-set matrices

**A graph with green bars

AI-generated content may be incorrect.**

MKL

OuterSPACE

SpArch GAMMA (w/o Preprocessing) GAMMA (w/ Preprocessing)

EIE

April 10, 2024

Sze and Emer

A screen shot of a computer program

AI-generated content may be incorrect.

A diagram of a graph

AI-generated content may be incorrect.

A diagram of a pe architecture

AI-generated content may be incorrect.

# Summary

* Design attributes of spMspM accelerators:
  + Data can be tiled to improve locality
  + Sparse data makes intersection an explicit operation
  + Intersection can be hierarchical – intersecting at higher levels of the fibertree
  + There are three major dataflows for spMspM
  + spMspM can be broken into multiple pipelined stages
  + Rank swapping can be required to achieve concordant traversals
  + Rank swapping can be implemented with a “merge” unit
  + Data movement can be optimized via data format selection
  + Data movement can be reduced with explicit-decoupled caching
* Most of the above can be expressed as a scheduled Einsum
* A loop nest implementation can be inferred from a scheduled Einsum
* Lots of interesting variations in spMspM acceleration!

Thank You