Fused-Tiled Layers: Minimizing Data Movement on RISC-V SoCs

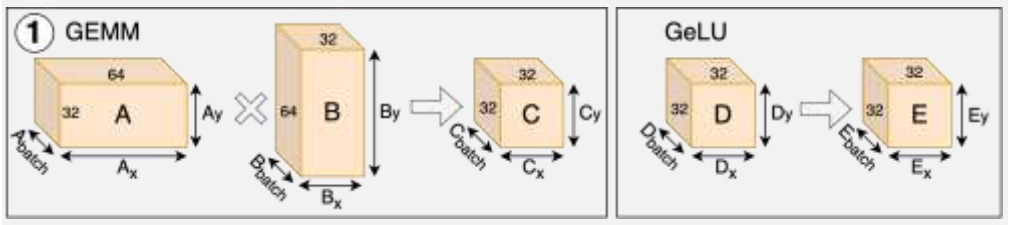
Victor J.B. Jung[1], Alessio Burrello[3], Francesco Conti[2], Luca Benini[1,2]
[1]Integrated Systems Laboratory, ETH Zurich | [2]DEI, University of Bologna | [3]DAUIN, Politecnico of Turin

1 Introduction

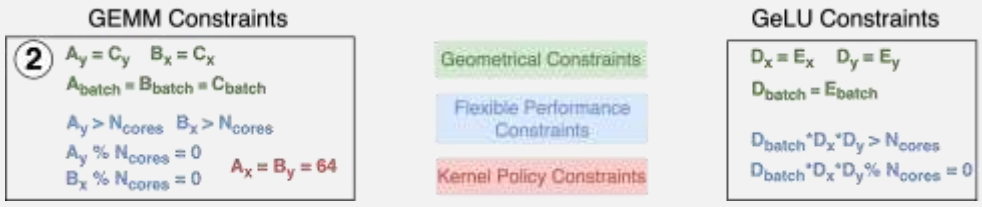
How can we **minimize data movement** on RISC-V SoCs featuring **software-managed caches**? We propose **Fused-Tiled Layers (FTL)**, a novel algorithm for automatic fusion between tiled layers. We leverage the flexibility and efficiency of a **RISC-V heterogeneous SoC** [1] to integrate FTL and benchmark it on a Multi-Perceptron stage of **Vision Transformers.**

2 Method Overview

Fused-Tiled Layers (FTL) formulates the tiling of each DNN layer as a constraint optimization problem, where each output tensor dimension is linked to input tensor dimensions via a linear transformation, allowing us to merge several layers to generate valid layer fusion solutions for any layer combination. By doing so, we minimize transfers from L2 memory to LLC.



1: We attribute a variable for each tensor dimension related to the given operator.



2: We formulate the constraints for the tiling of the single operator:

- **Geometrical Constraints**: decribe the data dependency between the dimensions of the output and input tensors.
- **Kernel Policy Constraints**: ensure that we respect the specificities of the kernel's dataflow.
- **Flexible Performance Constraints**: to boost the hardware utilization, for instance to encourage parallelization.

3 Fuse Layer Constraints	$C_{batch} = D_{batch}$	$C_x = D_x$	$C_y = D_y$

3: We select the consecutive layers to fuse and bind their shared tensors dimensions. This step effectively constructs one constraint optimization problem representing the tiling of several layers.

4 Tiling and Memory Allocation Solver	Objective function:	$\max(L1_{utilization})$

4: We solve the constraint optimization problem representing the tiling and memory allocation. To guide the search, we use a L1 memory maximization heuristic and initialize the tensor dimension variables to their maximum values.

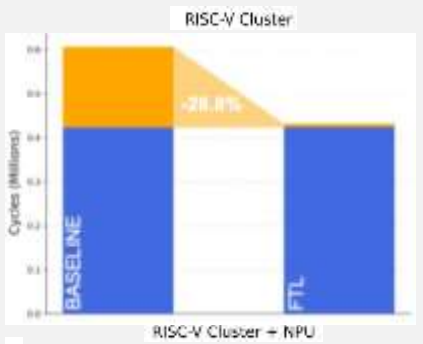
3 Results and Discussion

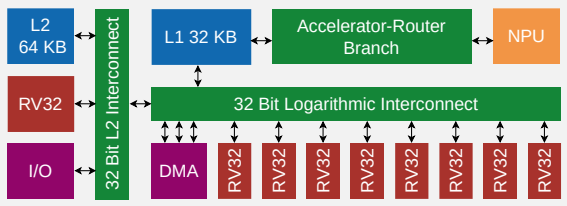
We perform our benchmark on a reduced version of the RISC-V Siracusa [1] SoC; its architecture is described below. The 8 RISC-V cores are using the *RV32IMCF-XPulpV2* ISA tailored to DSP tasks, and the NPU is targeting GEMM and convolution.

We benchmark a GEMM followed by a GeLU activation function. These layers are commonly found in the MLP stage of ViT [2]. There are two reasons to explain such runtime reduction:

- First, FTL reduces the number of DMA transfers by 47.1% by preventing the materialization of the MLP's intermediate tensor.
- Second, the L2 memory capacity is exceeded when materializing the MLP's intermediate tensor; hence, this tensor is stored in L3 RAM. With FTL, we don't need to perform costly off-chip memory transfers to bring back the intermediate tensor from L3 to L1, leading to a reduction of the runtime.

If double-buffering is used, FTL speeds up execution only if the kernel runtime is less than the DMA's runtime. As reported in the nearby figure, this is the case when using both the cluster and the NPU.





4 Conclusion

- We presented **Fused-Tiled Layers**, a *new* algorithm to fuse the tiling of several consecutive layers.
- We benchmarked Fused-Tiled Layer on a RISC-V heterogeneous SoC, for:
  - **60.1%** improvement in runtime.
  - **47.1%** reduction in off-chip transfers and on-chip data movement.

References

[1] A. S. Prasad, M. Scherer, F. Conti et al., "Siracusa: A 16 nm heterogenous RISC-V SoC for extended reality with at-MRAM neural engine," IEEE Journal of Solid-State Circuits, 2024.

[2] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. 9th Int. Conf. Learning Representations (ICLR), Austria, OpenReview.net, 2021.

SwissChips
pulp-platform
@pulp_platform
jungvi@iis.ee.ethz.ch

RISC-V Summit Europe 2025