Tiled Matrix Multiplication Unit for Edge Devices: Architecture, Implementation, and Floorplanning

Chao-Jia (Peter) Liu

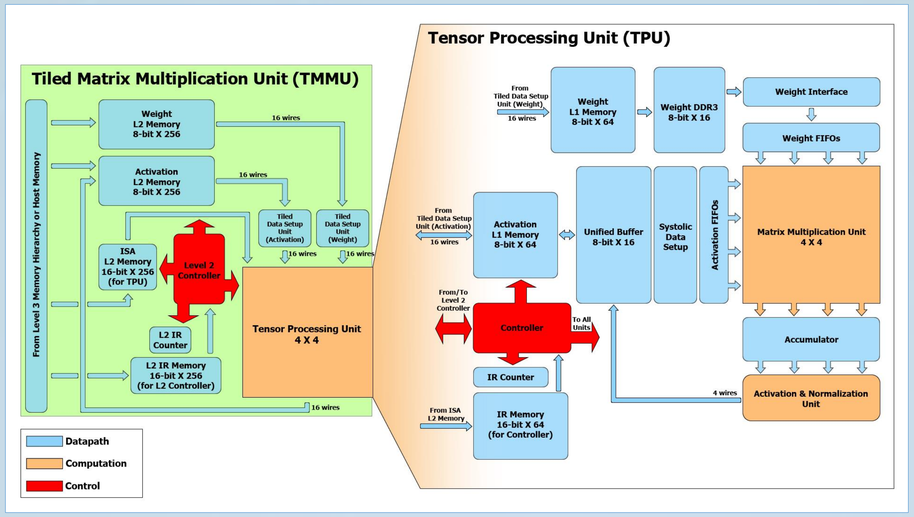
THE UNIVERSITY OF TEXAS AT SAN ANTONIO — UTSA

ELECTRICAL & COMPUTER ENGINEERING

Abstract

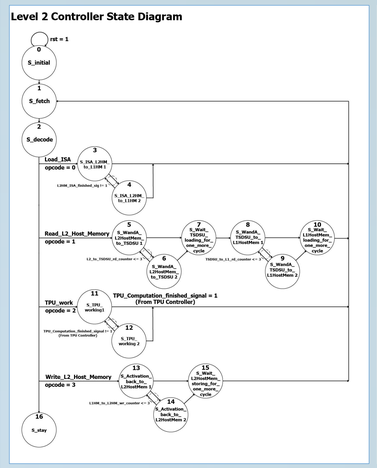
Tiled Matrix Multiplication (TMM) is an efficient technique for large-scale matrix operations, playing a crucial role in advancing state-of-the-art artificial intelligence (AI) applications on edge devices. TMM enhances computational efficiency while optimizing hardware resource utilization for large matrix computations. Many companies have integrated Tiled Matrix Multiplication Units (TMMUs) into modern advanced processors; however, the hardware design and implementation of these units remain proprietary and undisclosed. This study aims to unveil the architecture, implementation, and floorplanning of a TMMU, providing insights into its potential fabrication feasibility. The work includes Verilog-based hardware design, simulation results, and power estimation analysis, along with discussions on development challenges and future improvements. The proposed TMMU supports two operation modes: basic matrix multiplication and tiled matrix multiplication. Utilizing a customized CISC instruction set and leveraging a previously developed 4×4 Tensor Processing Unit (TPU), with a clock cycle time of 20 ns per cycle, the TMMU efficiently performs a 4×4 matrix multiplication in 3,000 ns (150 clock cycles) and an 8×8 matrix multiplication in 17,000 ns (850 clock cycles). Future optimizations are expected to enhance the design's performance, as this project serves as a foundational study for developing high-performance ASICs.

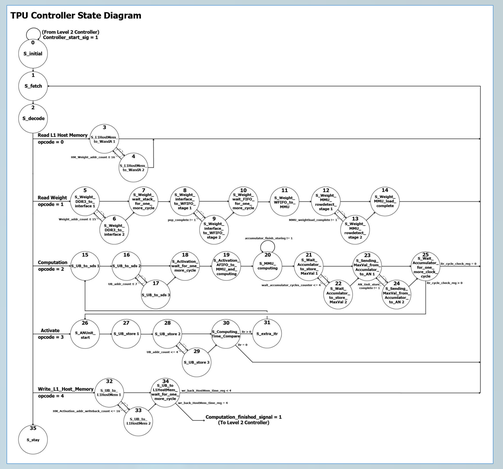
Block Diagram



The block diagram of the Tiled Matrix Multiplication Unit (TMMU) and the Tensor Processing Unit (TPU) illustrates the overall architecture. The light blue sections represent the datapath and data flow within the unit, while the orange blocks indicate the computation units, and the red blocks highlight the controller. It is important to note that the controller connects to all units in the design, although these connections are not explicitly shown in the diagram.

Controller State Diagram





The state diagrams for both the Level 2 controller and the TPU controller are presented. The Level 2 controller consists of 16 states, while the TPU controller operates with 35 states.

Customized CISC Instruction Set

Top Module (Tiled MM) Control Unit

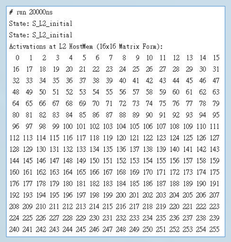
	ISA	IR[15:12]	IR[11:7]	IR[6]	IR[5:4]	IR[3:0]
1	Load ISA	0000	00000	0	00	0000
2	Read L2 Host Memory	0001	00000	Tiled MM defined bit	6 bits for computed matrix size	
3	TPU work	0010	00000	0	00	0000
4	Write L2 Host Memory	0011	00000	0	00	0000

TPU4x4 Controller

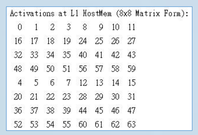
	ISA	IR[15:12]	IR[11:9]	IR[8]	IR[7:4]	IR[3:0]
1	*Read Host Memory*	0000	000	0	0000	0000
2	*Read Weight*	0001	000	0	0000	0000
3	*Computation*	0010	12 bits for the number of iteration			
4	*Activation*	0011	Activation Function Select	V/M Max	8 bits for User defined Factor	
5	*Write Host Memory*	0100	000	0	0000	0000

The customized CISC instruction set for the two controllers is shown in the diagram. The top section represents the Level 2 controller in the Tiled Matrix Multiplication Unit (TMMU), while the bottom section corresponds to the Tensor Processing Unit (TPU).

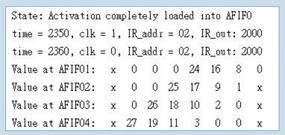
Simulation Result



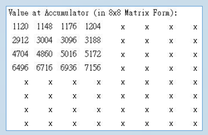
(1) Time = 0 ns, initial value in L2 Host Memory.



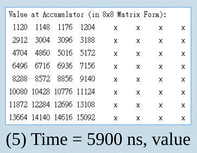
(2) Time = 1040 ns, value in L1 Host Memory.



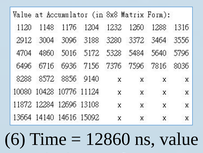
(3) Time = 2340 ns, value in Activation FIFOs.



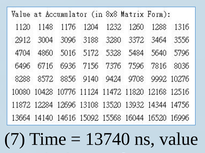
(4) Time = 5020 ns, value in Accumulator.



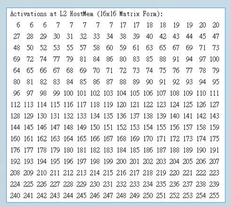
(5) Time = 5900 ns, value in Accumulator.



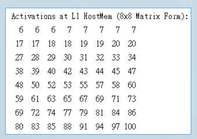
(6) Time = 12860 ns, value in Accumulator.



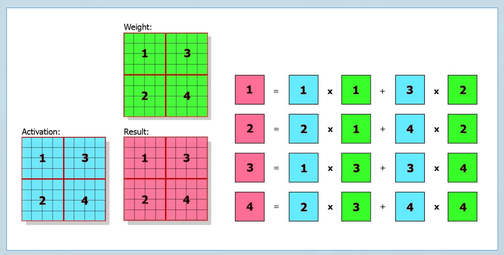
(7) Time = 13740 ns, value in Accumulator.



(9) Time = 17000 ns, final result in L2 Host Memory.

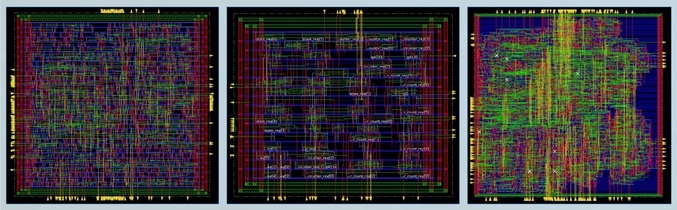


(8) Time = 16860 ns, final result in L1 Host Memory.



The simulation results illustrate an 8×8 tiled matrix multiplication within a specific timeline. The figure in the bottom-right corner provides a visual representation of the tiled matrix multiplication flow.

Innovus Floorplanning Result



The Floorplanning results from Innovus show the TPU controller (left), Level 2 controller (middle), and Matrix Multiplication Unit (right).

Conclusion

This project provides insights into designing an efficient tiled matrix multiplication unit, covering both architectural and implementation aspects. The final report includes:

1. A block diagram along with an explanation of the data flow.
2. Two state diagrams for the L1 and L2 controllers, with a detailed introduction to each state.
3. A customized CISC instruction set and the functionality of each bit.
4. Schematics generated by Cadence Innovus to illustrate the design structure
5. Power estimation for each component, generated by Xilinx Vivado, to evaluate performance.
6. Simulation results from Xilinx Vivado, demonstrating the data flow through each component and showcasing the tiled matrix multiplication process in hardware.
7. Floorplanning results generated by Cadence Innovus, showcasing potential fabrication opportunities.
8. Identified improvements and future work needed to optimize the project's results.

Overall, this study provides a foundation for advancing tiled matrix multiplication hardware in AI and edge computing.

References

[1] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Yoon, D. H. (2017, June). In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th annual international symposium on computer architecture (pp. 1-12).

[2] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

[3] L. Bottou et al., "Comparison of classifier methods: a case study in handwritten digit recognition," Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5), Jerusalem, Israel, 1994, pp. 77-82 vol.2, doi: 10.1109/ICPR.1994.576879.

[4] Ross, J., Jouppi, N., Phelps, A., Young, C., Norrie, T., Thorson, G.,Luu, D., 2015. Neural Network Processor, Patent Application No. 62/164,931.

[5] Ross, J., Phelps, A., 2015. Computing Convolutions Using a Neural Network Processor, Patent Application No. 62/164,902.

[6] Ross, J., 2015. Prefetching Weights for a Neural Network Processor, Patent Application No. 62/164,981.

[7] Ross, J., Thorson, G., 2015. Rotating Data for Neural Network Computations, Patent Application No. 62/164,908.

[8] Thorson, G., Clark, C., Luu, D., 2015. Vector Computation Unit in a Neural Network Processor, Patent Application No. 62/165,022.

[9] Young, C., 2015. Batch Processing in a Neural Network Processor, Patent Application No. 62/165,020.