

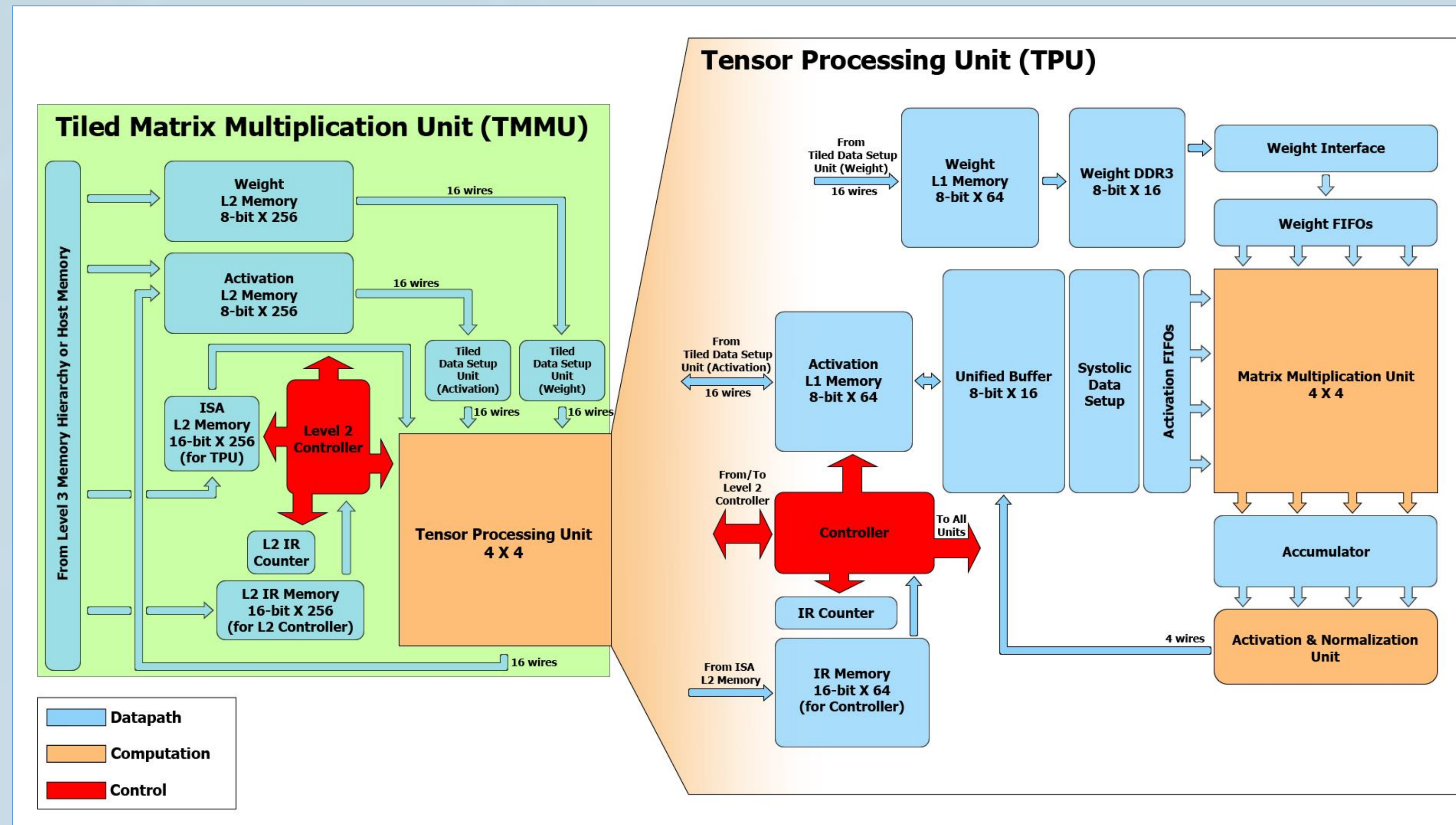
Tiled Matrix Multiplication Unit for Edge Devices: Architecture, Implementation, and Floorplanning

Chao-Jia (Peter) Liu

Abstract

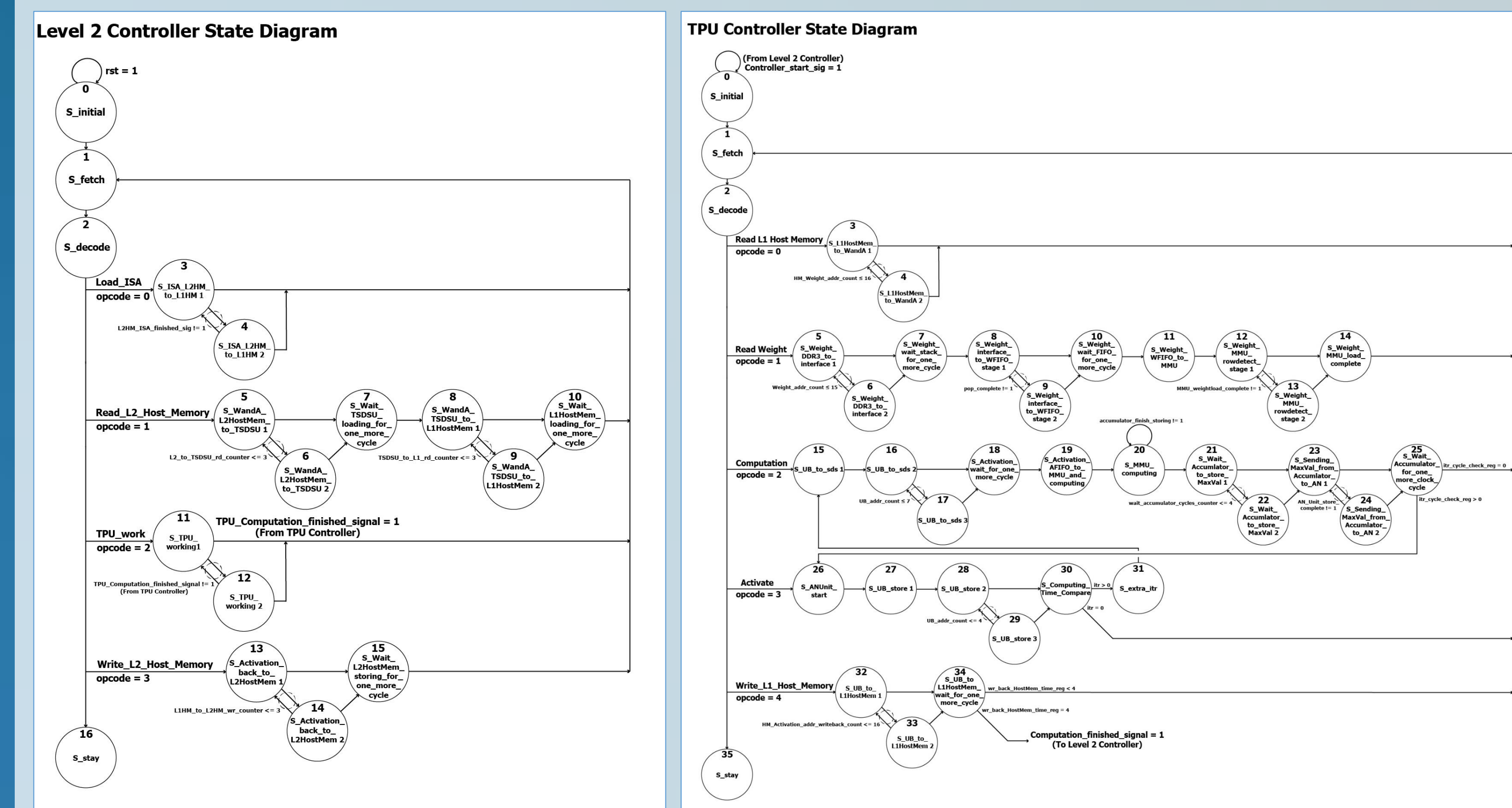
Tiled Matrix Multiplication (TMM) is an efficient technique for large-scale matrix operations, playing a crucial role in advancing state-of-the-art artificial intelligence (AI) applications on edge devices. TMM enhances computational efficiency while optimizing hardware resource utilization for large 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 4x4 Tensor Processing Unit (TPU), with a clock cycle time of 20 ns per cycle, the TMMU efficiently performs a 4x4 matrix multiplication in 3,000 ns (150 clock cycles) and an 8x8 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.

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.

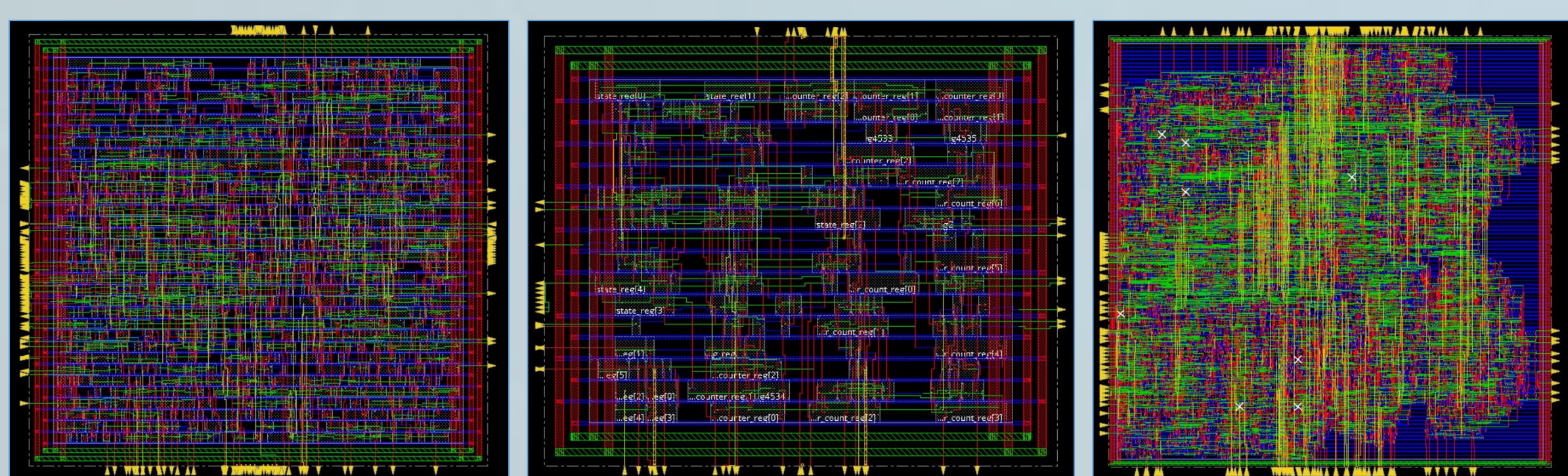
Customized CISC Instruction Set

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).

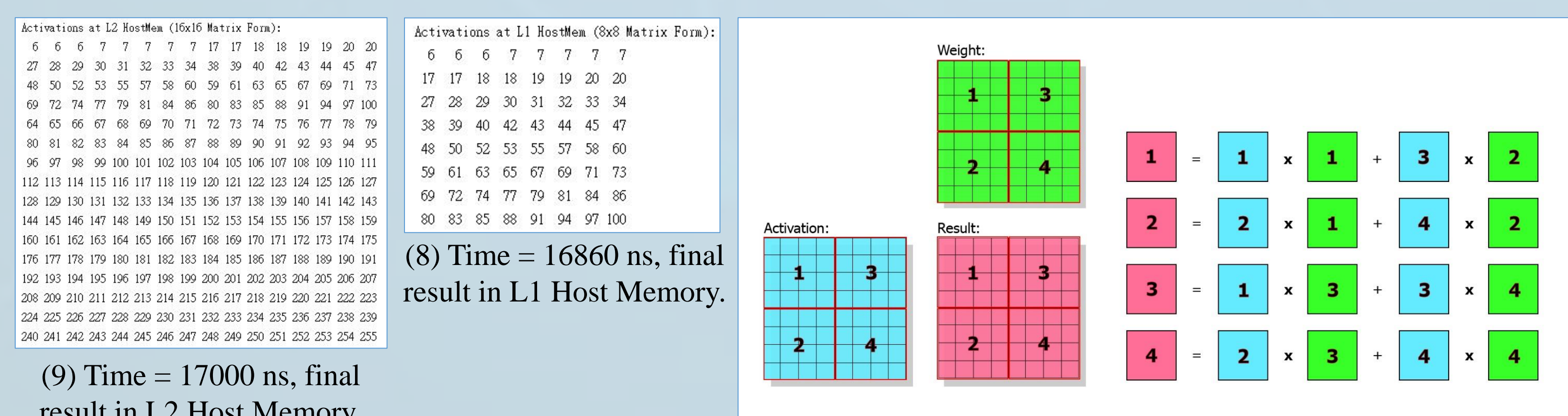
Innovus Floorplanning Result



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

Simulation Result

<div># run 2000ns</div> <div>State: S_L2_Initial</div> <div>State: S_L2_Initial</div> <div>Activations at L2 HostMem (16x16 Matrix Form):</div> <div>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</div> <div>16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</div> <div>32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47</div> <div>48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63</div> <div>64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79</div> <div>80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95</div> <div>96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111</div> <div>112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127</div> <div>128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143</div> <div>144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159</div> <div>160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175</div> <div>176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191</div> <div>192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207</div> <div>208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223</div> <div>224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239</div> <div>240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255</div>	<div>Activations at L1 HostMem (8x8 Matrix Form):</div> <div>0 1 2 3 8 9 10 11</div> <div>16 17 18 19 24 25 26 27</div> <div>32 33 34 35 40 41 42 43</div> <div>48 49 50 51 56 57 58 59</div> <div>64 65 66 67 72 73 74 75</div> <div>80 81 82 83 88 89 90 91</div> <div>96 97 98 99 104 105 106 107</div> <div>112 113 114 115 120 121 122 123</div> <div>128 129 130 131 136 137 138 139</div> <div>144 145 146 147 152 153 154 155</div> <div>160 161 162 163 168 169 170 171</div> <div>176 177 178 179 184 185 186 187</div> <div>192 193 194 195 200 201 202 203</div> <div>208 209 210 211 216 217 218 219</div> <div>224 225 226 227 232 233 234 235</div> <div>240 241 242 243 248 249 250 251</div>	<div>State: Activation completely loaded into AFIFO</div> <div>time = 2350, clk = 1, IR_addr = 02, IR_out: 2000</div> <div>time = 2360, clk = 0, IR_addr = 02, IR_out: 2000</div> <div>Value at AFIFO1: x 0 0 0 24 16 8 0</div> <div>Value at AFIFO2: x 0 0 25 17 9 1 x</div> <div>Value at AFIFO3: x 0 26 18 10 2 0 x</div> <div>Value at AFIFO4: x 27 19 11 3 0 0 x</div>	<div>Value at Accumulator (in 8x8 Matrix Form):</div> <div>1120 1148 1176 1204 x x x x</div> <div>2912 3004 3096 3188 x x x x</div> <div>4704 4860 5016 5172 x x x x</div> <div>6496 6716 6936 7156 x x x x</div> <div>8288 8572 8856 9140 x x x x</div> <div>10080 10428 10776 11124 x x x x</div> <div>11872 12284 12696 13108 x x x x</div> <div>13664 14140 14616 15092 x x x x</div>	<div>(1) Time = 0 ns, initial value in L2 Host Memory.</div>
<div>(5) Time = 5900 ns, value in Accumulator.</div>	<div>Value at Accumulator (in 8x8 Matrix Form):</div> <div>1120 1148 1176 1204 x x x x</div> <div>2912 3004 3096 3188 x x x x</div> <div>4704 4860 5016 5172 x x x x</div> <div>6496 6716 6936 7156 x x x x</div> <div>8288 8572 8856 9140 x x x x</div> <div>10080 10428 10776 11124 x x x x</div> <div>11872 12284 12696 13108 x x x x</div> <div>13664 14140 14616 15092 x x x x</div>	<div>(3) Time = 2340 ns, value in Activation FIFOs.</div>	<div>(4) Time = 5020 ns, value in Accumulator.</div>	
<div>(5) Time = 5900 ns, value in Accumulator.</div>	<div>Value at Accumulator (in 8x8 Matrix Form):</div> <div>1120 1148 1176 1204 1232 1260 1288 1316</div> <div>2912 3004 3096 3188 3280 3372 3464 3556</div> <div>4704 4860 5016 5172 5328 5484 5640 5796</div> <div>6496 6716 6936 7156 7376 7596 7816 8036</div> <div>8288 8572 8856 9140 9424 9708 9992 10276</div> <div>10080 10428 10776 11124 11472 11820 12168 12516</div> <div>11872 12284 12696 13108 13530 13932 14344 14756</div> <div>13664 14140 14616 15092 15568 16044 16530 16996</div>	<div>(6) Time = 12860 ns, value in Accumulator.</div>	<div>(7) Time = 13740 ns, value in Accumulator.</div>	



The simulation results illustrate an 8x8 tiled matrix multiplication within a specific timeline. The figure in the bottom-right corner provides a visual representation of the tiled matrix multiplication flow.

References

- [1] Jouppe, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Yoon, D. H. (2017, June). In-datascenter 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., Jouppe, 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.