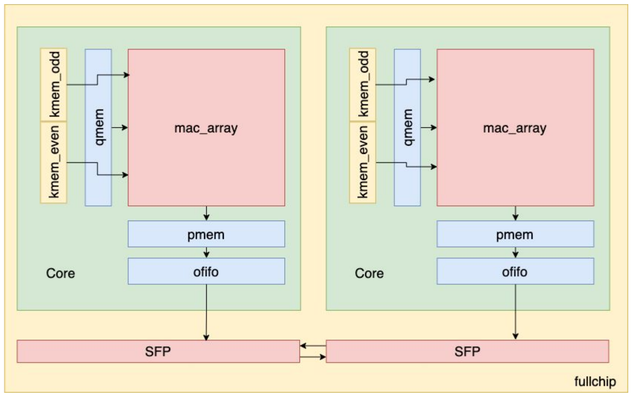
Dual core machine learning accelerator for attention mechanism

Sin-Yu Chen, Yu-Hsiang Tseng, and Ya-Chi Liao (Group Verilog Vanguard)
University of California, San Diego, USA

Motivation

In light of the growing requirements for machine learning capabilities, it is imperative to enhance the adaptability, performance, and energy efficiency of ML accelerators. Achieving higher performance with low power consumption is crucial for scalability. To tackle this challenge, we have developed an dual core ML accelerator with the objective of attaining better throughput and energy efficiency in comparison to conventional scalar processing architectures.

Basic Design



	Basic Design	+Alpha 1
Frequency(GHz)	1	1
# of Logic Elements	17642	16976
Dynamic Power (mW)	48.5401	47.0033
Total Power	49.5499	47.97
Combinational Area	79649.27	76494.95

Performance after Synthesis

Alpha 1 – SRAMs Parallel Read

Q SRAM and K SRAM have no inherent dependencies. Therefore, by partitioning their input and address, we can readily parallelize both their writing and reading processes..

Alpha 2 – Double Buffering

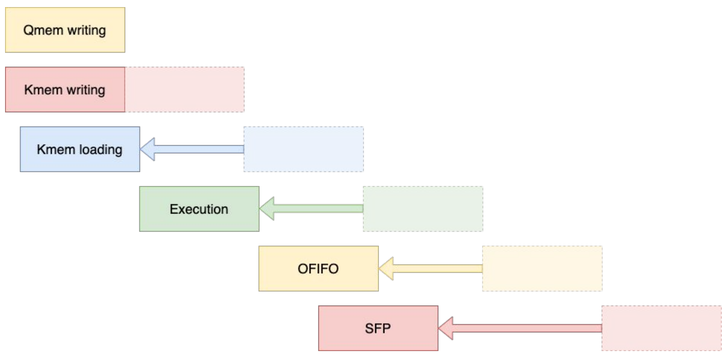
In the initial design, loading K data could start only after all K values had been written into the SRAM. Nevertheless, through the implementation of double buffering, we can simultaneously read K data from one SRAM while writing new K data to another.

Alpha 3 – SFP Pipelining

In the provided controller, data transmission to the special function processor begins only after all matrix multiplication results are stored in the OFIFO. However, for the final Q data, correct value collection begins immediately upon its arrival at the mac_array, allowing us to commence writing the results to the OFIFO immediately after the final Q data has been sent. The diagram below illustrates the overall high-level pipeline design, while the figure on the right provides a rough estimation of the total speedup.

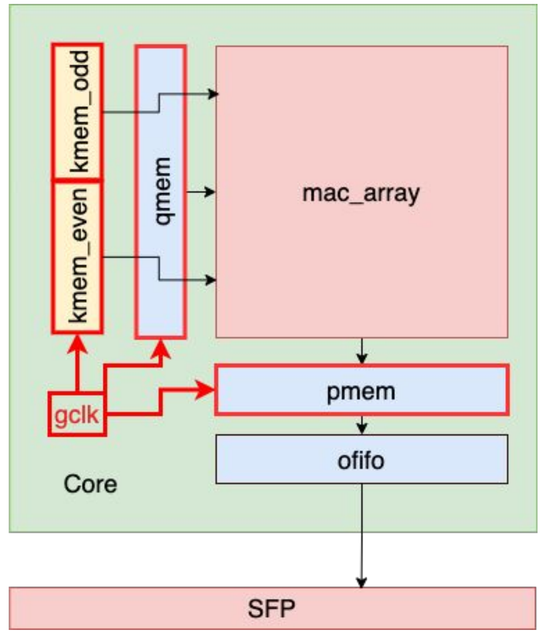
Parallelism Result

Method	Cycle Count	Speedup
Vanilla	20*num_col	
Parallel SRAM writing	18*num_col	~11%
SRAM double buffering	16*num_col	~25%
Execution SFP pipelining	14*num_col	~67%



Alpha 4 – Clock Gating

To conserve power and prevent wastage, we implemented clock gating techniques. Specifically, we introduced gated clocks to all SRAMs, as their control signals are comparatively easier to identify, and they tend to remain idle for significant periods.



Hierarchical Synthesis of Dual Core – still testing on different dimensions to maximize the density