Santhosh Ramachandran

Assignment 5 – Implementation & Report of DLP Techniques

Dr. Lively

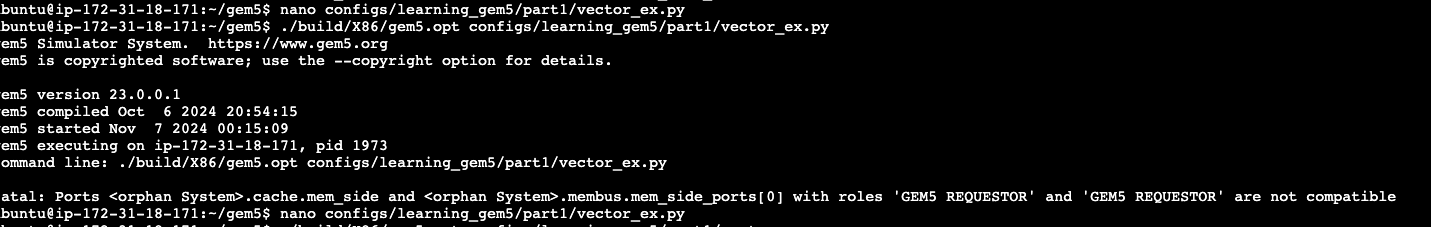
**Troubleshooting steps while setting up a custom CPU**

Port error:

A screen shot of a computer

Description automatically generated

Different port error:



Memory size of cache issue:

A computer screen shot of a black screen

Description automatically generated

Workload issue:  
A computer screen with white text

Description automatically generated

For simulating the vector architecture with SIMD multiplication, I created a custom Vector CPU that inherits DeriveO3CPU. I also implemented a custom FUPool with SimdMult opClass, and opLat up to 4. To simulate an intensive operation, I created a matrix multiplication program in C. Below is the output of the multiplication and performance metrics compared with a normal architecture:

Output

A screenshot of a computer

Description automatically generated

Performance metrics

A graph on a paper

Description automatically generated

A graph on a graph paper

Description automatically generated

A graph on a graph

Description automatically generated

**Loop level parallelism**

To simulate loop level parallelism, I created a C program which performs loop operations that are independent of each other and can be parallelized. I generated the binary file for the loop code and ran it with and without loop level parallelism. For applying loop level parallelism, I used O3CPU since it can detect loops and other parallelization oppurtunities that can be ran in parallel and be performant. Below is the output and the metrics:

Output:

A screenshot of a computer

Description automatically generated

**Performance metrics**

For each of the following metric, we can see how the loop level parallelism can achieve significantly better results for the same loop program.

A graph on a graph paper

Description automatically generated

A graph with text on it

Description automatically generated

A graph with numbers and a bar

Description automatically generated