**ECE 462/562   
Homework 2 Deliverables**

**Due: Friday, March 20, 11.59pm.**

**Group members (and netID) who participated:**

|  |  |
| --- | --- |
| **Name** | **NetID** |
| **Mitchell Dzurick** |  |
| **Lena Voytek** |  |
| **Amir** |  |
|  |  |
|  |  |

1. In-order vs Out-of-Order performance impact.

|  |  |  |  |
| --- | --- | --- | --- |
| **Benchmark** | Simulation seconds in-order | Simulation seconds Out-of-order | % time improvement with out-of-order |
| a2time01 |  |  |  |
| cacheb01 |  |  |  |
| bitmnp01 |  |  |  |
| mcf |  |  |  |
| libquantum |  |  |  |
| AVERAGE |  |  |  |

For the benchmarks that benefited the most from OoO execution, why do you think OoO worked as well for those benchmarks?

1. ROB impact

(Copy and paste your graph here. The y-axis should be the “ROB normalized to the base”, and the bars on the x-axis should represent 20, 40, 80 for each benchmark. Smaller is better; i.e., bars above 1 represent configurations that are worse than the base, while bars below 1 represent configurations better than the base.)

1. LSQ impact

(Copy and paste your graph here. The y-axis title should be the “LSQ normalized to the base”, and the bars on the x-axis should represent 4, 8, 32 for each benchmark. Smaller is better; i.e., bars above 1 represent configurations that are worse than the base, while bars below 1 represent configurations better than the base.)

Discuss with your group members your observations from the results about the impacts of ROB vs. LSQ on the different benchmarks explored.