**Final Report**

**A-1: Describe in 100 words or less how the provided framework and its components**   
**enable a design space exploration.**

In this project, it provided us the Workload, Design Parameters, Simple Simulator, and System statistic for us to build this DSE design. The Workloads are five benchmarks that provided to test the program. The Design Parameters consist of 4 sections which are BP, Cache, FPU, and Cores. There are total 18 parameters or dimansions need to explore. The Simple Simulator are designed to loop through all the possiable configurations and return the best configuration based on performance and energy-delay cost. And then, System statistics are the outptus that contain number of instructions, number of cycles, CPI, and IPC etc. In each iteration, the Simple Simulation will give us the best configuration with statistics of best EnergyEfficiency and ExecuteTime.

**A-2: List the design point chosen by your DSE.**

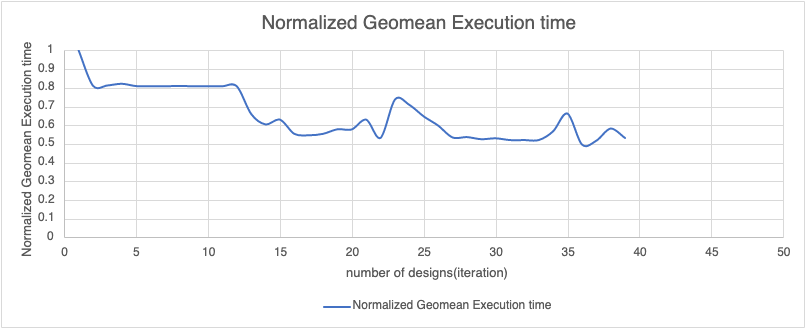
My PSU ID = 962361481. 962361481 % 24 = 1. So my exploration order is BP, Cache, FPU, and Core.

**A-3: Table**

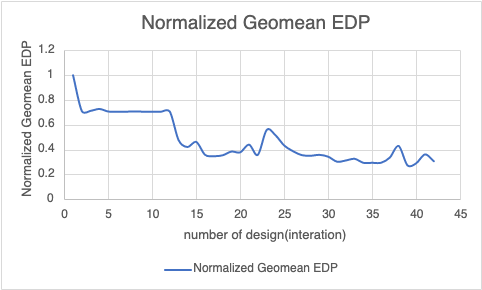
|  |  |  |
| --- | --- | --- |
| Parameter | Performance | EDP |
| width | Index = 0  Value = 1  Why: smaller fetch width decrease the CCT so that increace the performance. | Index = 0  Value = 1  Why: smaller fetch width has lower core leakage power. |
| scheduling | Index = 0  Value = -issue:inorder true -issue:wrongpath false  Why: inorder scheduling decreae CCT so that increase the performance. | Index = 0  Value = -issue:inorder true -issue:wrongpath false  Why: inorder scheduling with low width generate less power according to 8.2.4 in PDF |
| l1block | Index = 2  Value = 32  Why: larger block size can utilize spacial locality better which can increase the hit rite of cache which can improve the performance | Index = 2  Value = 32  Why: since miss in cache will goes to memory which will cause extra energy use. Larger block size utilize spacilly locality better to increase the hit rate so that it can lower energy consumption |
| dl1sets | Index = 2  Value = 128  Why: more sets can increase the hit rate of cache which improve the performance. | Index = 2  Value = 128  Why: smaller sets can reduce the energy comcumption. |
| dl1assoc | Index = 0  Value = 1  Why: small association has less complexity of logic which can improve the performance | Index = 0  Value = 1  Why: small association has less complexity of logic which can reduce the energy comsumption |
| il1sets | Index = 6  Value = 2048  Why: more sets can increase the hit rate of cahce which improve the performance | Index = 5  Value = 1024  Why: smaller sets can reduce the energy comcumption. |
| il1assoc | Index = 0  Value = 1  Why: small association has less complexity of logic which can improve the performance | Index = 0  Value = 1  Why: small association has less complexity of logic which can reduce the energy comsumption |
| ul2sets | Index = 2  Value = 1024  Why: more sets can increase the hit rate of cahche which can improve the performance | Index = 1  Value = 512  Why: smaller sets can reduce the energy comcumption. |
| ul2block | Index = 3  Value = 128  Why: larger block size can utilize spacial locality better which can increase the hit rite of cache which can improve the performance | Index = 3  Value = 128  Why: since miss in cache will goes to memory which will cause extra energy use. Larger block size utilize spacilly locality better to increase the hit rate so that it can lower energy consumption |
| ul2assoc | Index = 1  Value = 2  Why: small association has less complexity of logic which can improve the performance | Index = 1  Value = 2  Why: small association has less complexity of logic which can reduce the energy comsumption |
| replacepolicy | Index = 0  Value = “L”  Why: LRU replacement police can track and replace the data fast to improve the performance | Index = 0  Value = “L”  Why: LRU has less complexity and faster access which makes it consume less energy |
| fpwidth | Index = 0  Value = 1  Why: smaller fpwidth can reduce the CCT which can improve the performance | Index = 0  Value = 1  Why: samller fpwidth will have less core leakage power according to 8.2.1 in PDF |
| branchsettings | Index = 4  Value = "-bpred comb -bpred:comb 1024"  Why: this branchsetting can predict branch instruction to improve the performance | Index = 4  Value = "-bpred comb -bpred:comb 1024"  Why: this branch setting reduce flash instruction in total which can reduce the total CCT such that lower the energy usage. |
| ras | Index = 3  Value = 8  Why: large ras size can increase the accuracy and speed of return address which can improve the performance | Index = 3  Value = 8  Why: large ras size can increase the accuracy and speed of return address which can reduce the energy consumption |
| btb | Index = 2  Value = "512 4"  Why: larger btb can track more branch predtction which can improve the performance | Index = 0  Value = "128 16”  Why: smaller size of btb use less power |
| dl1lat | Index = 1  Value = 2  Why: this is the best latency value that calculated from dl1 cache settings for better performance | Index = 1  Value = 2  Why: this is the best latency value that calculated from dl1 cache settings for less energy consumption |
| il1lat | Index = 5  Value = 6  Why: this is the best latency value that calculated from il1 cache settings for better performance | Index = 4  Value = 5  Why: this is the best latency value that calculated from il1 cache settings for less energy consumption |
| ul2lat | Index = 4  Value = 9  Why: this is the best latency value that calculated from ul2 cache settings for better performance | Index = 3  Value = 8  Why: this is the best latency value that calculated from ul2 cache settings for less energy consumption |

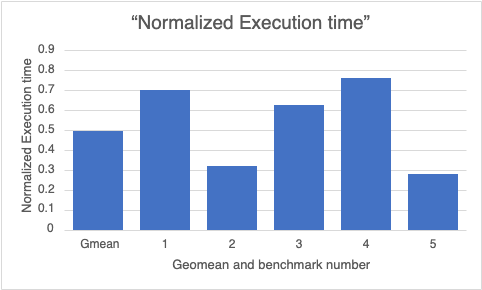
**A-4: Plots**

**A. Line plot of normalized geomean execution time (y axis) for each considered de-**   
**sign point vs. number of designs considered (x axis)**

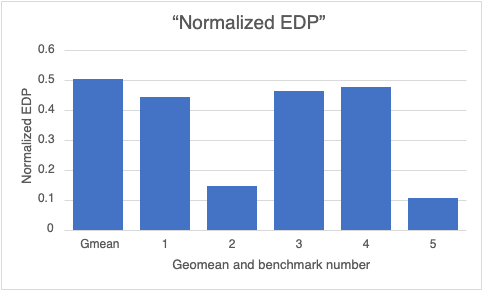


**B. Line plot of normalized geomean of energy-delay product (y axis) vs number of**   
**designs considered**



**C. Bar chart showing normalized per-benchmark execution time and geomean nor-**   
**malized execution time for the best performing design**  


**D. Bar chart showing per-benchmark normalized energy-delay product and geomean**   
**normalized energy delay product for the most energy-efficient design found**



**A-5: Describe a more sophisticated heuristic which you expect will perform design**   
**space exploration (limited by 1000 design points) more effectively to find a better**   
**performing design (with respect to execution time).**

According to my PSU ID, I need to explore BP first, Cache seconf, FPU fhird, and Core at the last. Based on the simple heuristic that already build in the project. I expect to do some string manipulation to reorder my configutation first to set BP to the left most and then, follow by Cache, FPU, and Core. And When exploring all the things in each dimensions. In the original heuristic, it always starts from index 0 to explore dimensions. However, we can not do that since we have our baseline configuration. If we explore one dimension that does not start from 0. We need to make sure to loop back to explore all possible index. Moreover, In the simple simulator, we have 1000 interations to do every time. We can have a global flag to track the exploring situation such that we can know the timw to exit the for loop ealier to reduce the execution time.

**A-6. Elaborate on any 2 new insights you gained while working on this project.**

1. DSE is a complicating design for me so far. There are so many parameters can affect the performance and energy-delay product. And string manipulation in C++ is a new thing to me, and I learn some of string in this project.

2. It is surprised that in my result. The assocation of each level of cache has a small number for better performance and EDP. I used to think that accociation can increase the hit rate of cache which can improve the performance in theory. However, based on my result, small accociation can have better performance and EDP. I think the reason is that more number of ways of accociation will increase the conplexity of the logic design which might have a negative effect on performance and EDP.