# DAT105 Computer Architecture: Lab 3: Power-Performance Trade-offs

Max Villing, Vinaykumar M K, Lab Group 4

September 2021

## 1 Project Goals

The goal of this project is to gain an understanding on the impact processor hardware has on processor power consumption. Additionally analyses will be made on the trade offs between power efficiency and hardware performance.

## 2 Methodology

This project was conducted with the use of the Sniper simulator. Sniper is able to analyze the performance of differing processor configurations for given benchmark programs. For this project 3 benchmark programs were used, qsort, gsm-untoast and jpeg-cjepg. Sniper can also optionally record data on power consumption of the processor when executed with the --power option. Said option was always used during the course of this project. By comparing simulator results for differing configurations it was possible to observe the impacts they had on performance and power efficiency.
Note that due to a simulator being used for this project and not real hardware these results may differ from what would occur in real world circumstances. Additionally there is the possibility of software errors in Sniper resulting in incorrect results. In particular the simulator provided inconsistent instruction counts for consecutive executions of the same benchmark program, which casts some doubts on the accuracy of the performance results. We were instructed by the lab assistants to ignore these errors and use the same instruction count values for all calculations.

## 3 Observations

### 3.1 Processor Configurations

Before differing configurations were tested the simulator was run with default values to provide a baseline for use in comparisons. Afterwards based on results from previous projects we configured an alternative High Performance Processor (HPP) that sought only to maximize performance. Project constraints meant we were restricted to specific ranges of values for the processor configuration. Since previous projects had shown that more hardware units gave higher performance we opted for the highest possible value for each parameter. Later during the course of the project we were asked to design a more power efficient version of the HPP, this configuration sacrificed performance to fit a specific budget and was titled HPP+PW. Finally we designed a processor that aimed only to achieve an ideal Energy-Delay-Product (EDP) and was titled E-eff. EDP is an attempt to combine performance and energy efficiency into one metric. Since the goal for E-eff was only to achieve a lower E-eff we opted to not give it the ability to perform instruction level parallelism, as while said feature gives high performance it also greatly amplifies power use. The configurations for each processor can be seen in the table below:

| Parameter | Possible values | Base | HPP | HPP+PW | E-eff |
|---|---|---|---|---|---|
| in_order | True/False | TRUE | FALSE | FALSE | TRUE |
| Outstanding loads and stores (LSQ) | 1,8,6 | 1 | 16 | 16 | 16 |
| RS entries | 1,16,32 | 1 | 32 | 32 | 16 |
| Interval timer window size (ROB) | 16,64,128 | 16 | 64 | 64 | 64 |
| Dispatch & Commit width | 1,4,8 | 1 | 8 | 7 | 1 |
| L2 cache size (KB) | 4,8,16,32 | 4 | 32 | 32 | 32 |
| L2 cache associativity | 1,2,4,8 | 1 | 2 | 2 | 2 |
| block size(B) | 16,32,64 | 16 | 64 | 64 | 64 |

## 3.2 Performance

The base, HPP and HPP+PW configurations described above were tested on the 3 benchmark programs to observe their performance. The results of the tests can be observed in the table below:

| | Application | qsort | gsm-untoast | jpeg-cjpeg | GM |
|---|---|---|---|---|---|
| Base | Instruction Count* | 15468357 | 7900888 | 23179222 | NA |
| | Execution Time | 0.4524 | 0.0743539 | 0.3516 | NA |
| | CPIbase | 50 | 20 | 33.33333333 | NA |
| HPP | Execution Time | 0.031 | 0.0051 | 0.0158 | NA |
| | CPIbase | 4 | 1.298701299 | 1.369863014 | NA |
| | SP | 12.5 | 15.4 | 24.33333333 | 3.738085643 |
| HPP+PW | Execution Time | 0.0297 | 0.011 | 0.0161 | NA |
| | CPIbase | 3.846153846 | 1.408450704 | 1.388888889 | NA |
| | SP | 13 | 14.2 | 24 | 3.713271067 |

We find that the HPP configuration performs best which is to be expected. The difference in CPI for qsort between HPP and HPP+PW is likely due to problems with the simulator. Given that the only difference between HPP and HPP+PW is a slightly decreased dispatchcommit width we find the similarities in their results to be logical. We note that HPP+PW still performs far better than the base configuration. We also note that the L2 cache has a significant impact on performance, though not as high as the L1 cache had in previous projects.

## 3.3 Energy Efficiency

For each execution of the benchmark programs we also recorded power consumption data. The results for the base, HPP and HPP+PW configurations can be seen in the table below:

| | Application | qsort | gsm-untoast | jpeg-cjpeg |
|---|---|---|---|---|
| Base | Area | 7.32612 | 7.32612 | 7.32612 |
| | Peak Dynamic Power | 1.50923 | 1.53368 | 1.52224 |
| | Runtime Dynamic Power | 0.425039 | 0.443347 | 0.435929 |
| | Subthreshold Leakage Power | 0.499951 | 0.499951 | 0.499951 |
| | Energy | 0.418465476 | 0.07013788516 | 0.329055408 |
| HPP | Area | 17.1648 | 17.1648 | 17.1648 |
| | Peak Dynamic Power | 11.5726 | 12.5943 | 12.9613 |
| | Runtime Dynamic Power | 0.960947 | 1.87859 | 1.99169 |
| | Subthreshold Leakage Power | 0.803737 | 0.803737 | 0.803737 |
| | Energy | 0.054705204 | 0.0136798677 | 0.0441677466 |
| | Area increased (%) | 2.342959165 | 2.342959165 | 2.342959165 |
| | Energy increased(%) | 0.1307281177 | 0.1950424891 | 0.1342258645 |
| HP+PW | Area | 15.49 | 15.49 | 15.49 |
| | Peak Dynamic Power | 9.17176 | 10.0257 | 10.4787 |
| | Runtime Dynamic Power | 0.91281 | 1.59603 | 1.77967 |
| | Subthreshold Leakage Power | 0.763197 | 0.763197 | 0.763197 |
| | Energy | 0.0497774079 | 0.025951497 | 0.0409401587 |

We note that while both HPP and HPP+PW have very similar performance (see table in the previous subsection) they have very differing energy consumption. From this we gather that performance improvements do not scale linearly with energy consumption. We also note that dispatchcommit width has a large impact on power consumption as even though we only decreased it by 1 for HPP+PW it resulted in a notable increase in efficiency.

### 3.4 EDP

Finally we also measured the EDP for the base and HPP configurations. Based on those results we constructed a new configuration, E-eff, that sought to minimize EDP. EDP for all results can be seen in the table below:

| | Configuration | qsort | gsm-untoast | jpeg-cjpeg |
|---|---|---|---|---|
| EDP | BASE | 0.1893137813 | 0.0052150253 | 0.1156958815 |
| | HPP | 0.001695861324 | 0.00006976732527 | 0.0006978503963 |
| | E-eff | 0.0005412759817 | 0.00002980130052 | 0.0004482257199 |

From these results we find that while parallelism results in major performance improvements it is fully possible for a non parallel processor to have a better EDP value than a parallel one. Parallelism, and the hardware needed to support it, comes at a significant cost that non parallel processors such as the base and the E-eff does not need to worry about.

## 4 Conclusion

During the course of this project we have gained a deeper understanding of processor power consumption. We have seen how higher performance can incur disproportionate loses in efficiency and that parallelism has a notable impact on EDP. The growing needs for computer performance push for the industry to move more and more towards parallel processors and architectures. The challenge then for architects is to find ways to maximize the performance gains from parallelism while minimizing the costs associated with it.