The first assignment and the term paper are on a common topic: Comparing the Intel and AMD processor architectures. You can do the Assignment and the associated Term paper in a Group of at most 2 students. The same set of students should work together on this assignment and the term paper. Both are due on Oct. 8, 2021. Each group should submit two reports one for the assignment and another for the term paper. Assignment submission would also require submission of necessary scripts/programs etc.

The assignment (15 marks) and the term paper (10 marks) are somewhat open-ended and hence would be evaluated for the content, quality and quantum of work. Study/discussion which brings out not-so-routine insights and demonstrates deeper understanding would be appropriately rewarded.

Assignment #1

Comparing Intel and AMD Processors

In this assignment you will conduct a performance comparison of an Intel Sky Lake (or more recent) processor and an AMD Zen processor from the microarchitecture and its performance viewpoint, in lines similar to (but not limited by) the HPCA 2013 paper, "Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures" [1]
(https://research.cs.wisc.edu/vertical/papers/2013/hpca13-isa-power-struggles.pdf)
While the paper focuses on performance and power measurement on a variety of benchmark suites and different processors, this assignment would focus on performance study on a subset of SPEC benchmarks. You will use the Performance Monitoring Counters (PMC) available on the architecture.

You can compare the microarchitecture of the two processors using appropriate performance metrics, such as of the execution time, execution cycles, number of instruction retired, number of micro-ops retired, L1/L2/L3 cache accesses and misses, retired branch instructions, branch misses, L1/L2 TLB misses, page faults, front-end stalled-cycles, backend stalled cycles, ROB stalls, etc.

The counts of various miss events occuring during program execution can be obtained using hardware Performane Monitoring Counters (PMC) on modern architectures. (See e.g., [2] and [3] for details regarding PMCs available on Intel Skylake and AMD Ryzen processors). Performance counter values for running application can be obtained using performance monitoring tools such as PAPI [4] or `perf` [5]. Additionally, you can also make use of Intel's Vtune [6] or AMD's uProf [7].

You are required to run a set of programs (3 SPEC Int and 3 SPEC FP benchmark programs) on 2 different laptops/desktops, one consisting an Intel Sky Lake (or more recent) processor and another containing an AMD Ryzen processor. In each case choose the appropriate input for the program.

(a) Based on the resullts achieved, list at least 5 key findings/observations comparing the two processors. Justify your observations with appropriate performance results and analysis.

(b) In this experiment, you will find how some of the higher-level performance metrics given by Vtune or AMDuProf correlate to specific performance counters. Consider the following list:

Front-end Bound	itlb_misses.miss_causes_a_walk, itlb_misses.stlb_hit, itlb.itlb_flush, branch-misses, icache_16b.ifdata_stall, frontend_retired.latency_ge_4, frontend_retired.l1i_miss
L1 Bound	uops_dispatched_port.port2, DTLB_LOAD_MISSES.MISS_CAUSES_A_WALK, DTLB_LOAD_MISSES.MISS_CAUSES_A_WALK, mem_trans_retired.load_latency_gt_4
Port Utilization	bus-cycles, branch-misses, DTLB_LOAD_MISSES.MISS_CAUSES_A_WALK, mem_load_retired.l2_miss, mem_load_retired.l3_miss, uops_dispatched_port.port2

Each of these higher-level performance metric (Front-end bound, L1-bound, or Port Utilization), may be correlated to one or more of the performance counters listed against them in the two architectures. By conducting appropriate experiments and performance measurements done on the subset of SPEC benchmarks, identify the correlated performance counters. Provide suitable explanation for your findings.

(c) The number of dispatched instructions (mesured in micro-ops) is always higher than that of the retired micro-ops. In some sense this reflects wasted work due to speculation. For the subset of SPEC benchmarks determine the percentage of wasted work across different programs. Write a simple (micro)benchmark involving D-cache misses and/or branch misprediction which will increase the wasted work significantly. Report the percentage of wasted work and relate that to the microbenchmark and microarchitecture parameters (such as the ROB size, cache miss penalty, misprediction latency, etc.) that affect them.

You are required to submit a report (not exceeding 8-pages) along with the programs/scipts you have written to conduct the experiments. If required, your scripts/program would be run to validate your experimental results.

A few remarks/points to be noted are listed below:

i. As the performance counters available on a processor differ across generation of processors, you can go through the processor manual to know the list of available counters specific to your processor generation.
ii. In order to read these performance counters, performance monitoring tools like perf and PAPI can used in Linux environment. Perf tool presents a simple command line interface and can be installed as a linux utility tool using the command
"apt-get install linux-tools-common linux-tools-generic linux-tools-`uname -r`"
iii. install linux-tools corresponding to your exact Linux Kernel version. You can read more about perf at https://perf.wiki.kernel.org/index.php/Tutorial
iv. PAPI provides APIs that can be inserted in the source code to query the performance monitoring counters.
v. You can find more information about PAPI and installation at https://icl.utk.edu/papi/
vi. AMDuProf can be downloaded from https://developer.amd.com/amd-uprof/ and for installion refer to [7].
vii. When you run the application on your laptop/desktop and gather performance counter values, make sure that no other processes (like browsers, video players, other C programs etc.) are running on your system/or on the specific core as they could influence the counter values. You can also pin your process to a specific core and run the above tools to read counters on that core. Also ensure you run all the experiments in the same system while collecting the performance counters

References:

[1] Emily Blem, Jaikrishnan Menon, and Karthikeyan Sankaralingam, "Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures", 19th IEEE International Symposium on High Performance Computer Architecture (HPCA 2013)

[2] Intel® 64 and IA-32 Architectures Software Developer's Manual - Volume 3B: System Programming Guide, Part 2

[3] AMD Ryzen Performance Counters: https://developer.amd.com/wordpress/media/2017/11/54945_PPR_Family_17h_Models_00h-0Fh.pdf

[4]PAPI User's Guide

[4]PAPI User's Guide. Available at: http://icl.cs.utk.edu/projects/papi/files/documentation/PAPI_USER_GUIDE_23.htm

[5] Linux Kernel Profiling with perf. Available at: https://perf.wiki.kernel.org/index.php/Tutorial

[6] Intel Vtune Profiler User Guide https://software.intel.com/content/dam/develop/external/us/en/documents/vtune-profiler-user-guide.pdf

[7] AMD uProf. User Guide https://developer.amd.com/wordpress/media/files/AMDuprof_Resources/User_Guide_AMD_uProf_v3.4_GA.pdf

Term Paper #1

Comparing Intel Ice Lake and AMD Ryzen Processors

In this term paper you will compare the microarchitectre of Intel Ice Lake processor with AMD Ryzen processor. Refer to [1-4] and also various internet blogs on these processors. Only a few pointers to internet blogs are given below, so do not limit yourselves only to these!

You are required to compare various microarchitecture components such as cache hierarchy, branch predictor, micro-ops cache, issue width, reorder buffer, integer/floating point/load store functional units, vector instruction capabilities, etc. of these two architectures from the information available on the internet. For example, you would come across converting instructions to micro-ops, micro-op fusion, micro-op cache, etc. Similarly the branch predictors they have may be different from what you have seen in the course. Your term paper should bring out your understanding of these

The emphasis of the term paper should be on understanding the design and their implications on performance, *rather than on copy-paste*! Focus primarily on the microarchitecture of a single core. Wherever possible, explain the intuition/ reasoning behind the design decisions and how they would impact the performance. Wherever applicable, you can justify your answers with the quantiative performance study obtained from your Assignment #1.

You are required to submit a report (less than 8 pages). You will be graded on the completeness of your study, how well it reflects your understanding, the accuracy of your summaries, the supporting evidence for your opinions (cite appropriate references wherever applicable), and the quality of your report.

[1] David Suggs, Mahesh Subramony, and Dan Bouvier, "The AMD Zen 2 Processor", IEEE Micro, March/April 2020.
[2] Anandtech "AMD Zen 2 Microarchitecture Analysis", https://www.anandtech.com/show/14525/amd-zen-2-microarchitecture-analysis-ryzen-3000-and-epyc-rome
[3] AnandTech, "Examining Intel's Ice Lake Processor", https://www.anandtech.com/show/14514/examining-intels-ice-lake-microarchitecture-and-sunny-cove/
[4] https://www.agner.org/optimize/microarchitecture.pdf