Algorithm Engineering Lab Assignment 8

Brian Zahoransky (brian.zahoransky@uni-jena.de)

February 22, 2021

1. Explain the naming conventions for intrinsic functions.

_<vector_size>_<operation>_<suffix>

Commonly, a name for an intrinsic procedure consists of three parts separated by an underscore, as shown above. The first part indicates the **size of the returned vector**. Typically, its "mm", "mm256", or "mm512" which stands for 128-bit, 256-bit, or 512-bit vectors.

The second part specifies the **operation** of the intrinsic procedure. Operations like add, sub, and mul were listed in the lecture as well as more complex ones like fmadd, representing fused-multiply-add. A fused-multiply-add operation performs a multiplication on the first both arguments and adds the third input vector, all within a single instruction.

The third part can also be declared as a **suffix** describing the type of the main input arguments. In this context, "p" stands for packed and is commonly used with "s" or "d", meaning single-precision or double-precision floating-point numbers. The letters "ep" are the short version for extended packed and is typically used in combination with "i" or "u", which is short for either a signed or unsigned integer. Note the following link explaining the purpose of the letter "e"; intel seems to have added it just for distinction. (`https://stackoverflow.com/questions/64600563/what-is-packed-and-unpacked-and-extended-packed-data`) Furthermore, the integer suffix includes a number, too. It expresses the size of the integer numbers (e.g. 8-bit, 16-bit, 32-bit, etc.). However, packed only says that the numbers are stored linearly without any padding.

2. What do the metrics latency and throughput tell you about the performance of an intrinsic function?

Latency shows the number of clock cycles that are needed to complete an operation. For example, a processor of the Broadwell architecture needs five cycles to complete a fused-multiply-add operation. **Throughput** describes how often it is possible to start an operation of the same kind. For example, the Broadwell processor can start two independent fused-multiply-add functions for each clock cycle. Thus, the throughput is 0.5.

3. How do modern processors realize instruction-level parallelism?

Instruction-level parallelism means that distinct operations can be performed by a single core at once. That is enabled by a scheduler which can access multiple ports. Each port provides different instruction units, such as arithmetic logic units (ALU), vector ALUs, floating-point units of various kinds, loading / storing units, etc. The instruction units of one port are independent of the instruction units of all the other ports. Due to this, the best performance will be achieved if the scheduler can assign work to all ports.

4. How may loop unrolling affect the execution time of compiled code?

Loop enrolling may affect the execution time on different stages. Thus, there is no ever valid answer. The goal of loop enrollment is to give compilers a hint for using vector instructions. As in the lecture presented, openMP directives are interpreted very differently. Some compilers nearly reach the same performance as the loop was enrolled with intrinsic procedures, whereas other compilers performing worse than if the for-loop were not enrolled at all. Furthermore, the execution time also depends on the processor architecture. While for a processor with avx512 support an enrolment of 16 could be the best, on a processor only supporting avx2 instructions, an enrolment of 8 might be faster.

5. What does a high IPC value (instructions per cycle) mean in terms of the performance of an algorithm?

A high IPC value indicates high instruction-level parallelism. Operating on huge arrays also allows concluding that the algorithm has a few cache losses. However, a high IPC score says nothing about the quality of an algorithm. Bubble sort was mentioned as an example in the lecture. Even if the algorithm is optimized to have a high IPC value, it would have a squared running time.