

# Algorithm Engineering Lab Assignment 6

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

January 15, 2021

## 1. Name some characteristics of the instructions sets: SSE, AVX(2) and AVX-512.

---

SSE is one of the first vector instruction sets. Its first version was released in 1999. Due to a maximal vector size of 128 bit, operations on vectors within four single-precision floating point scalars can be executed at once. Reading assembler code, SSE vectors can be identified by the register names xmm0 to xmm15.

AVX is available since 2011 and the extended version AVX2 since 2013. They enable the use of vector registers with a size of 256 bits. Thus AVX allows working with vectors containing up to eight floating-point scalars. AVX vectors can be identified by the register names ymm0 to ymm15 in assembler.

AVX-512 is the third version of AVX. Processors released after 2017 may support this instruction set. It provides vectors with a length of 512 bits. These vectors can be spotted in assembler code by the registers zmm0 to zmm31.

## 2. How can memory aliasing affect performance?

---

Memory aliasing occurs if two pointers are pointing to overlapping memory locations. When the compiler does not know whether memory aliasing is possible or not, it has to consider that it happens. This disables optimizations of the compiler such as vectorization and thus slows down the program.

## 3. What are the advantages of unit stride (stride-1) memory access compared to accessing memory with larger strides (for example, stride-8)?

---

Operating with unit stride access is described by working sequentially with each element of an array. It allows using the full memory bandwidth in contrast to accessing memory in larger strides. This is essential when an algorithm is memory intensive. Unit stridden memory access enables also vectorization, especially if an array is aligned. To execute vector operations with a stride-8 memory access, the vector has to be gathered first.

## 4. When would you prefer arranging records in memory as a Structure of Arrays?

---

Structures of Arrays are favourable when a algorithm operates more often on a specific variable of a structure and less often on the whole structures. For example if the goal is to determine the average for each single variable of a structure using the SoA approach should deliver results faster. This can be explained by having a look at the different memory usage. In the AoS