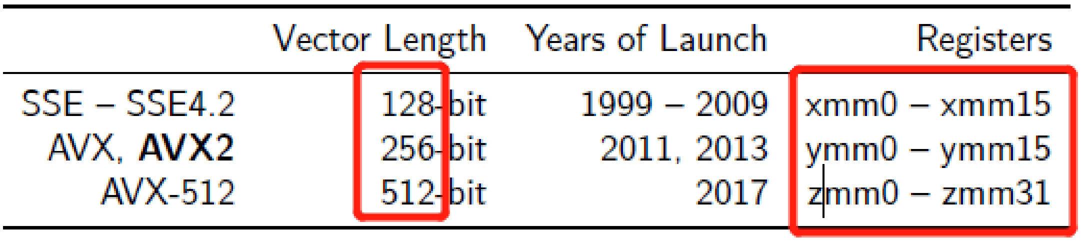
Exam Assignments V06

chengchengguo 183090

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

they are vector instruction sets on Intel CPUs, are extensions to the x86 instruction set architecture for microprocessors from Intel and Advanced Micro Devices (AMD).[1]
with those instructions ,we can in one instruction multiple element load, add and store



	Vector Length	Years of Launch	Registers
SSE – SSE4.2	128-bit	1999 – 2009	xmm0 – xmm15
AVX, **AVX2**	256-bit	2011, 2013	ymm0 – ymm15
AVX-512	512-bit	2017	zmm0 – zmm31

a float is 32 bit, so a SSE vector can save/operate 4 float

AVX2 8 float, AVX-512 16 float(or 8 double)

2. How can **memory aliasing** affect **performance**?

The case where two **pointers may point** to the **same memory location** is known as **memory aliasing**. In performing only **safe optimizations**, the compiler must assume that different pointers may be aliased.[2]

the compiler don't know the overlapped region, cannot good optimize
the overlap not only for pointer, but also for reference.

a pointer is a value that represents a memory address, aliasing is sometimes 2 pointers represent the same memory address

Because could be an alias for an element of the array it could change on any given iteration. Therefore the code has to load the value every single iteration, resulting in a potentially large slowdown.[3]

[1] Advanced Vector Extensions - Wikipedia
[2] folie
[3] c++ - What is aliasing and how does it affect performance? - Stack Overflow

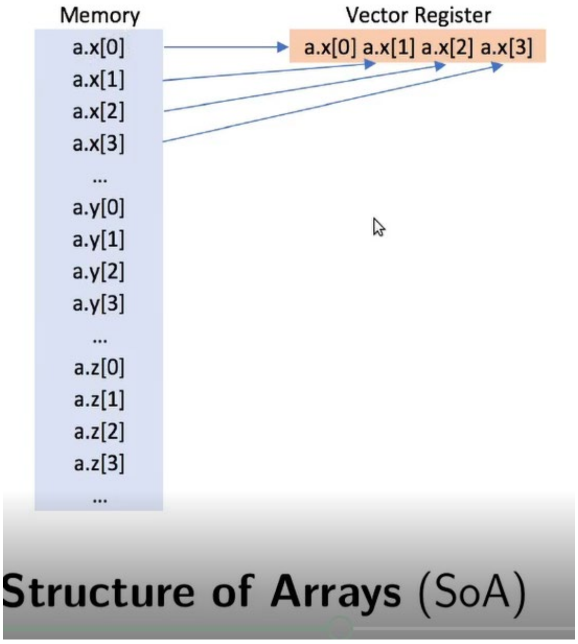
When the compiler knows that the referenced memory locations are unique, a number of optimizations can be enabled including vectorization.

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

we try to make best use of bandwidth, so compared with sreide-8 , stride-1 higher bandwidth utilization, easier for the compiler to vectorize code

and the gather befehl will be expensive, because stride-8 doesn't gather data continuously

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



Structure of Arrays (SoA)

it is good when the goal only want to get x -coordination, then all the element of x-coordination in one array/vector, it is good to vectorize

Keeping **separate arrays** for each structure-field keeps **memory accesses contiguous** when vectorization is performed over structure instances. For example, you want to compute **min**, **max**, **average**, **sum**, . . . of a structure-field over all instances (like in column store databases). In that case, irregular memory operations are avoided, which can **reduce latency**, and, **increase bandwidth usage**, or, **help** the **compiler** to **vectorize** the code automatically.