

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

Befehlssatz	Eigenschaften	Details
SSE (Streaming SIMD Extensions)	128-Bit-XMM-Register	Operationen: Ganzzahlen: Addition, Subtraktion, Multiplikation, logische Operationen (AND, OR, XOR, NOT) Fehlt: Gather/Scatter, Ganzzahldivision, Bitmanipulation, Maskenoperationen Floats (Single-Precision, 32-Bit): Addition, Subtraktion, Multiplikation, Division, Quadratwurzelberechnungen <i>Ab SSE2: Double-Precision-Floats, 64-Bit</i>
	Parallelität:	Acht 16-Bit-Ganzzahlen Vier 32-Bit-Ganzzahlen Zwei 64-Bit-Ganzzahlen speichern, aber eingeschränkt rechnen (über 32-Bit-Operationen zerlegt) Vier 32-Bit-Floats
	Geeignet für:	Grafik- und Multimediaanwendungen
AVX (Advanced Vector Extensions)	256-Bit-YMM-Register	Operationen: Ganzzahlen: wie SSE + Gather/Scatter, Maskenoperationen Floats: wie SSE + Double-Precision-Floats (64-Bit) VEX-Coding: Drei Operanden pro Befehl möglich
	Parallelität:	Sechzehn 16-Bit-Ganzzahlen Acht 32-Bit-Ganzzahlen Vier 64-Bit-Ganzzahlen Acht 32-Bit-Floats Vier 64-Bit-Floats
	Geeignet für:	wie SSE + High-Performance-Computing Maschinelles Lernen Datenverarbeitung (Big Data) Grafik- und Multimediaanwendungen
AVX-512	512-Bit-ZMM-Register	Operationen: Ganzzahlen: wie AVX + erweiterte Gather/Scatter, Maskenoperationen Floats: wie AVX
	Parallelität:	32 16-Bit-Ganzzahlen 16 32-Bit-Ganzzahlen 8 64-Bit-Ganzzahlen 16 32-Bit-Floats 8 64-Bit-Floats
	Zusätzliche Features:	Maskenregister (k0-k7) für selektive Operationen, gut für wissenschaftliche Berechnungen und KI-Anwendungen

2. "How can memory aliasing affect performance?"

Zur Erinnerung: "Memory Aliasing" = zwei (oder mehr) Zeiger im Code zeigen auf denselben Speicherbereich

- Wenn Daten, die laut Code unabhängig voneinander sind, in denselben Cacheline liegen, wird jede Änderung an einer der Adressen auch bei der anderen unnötige Invalidierenden auslösen (**False Sharing**).
- Damit Compiler Vektorisierung durchführen kann, muss er sicher sein, dass es keine Datenabhängigkeiten zwischen Speicherzugriffen gibt, sonst kann paralleles Schreiben vorherige Zugriffe überschreiben.
- Wenn der Compiler nicht sicher ist, ob Aliasing vorliegt (z. B. wenn mehr Zeiger in der `for`-Schleife Code verwendet werden), geht er vom Worst-Case Szenario aus und verzichtet auf die Vektorisierung und führt die Berechnungen seriell aus - **langsame Performance**.

Anmerkung: Mit `restrict` könnte man dem Compiler explizit sagen, dass Zeiger auf nicht überlappende Speicherbereiche zeigen.

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

Zur Erinnerung: Stride" = Abstand zwischen aufeinanderfolgenden Datenzugriffen im Speicher

Vorteile von Stride-1:

1. Bei **stride-1** liegen die benötigten Daten direkt nebeneinander. Und bei **stride-8** überspringt jeder Zugriff 7 Datenpunkte und jede Cacheline wird nur teilweise benutzt, das bedeutet, der Prozessor muss häufiger auf den Hauptspeicher zugreifen - also Laden von Cache-Lines ist mit **stride-1** effizienter.
2. Bei **stride-1** können Vektorregister schnell und effizient gefüllt und verarbeitet werden. Bei **stride-8** müssten die Daten erst aus nicht zusammenhängenden Speicherbereichen geladen und dann neu angeordnet werden.
3. Aus den oben genannten Gründen - **geringerer Energieverbrauch bei stride-1**.

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

Zur Erinnerung: Array of Structures (AoS): Attribute eines Objekts liegen zusammen. Structure of Arrays (SoA): Alle Attribute eines Typs liegen in eigenen Arrays.

Wann SoA?

- Bei Operationen, wo alle x-Koordinaten auf einmal verarbeitet werden - gut im Bezug auf Vektorisierung
- Wenn man nur manche Felder einer Struktur braucht - spart Bandbreite
- Bei der spaltenbasierten Verarbeitung, wie in Datenbanken