

Blatt 4

Aufgabe 1 (SIMD; Votieraufgabe)

Schreiben Sie eine Implementierung des Skalarprodukts, die SIMD verwendet und auf der fused multiply-add (FMA / FMADD) Operation beruht.

Bonusaufgabe 1: Schreiben Sie Ihren Code so, dass er je nach CPU-Architektur die Instruction Sets SSE, AVX und Neon nutzen kann.

Aufgabe 2 (Bekannte Tricks; Votieraufgabe)

Verwenden Sie die bereits bekannten Tricks um ihre Implementierung von Aufgabe 1 in Bezug auf Geschwindigkeit und Präzision mit SIMD und Präprozessoranweisungen zu verbessern.

Z.B.: `#define ADD_DOUBLE(X,Y) _mm512_add_pd(X,Y);`

Aufgabe 3 (Makefile; Votieraufgabe)

Schreiben Sie ein Makefile, das alle Compile Flags in einer Variable `CFLAGS` speichert und

1. eine C-Datei zunächst in Assembler-Code umwandelt,
2. den Assembler-Code in ein Objekt umwandelt
3. und das Objekt schließlich zu einer Binary linkt.

In jedem Schritt soll mittels `echo` ausgegeben werden, was gerade ausgeführt wird (Output bspw. "Assembler" für Schritt 1).

Orientieren Sie sich dabei an den Compile Flags aus der Vorlesung und informieren Sie sich, falls notwendig, über die Funktionsweise von Makefiles (<https://makefiletutorial.com/>).