LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

Grundlagenpraktikum: Rechnerarchitektur

Gruppe 213 – Abgabe zu Aufgabe A326
Sommersemester 2023

Noah Schlenker Leon Baptist Kniffki Christian Krinitsin

1 Einleitung

Verwendung von Wurzel 2:

- Das verhältnis der beiden seitenlängen eines blattes im din-a-format beträgt 1 / sqrt(2) mit rundung auf ganze millimeter. Dadurch ist sichergestellt, dass bei halbierung des blattes entlang der längeren seite wieder ein blatt im din-a-format (mit um eins erhöhter nummerierung) entsteht.
- Die wurzel aus 2 ist das frequenzverhältnis zweier töne in der musik bei gleichschwebender stimmung, die einen tritonus, also eine halbe oktave bilden.
- In der elektrotechnik enthält die beziehung zwischen scheitelwert und effektivwert von sinusförmiger wechselspannung ebenfalls die konstante sqrt(2).

Geschichte über Näherung der Wurzel:

- Die alten Inder schätzen $\sqrt{2} \approx \frac{577}{408} = 1,414215686\ldots$. Stimmt auf 5 Nachkommastellen. Abweichung beträgt nur +0,0001502 Prozent.
- Babylonier aus 1800 v. Chr.: $\frac{30547}{21600} = 1,414212962\ldots$. Abweichung von -0,0000424 Prozent.

Möglichkeit, $\sqrt{2}$ mit einer unendlichen Präzision zu berechnen. Vorwegnahme: Bignums, Newton-Raphson-Division, Karazuba, Matrixexponentiation.
(0,75 Seiten)

2 Lösungsansatz

2.1 Big-Num

Diskussion über die Notwendigkeit von Bignums, Erklärung der Implementierung (Little Endian, Arithmetische Operationen, nicht Division). Laufzeiten thematisieren?
(1 Seite)

2.2 Karazuba-Multiplikation

Erklärung des Algorithmus und der Umsetzung im Code.
(0,75 Seiten)

2.3 Newton-Raphson-Division

Erklärung des Algorithmus und der Umsetzung im Code.
(1 Seite)

3 Korrektheit/Genauigkeit

Wahrscheinlich Genauigkeit, da es die Aufgabe ist, $\sqrt{2}$ beliebig genau darzustellen.

Umfangreiche Erklärung darüber, wie die Matrix Elemente an $\sqrt{2}$ konvergiert und Newton-Raphson and die Division. Erklärung, wie die Kombination aus Bignum und Fixkommazahlen unendliche Genauigkeit ermöglicht, auf Kosten von Laufzeit, die im nächsten Kapitel beleuchtet wird.
(1,5 - 2 Seiten?)

4 Performanzanalyse

Newton Raphson Laufzeit erklären, mit Graphiken demonstrieren, das selbe mit Exponentiation.
Vergleichsimplementierungen ansetzen (SIMD, nicht SIMD? / Karazuba, normale Multiplikation / Bitshifts?), Graphisch laufzeiten vergleichen, tatsächliche Performanz erklären und schlussfolgern.
(2 - 3 Seiten)

5 Zusammenfassung und Ausblick

Ziel erreicht, unendliche Präzision ist gegeben. Nutzer können, abhängig von ihren Anforderungen oder "Computerspezifikationen", die Wurzel von 2 mit diesem Programm berechnen.
Ausblick: SIMD in AVX, mithilfe von 256 Bit kann man Multiplikationen noch schneller machen. Division lässt sich wahrscheinlich nicht optimieren, da SIMD nicht verwendet werden kann, und immer eine gewisse Anzahl an Iterationen gebraucht wird.
(0,75 - 1 Seite)
(Insgesamt 6 - 7,75 Seiten, 2 - 4 Seiten fehlen!)