

Arithmetik

Das Distributivgesetz sagt, dass $a(b + c) = ab + ac$. Gilt dies auch bei beschränkter

Rechengenauigkeit? Wie sollte man z.B. $m = \frac{1}{2}(a + b)$ (Mittelwertbildung) berechnen? Wann

ist $m = \frac{a}{2} + \frac{b}{2}$ günstig? Wann verwendet man lieber $m = \frac{1}{2}(a + b)$?

Antwort:

Arithmetik

Compilieren Sie den Ausdruck $y = 2(a - b)(a + b)$! Verwenden Sie die Register %eax, %ebx, %ecx usw. einer 2-Adress-Maschine für notwendige temporäre Variablen! Versuchen Sie, möglichst wenige Register zu verwenden!

Antwort:

Arithmetik

Die meisten Programmiersprachen erlauben es dem Compiler, alle Rechengesetze (Kommutativgesetz, Distributivgesetz, ...) zu nutzen. Versuchen Sie, den Ausdruck

$y = (2x - x) + \left(\frac{2n}{n} - 1\right)x$ mit einem möglichst einfachen Assembler-Programm schnell zu berechnen.

Antwort:

Arithmetik

Überlegen Sie sich, wie sich die Komplexität eines Assembler-Ausdruckes im Vergleich zu einem Hochsprachenausdruck verändert, wenn die Anzahl der Register nicht für alle temporären Zwischenergebnisse ausreichen.

Antwort:

Arithmetik

Berechnen Sie den Ausdruck $z = 17x + 15y$ mit einem einfachen Assembler-Programm ohne Multiplikationsbefehle!

Antwort:

Arithmetik

Programmieren Sie ein Programmfragment in Assembler-Code (i80x86, m68x05, o. ä.), das einen Links-Shift-Befehl mit Hilfe anderer arithmetischer Befehle emuliert. Es sollen keine Shift-, Multiplikations- oder Divisionsbefehle verwendet werden. (Ehemalige Klausuraufgabe.)

Antwort:

Arithmetik

Schreiben Sie eine (schnelle) C-, C++-, Pascal- oder Java-Funktion, die zwei (Unsigned-Short) Integer-Zahlen multipliziert, jedoch keinen ''-Operator verwendet (Shift-Operationen bzw. Multiplikation mit 2 sind erlaubt)!*

Arithmetik

Schreiben Sie eine (schnelle) C-, C++-, Pascal- oder Java-Funktion, die zwei (Unsigned-Short) Integer-Zahlen dividiert, jedoch keinen '/'-Operator verwendet (Shift-Operationen bzw. Division durch 2 oder Multiplikation mit 2 sind erlaubt)!

Arithmetik

Schreiben Sie ein Assembler-Programm, das eine im Register %eax übergebene Zahl auf die nächst-kleinere, durch 4096 teilbare Zahl abrundet (Beispiel: 4097 => 4096, 20825 => 20480).

Verwenden Sie hierzu keine Division/Multiplikation! (Hinweis: $4096 = 2^{12}$.)

Schreiben Sie ein Assembler-Programm, das eine im Register %eax übergebene Zahl auf die nächst-größere, durch 4 teilbare Zahl aufrundet (Beispiel: 2 => 4, 15 => 16, 8 => 12).

Verwenden Sie auch hier keine Division / Multiplikation!

Schreiben Sie eine sehr(!) schnelle Hash-Funktion, die Zahlen, die über den Bereich 0-4095 gleichmäßig verteilt sind (in %eax übergeben), gleichmäßig auf die Zahlen 0-255 abbildet!

Arithmetik

Compilieren Sie per Hand das folgende Programmfragment! Was fällt dabei auf? Was lässt sich einfach optimieren?

```
a = b + c;  
d = a + b;
```

Antwort:

Arithmetik

Schreiben Sie eine (einfache) C-, C++, Pascal- oder Java-Funktion, die $\exp(x)$ berechnet!

Arithmetik

Was müsste alles programmiert werden, um auf einem einfachen Mikro-Controller ohne Befehle für Integer-Multiplikation und -Division sowie ohne Gleitpunktoperationen die Funktion e^x berechnen zu können? Schreiben Sie keine Programme, sondern skizzieren Sie nur grob, welche Operationen Sie durch welche einfacheren Konstrukte nachbilden können! (Ehemalige Klausuraufgabe.)

Arithmetik

Antwort: