

Zentralübung Rechnerstrukturen: Vektorrechner 8. Übungsblatt

Besprechung: 21. Juli 2016

1 Vektorrechner

- a) Implementieren Sie mit den Vektorbefehlen aus dem Foliensatz 9 der Vorlesung, Folien 11 und 12, die DAXPY-Operation: $Y = a \cdot X + Y$, wobei X und Y Vektoren sind und a ein Skalar. Welches Programmierkonstrukt wird durch diese Vektoroperationen ersetzt?
- b) Gruppieren Sie voneinander unabhängige Vektorbefehle der DAXPY-Berechnung in sogenannte *Convoys* und stellen Sie eine Ausführungsreihenfolge dieser Gruppen auf. Gehen Sie davon aus, dass jede Vektorfunktionseinheit nur einmal existiert. Die Vektorinstruktionen der jeweiligen Gruppe werden parallel zur Ausführung gebracht und benötigen zur Ausführung jeweils ein sogenanntes *chime*. Wie viele *chimes* pro FLOP werden insgesamt benötigt?
- c) Die Ausführungszeit einer Folge von Vektoroperationen hängt auch von der Zeit für das Aufsetzen der Operationen ab. Dieser Overhead ist wie folgt gegeben:

Einheit	Start-up Overhead
Load and Store Unit	12 Zyklen
Multiply Unit	7 Zyklen
Add Unit	6 Zyklen

Geben Sie für jeden der Convoys aus Aufgabe 1b) und einer Vektorlänge n die folgenden Zeitpunkte an:

- den Startzeitpunkt,
- den Zeitpunkt an dem das erste Ergebnis des jeweiligen Convoys geliefert wird,
- den Zeitpunkt des letzten Ergebnisses.

Wie verhält sich für $n = 64$ diese Betrachtung zur Abschätzung mittels *chimes* aus Aufgabe 1b)?

- d) Vektorbefehle werden oft durch ein spezielles Speichersystem mit Verschränkung (memory interleaving) und mehreren Speicherbänken unterstützt. Wie lange dauert ein Ladebefehl eines 64-elementigen Vektors bei 16 Speicherbänken und einer Latenz von 12 Zyklen
- mit einem Stride von 1?
 - bei einem Stride von 32?
- e) Um die Abarbeitung der Vektorbefehle zu beschleunigen, haben Sie in der Vorlesung die Verkettung (engl. chaining) von Vektoroperationen kennengelernt. Vergleichen Sie die Ausführung mit und ohne Verkettung der folgenden Instruktionssequenz miteinander:

```
MULTV V1, V2, V3  
ADDV  V4, V1, V5
```

Die Vektoren haben 64 Elemente und die Verzögerung der Additionseinheit - und der Multiplikationseinheit sind 6 und 7 Zyklen. Wie groß ist der erzielte Speedup?

- f) Gegeben sei folgendes Fragment eines C-Programmes:

```
int i;  
int a[n], b[n];  
  
for (i = 0; i < n; i++)  
{  
    if (a[i] < b[i])  
    {  
        b[i] = i;  
    }  
}
```

Realisieren Sie dieses Code-Fragment mittels Vektorbefehlen. Gehen Sie bei Ihren Überlegungen davon aus, dass ein Vektorregister je alle n Werte der Arrays a oder b aufnehmen kann. Die Vektorbefehle dafür finden sie wiederum im Foliensatz 8 der Vorlesung auf den Folien 28 und 29.

2 Klausuraufgaben

In der Übung werden sofern die Zeit ausreicht, zusätzlich Klausuraufgaben aus den verschiedenen Bereichen der Übung berechnet und Fragen zur Klausur, sofern möglich, beantwortet. Aufgaben-/Themenwünsche bitte per Mail an den Übungsleiter. Die Klausuren finden Sie auf der Homepage zur Vorlesung Rechnerstrukturen.