

Bitte beachten Sie die allgemeinen Hinweise auf Übungszettel 1

Aufgabe 1: Fließbandverarbeitung

a) Einfache Pipeline

Gehen Sie im Folgenden von einer einfachen 5-stufigen Pipeline aus: Befehl holen (IF), Befehl dekodieren (ID), Operanden holen (OF), Ausführung (EX), Rückspeichern (WB). Weiterhin liegt eine reine Load/Store-Architektur ohne architekturelle Beschleunigungsmaßnahmen (z.B. forwarding, reordering etc.) oder Hardware zur Erkennung von Hemmnissen vor. Operanden können erst dann aus Registern geholt werden, nachdem sie zurück gespeichert wurden.

Sorgen Sie dafür dass die folgende Befehlsfolge aus Pseudoinstructions konfliktfrei ausgeführt wird:

```
ADD r0, r1, r2
SUB r1, r5, r0
MOV [rsp+8], r5
OR r0, r5, r4
MOV r3, [rsp+24]
AND r1, r0, r3
ADD r0, r1, r3
ADD r0, r0, 0x341D
ADD r0, r0, 0x52F6
```

Hinweis: ADD a, b, c entspricht $a=b+c$

b) Verbesserte Pipeline

Gehen Sie jetzt davon aus, dass die Pipeline aus dem vorherigen Aufgabenteil die in der Vorlesung kennengelernten Forwarding-Varianten/Shortcuts verwendet. Sorgen Sie wieder dafür, dass die Befehlsfolge aus dem vorherigen Aufgabenteil konfliktfrei ausgeführt wird.

Aufgabe 2: Bubblesort

Implementieren Sie folgende Funktion:

```
void sort(uint64_t len, int64_t a[len]);
```

Die Funktion soll den Inhalt des Arrays *a* nach folgendem Pseudocode sortieren:

```
for(i=size(A); i>1; i--) {
    for(j=0; j<i-1; j++) {
        if(A[j] > A[j+1]) {
            // Swap entry j and j+1
            tmp = A[j];
            A[j] = A[j+1];
            A[j+1] = tmp;
        }
    }
}
```

Im KVV wird Ihnen ein C-Wrapper gestellt, mit dem Sie Ihre Funktion testen können.