

Übungsblatt 4

(10 Punkte)

Besprechung ab Montag, 4. Mai 2015

4.1 Befehlssatzarchitekturen (2 Punkte)

Charakterisieren Sie in eigenen Worten (kurz) alle folgenden Befehlssatz-Architekturen:

- RISC
- CISC
- DSP
- Netzwerk-Prozessoren
- VLIW
- EPIC
- Multimedia-Erweiterungen
- Abstrakte Maschinen

4.2 Booth-Algorithmus (4 Punkte)

Seien $a = -5$ und $b = -7$ zwei Zahlen in 8-Bit-Zweierkomplementdarstellung.

- a) Berechnen Sie das Produkt $a * b$ mit Hilfe des Booth-Algorithmus.
- b) In der Vorlesung wurde eine Maßnahme angedeutet, mit der der Booth-Algorithmus effizienter formuliert werden kann. Berechnen Sie das Produkt mit diesem verbesserten Algorithmus.

4.3 Itanium-Architektur (4 Punkte)

Arithmetische Gleitkomma-Befehle benötigen i.d.R. mehr als einen Zyklus. Auf einer Itanium-Architektur wurden folgende Operationslatenzzeiten gemessen:

Ausgangsoperation	Folgeoperation	Latenzzeit in Zyklen
GK-ALU-Operation	weitere GK-ALU-Operation	3
GK-ALU-Operation	Speichern <i>doubleword</i>	2
Laden <i>doubleword</i>	GK-ALU-Operation	1
Laden <i>doubleword</i>	Speichern <i>doubleword</i>	0

Die erste Spalte gibt den Ausgangsoperationstyp, der das Ergebnis erzeugt, an. In der zweiten Spalte befinden sich die Typen der datenabhängigen Folgebefehle, die erst nach einer gewissen Anzahl von Taktzyklen (3. Spalte) ausgeführt werden sollten, um Wartezyklen zu vermeiden. Wenn bspw. eine GK-ALU-Operation im Zyklus n ausgeführt wird, darf auf das Ergebnis erst im Zyklus $n+3$ mit einem Speicherbefehl zugegriffen werden.

Gegeben sei die folgende Schleife, die Gleitkomma-Elemente (64Bit) eines Arrays inkrementiert:

```
for ( i=1000; i>=0; i-- )
    x[i] = x[i] + s;
```

- Stellen Sie die Schleife in Assembler dar. Gehen Sie davon aus, dass Register R1 und R2 die Adresse des letzten bzw. ersten Arrays-Elements enthalten und dass Register FP2 den Wert von s beinhaltet.
- Rollen Sie die Schleife sieben Mal ab und plazieren Sie die Instruktionen in Bündel (*bundles*). Ignorieren Sie hierbei die Pipeline-Latenzzeiten, um die Anzahl der Bündel zu minimieren.
- Führen Sie ein Scheduling des Codes aus dem letzten Aufgabenteil durch, um möglichst viele Wartezyklen zu vermeiden.
- Wieviele Zyklen werden für das Ausführen der aufgerollten Schleife für die Codefragmente aus Aufgabenteil b) und c) benötigt, wenn davon ausgegangen werden kann, dass für die Ausführung der Bündel die Itanium-Restriktionen gelten und jeder *Stall* das gesamte Bündel anhält.

Hinweise:

- Der Einfachheit halber sollten MIPS-Instruktionen, wie LDC1, inklusive der Gleitkomma-Register FPn verwendet werden.
- Beschränken Sie sich im Aufgabenteil b) und c) auf folgende Kombinationen der execution union slots: M-M-I, M-M-F, M-M-B, M-F-I, M-I-I und M-I-B (siehe Folien 16-17 rsys-2-5).

Allgemeine Hinweise: Die Übungstermine und weitere Informationen finden Sie unter <http://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/sommersemester-2015/rechnerarchitektur.html>. Die Übungszettel werden zum Semesterbeginn online gestellt und sollen eigenständig bis zum jeweiligen Stichtag gelöst werden. Die Lösungen werden in den Gruppen besprochen. Auf Wunsch kann für diese Veranstaltung ein Übungsschein ausgestellt werden. Hierzu müssen die selbst erstellten Lösungen jeweils vor der Besprechung der Aufgaben beim Übungsgruppenleiter abgegeben werden. Dabei müssen 45% der Gesamtpunkte bei den Übungszetteln erreicht und eigene Lösungen in der Übungsgruppe präsentiert werden. Für die Teilnahme an der Klausur nach BPO 2013 / der Fachprüfung nach DPO 2001 ist der Übungsschein *nicht* erforderlich.