

Übungsblatt 12

(13 Punkte)

Besprechung am Montag, 14. Juli 2014

12.1 Multithreading (5 Punkte)

In dieser Aufgabe sollen Sie verschiedene Multithreading-Modelle exemplarisch anwenden. Gegeben sei dazu ein System mit

- einem Prozessor mit dynamischem Scheduling und
 - 2 Integer-Ausführungseinheiten (Latenz 1 Takt)
 - 1 Load/Store-Einheit (Latenz 1-4 Takte, je nach Cacheverhalten)
 - 1 Floating-Point-Einheit (Latenz 1 Takt)
- einem L1-Cache und einem L2-Cache. Die Latenzen der Zugriffe sind wie folgt:

L1-Zugriff	L2-Zugriff	Latenz (Takte)
Hit	-	1
Miss	Hit	2
Miss	Miss	4

Beide Caches unterstützen *Hit-Under-1-Miss*, d.h. während ein Miss bearbeitet wird, können weitere Hit-Zugriffe abgewickelt werden. Ein Zugriff, der einen weiteren Miss produzieren würde, wird abgebrochen und muß vom Prozessor wiederholt werden.

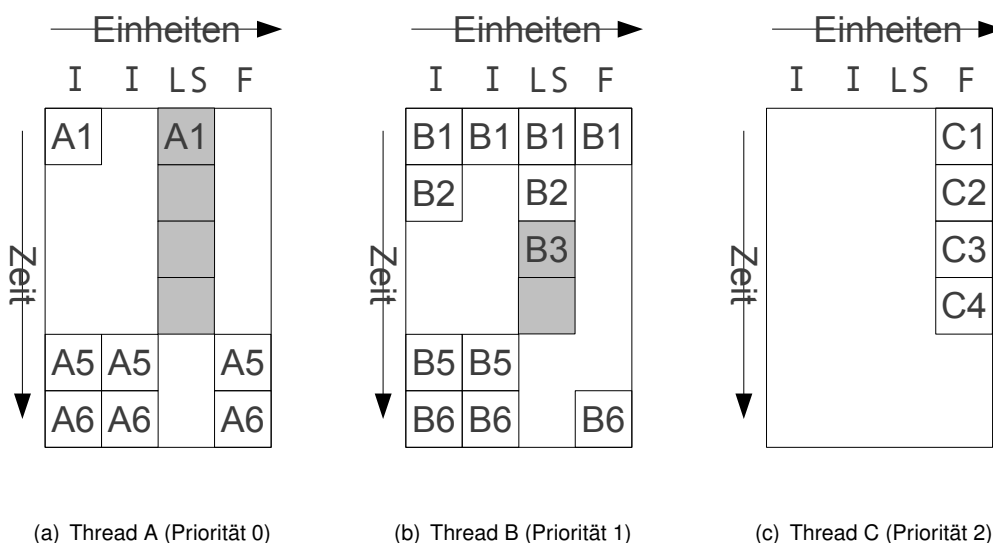


Abbildung 1: Ausführungsverlauf dreier Threads auf der gegebenen Plattform

Für dieses System sind die Ausführungsverläufe dreier Programme A-C in Abbildung 1 skizziert, wie sie sich bei isolierter Ausführung auf dem System ergeben. Operationen die mehr als einen Takt benötigen sind in hellgrau markiert.

Außerdem ist die Priorität jedes Threads angegeben (nur bei CGMT und SMT nötig), wobei niedrigere Zahlen höherer Priorität entsprechen.

- a) Zeichnen Sie für eine Ausführung mit *Coarse-Grained Multithreading (CGMT)*, *Fine-Grained Multithreading (FGMT)* und *Simultaneous Multithreading (SMT)* jeweils den zeitlichen Verlauf der Abarbeitung auf dem gegebenen System. Richten Sie sich bei der Notation nach Abbildung 1, d.h. tragen Sie die Verarbeitungseinheiten horizontal und die Zeit vertikal auf. Der *Context-Switch-Overhead* sei 1 Takt für CGMT und 0 Takte für die anderen Modelle.

Sie können davon ausgehen, daß die Speicherzugriffe der Threads auf disjunkte Cache-Bereiche abgebildet werden, d.h. dass diese Zugriffe interferenzfrei sind. Außerdem sei der verwendete Scheduler ein ASAP-Scheduler (*As Soon As Possible*). Falls nicht alle Instruktionen aus einem Zeitschritt aus den Abbildungen 1(a) bis 1(c) gleichzeitig gestartet werden können führt der Scheduler die verbleibenden so bald wie möglich danach aus. Nehmen Sie in diesem Fall außerdem an, dass erst *alle* Instruktionen eines Zeitschritts i aus Abbildung 1 ausgeführt worden sein müssen, bevor Instruktionen aus Schritt $i + 1$ desselben Threads ausgeführt werden können.

- b) Was sind die hardwareseitigen Voraussetzungen für die einzelnen Modelle?

12.2 Grobgranulares Multithreading (2 Punkte)

Für die Ausführungszeiten der Instruktionen aller Threads einer Anwendung gilt diese Verteilung:

- 40% der Taktzyklen entfallen auf die Ausführung der Instruktionen im Prozessor.
- 30% der Taktzyklen sind Warte-Zyklen, die durch L1-Cache-Misses (aber L2-Cache-Hits) entstehen. Die L1-Miss-Bearbeitung benötigt 10 Taktzyklen.
- 30% der Taktzyklen sind Warte-Zyklen, die durch L2-Cache-Misses hervorgerufen werden (jeweils 30 Taktzyklen).

Im Durchschnitt tritt pro Thread nach jeweils 10 Rechen-Takten ein Cache-Miss auf. Beantworten Sie folgende Fragen unter der Annahme, dass ausreichend viele Threads sowohl von der Hardware als auch durch die Anwendung unterstützt werden.

- a) Wie hoch ist die maximale Auslastung, wenn ein Threadwechsel 5 Taktzyklen dauert?
- b) Wie lange darf ein Threadwechsel höchstens dauern, so dass die Auslastung mindestens 50% beträgt?

12.3 Routing in Netzwerktopologien (6 Punkte)

Die folgenden Teilaufgaben beschäftigen sich mit Routing von Paketen/Flits in verschiedenen, aus der Vorlesung bekannten Netztopologien. Wir gehen hier vom Routing innerhalb eines einzelnen Switches aus, die Konzepte sind aber auch auf das Routing innerhalb eines On-Chip-Netzes übertragbar. Die Nummerierung der Ports des Switches entspricht hierbei dem System aus der Vorlesung, d.h. die Ports werden an der westlichen und östlichen Seite des Netzes angelegt; auf jeder Seite erhält der nördlichste Port den Index 000, alle weiteren Ports werden in südlicher Richtung aufsteigend binär durchnummeriert.

- a) Zeichnen Sie ein Gitter-Netz, der Dimension 3. Führen Sie das North-last Routing für folgende Ports durch:
- von 000 nach 111
 - von 101 nach 101
- b) Zeichnen Sie ein Benes-Netz, der Dimension 3. Überlegen Sie sich einen Algorithmus (Pseudo-Code) mit dem Sie das Routing durchführen können und wenden Sie ihn für die in Teilaufgabe a) gegebenen Routen an.
- c) In den vorherigen Aufgabeteilen sollten Sie das Routing für ein ungestörtes Netz zeichnen. Fügen Sie nun jeweils eine beliebige Störung ein, die die bisherige Route verändert und führen Sie das Routing für die gestörten Netze erneut durch.

Allgemeine Hinweise: Die Übungstermine und weitere Informationen finden Sie unter <http://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/sommersemester-2014/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.