

# Übungsblatt 12

## Rechnerarchitektur im SoSe 2020

Zu den Modulen P,Q

**Besprechung:** Besprechung der Übungsaufgaben in den Übungsgruppen vom 20. – 24. Juli 2020

### Aufgabe Ü25: Blockersetzung

(– Pkt.)

In der folgenden Aufgabe sollen Sie einige Überlegungen über Blockersetzungsstrategien bei Caches anstellen:

- a. Für welche Cache-Typen sind Blockersetzungsstrategien notwendig?
- b. Ein Cache habe Platz für vier Einträge (0 bis 3). Beim Verdrängen werde die LRU-Strategie angewandt. Vorher werden allerdings die noch unbelegten Einträge in aufsteigender Reihenfolge gefüllt.

Geben Sie für jeden Schritt den Inhalt der vier Adreßteile des Cachespeichers an, wenn nacheinander Zugriffe auf folgende Adressen erfolgen:

1 3 7 8 4 7 1 3 4.

Kennzeichnen Sie Verdrängungen.

- c. Vergleichen Sie verschiedene Ersetzungsstrategien bezüglich Hardware-Aufwand und Fehlzugriffsraten.

## Aufgabe Ü26: Pipelining

(– Pkt.)

Bei der folgenden Frage wird von einer 5–stufigen Pipeline mit einer Ausführungszeit von 2ns pro Stufe ausgegangen. Folgende Ausführungszeiten sind gegeben:

Befehl	IF	ID	EX	MA	WB
load word (lw)	2 ns	1 ns	2 ns	2 ns	1 ns
store word (sw)	2 ns	1 ns	2 ns	2 ns	—
add	2 ns	1 ns	2 ns	—	1 ns
Branch (beq)	2 ns	1 ns	2 ns	—	—

Die Stufen haben folgende Bedeutung:

**IF (Instruction Fetch):** Laden des nächsten Befehls aus dem Speicher.

**ID (Instruction Decode and register file read):** Lesen der Register und gleichzeitiges Entschlüsseln des Befehls. (Das Format der MIPS-Befehle ermöglicht das gleichzeitige Lesen und Entschlüsseln.)

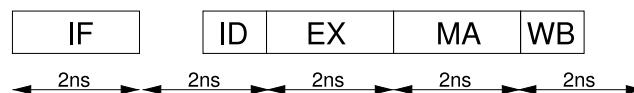
**EX (EXecution or adress calculation):** Ausführung der Operation oder Berechnung einer Adresse.

**MA (data Memory Access):** Zugriff auf einen Operanden im Datenspeicher.

**WB (Write Back):** Schreiben des Ergebnisses in ein Register.

Gehen Sie davon aus, dass das Schreiben eines Registersatzes (WB-Stufe) in der ersten Hälfte des entsprechenden Taktzyklus erfolgt und das Lesen aus dem Registersatz (ID-Stufe) in der zweiten Hälfte.

Die Pipeline sieht damit folgendermaßen aus:



Gegeben sei nun folgendes Programmfragment:

```
1 lw      $2, 100($5)
2 add     $3, $3, $4
3 add     $1, $4, $5
```

- Wie lang dauert die Ausführung dieses Programmfragments **ohne** Pipelining?
- Wie lang dauert die Ausführung dieses Programmfragments **mit** Pipelining?