

# ERA-Übungsblatt 10

1. In Aufgabe 1 werden folgende beiden Formeln verwendet. Sie sollten aus der ZU bekannt sein und sind deshalb auch klausurrelevant! (Letztes Jahr kam dazu eine Klausuraufgabe!)

$$\text{CPU Time} = \underbrace{IC}_{\text{Ausführungszeit}} \cdot \left( \underbrace{\frac{CPI}{f}}_{\text{CPU-Frequenz}} + \underbrace{\frac{\text{Memory Accesses}}{\text{Instruction}}}_{\text{durchschn. \# Speicherzugriff pro Instruction: Konstant!}} \cdot \underbrace{AMAT}_{\text{average memory access time}} \right)$$

$$AMAT = \text{Hit Latency} \cdot \text{Hit Rate} + \text{Miss Latency} \cdot \text{Miss Rate}$$

a) Umstellen auf AMAT:

$$AMAT = \frac{\frac{\text{CPU Time}}{IC} - \frac{CPI}{f}}{\frac{\text{Memory Accesses}}{\text{Instruction}}} = \frac{\frac{66s}{3 \cdot 10^3 \text{ ins}} - \frac{2 \frac{\text{cycles}}{\text{ins}}}{10^3 \frac{\text{cycles}}{s}}}{1,33 \frac{\text{accesses}}{\text{ins}}} \approx 15 \cdot 10^{-3} \frac{s}{\text{access}} = 15 \frac{ns}{\text{access}}$$

- b) Hit Latency: Dauer Zugriff falls Datum im Cache:  $t_{\text{CPU} \rightarrow \text{Cache}} + t_{\text{Cache Seek}} + t_{\text{Cache} \rightarrow \text{CPU}} = 10 \text{ ns}$   
Miss Penalty: Overhead, falls Datum nicht im Cache:  $t_{\text{Cache} \rightarrow \text{Mem}} + t_{\text{Mem Seek}} + t_{\text{Mem} \rightarrow \text{Cache}} = 60 \text{ ns}$   
Miss Latency: Dauer Zugriff, falls Datum nicht im Cache:  $\text{Hit Latency} + \text{Miss Penalty} = 70 \text{ ns}$   
Miss Rate: % an Zugriffen, die ein Miss sind.  $\text{Hit Rate} + \text{Miss Rate} = 1$

Umstellen von AMAT zu Miss Rate und Substitution  $\text{Hit Rate} = 1 - \text{Miss Rate}$

$$AMAT = \text{Hit Rate} \cdot \text{Hit Latency} + \text{Miss Rate} \cdot \text{Miss Latency}$$

$$\Rightarrow AMAT = (1 - MR) \cdot HL + MR \cdot ML$$

$$\Rightarrow AMAT = HL - MR \cdot HL + MR \cdot ML$$

$$\Rightarrow AMAT - HL = MR (-HL + ML)$$

$$\Rightarrow MR = \frac{AMAT - HL}{-HL + ML} = \frac{15 \text{ ns} - 10 \text{ ns}}{-10 \text{ ns} + 70 \text{ ns}} = \frac{5 \text{ ns}}{60 \text{ ns}} \approx 0,083 = 8,3\%$$

c, d, e) Siehe ML

2. a) 2-fach assoziativ  $\hat{=}$  2 Cachezeilen pro Cacheset. Bei 4 Cachezeilen also  $\# \text{ Cachezeilen} = 2 \cdot 4 = 8$

- b)  $\# \text{ Offsetbits} = \lceil \log_2 (\text{Cachezeilenlänge}) \rceil = \lceil \log_2 (32) \rceil = 5$   
 $\# \text{ Indexbits} = \lceil \log_2 (\# \text{ Cachesets}) \rceil = \lceil \log_2 (4) \rceil = 2$   
 $\# \text{ Tagbits} = \text{Adresslänge} - \# \text{ Offsetbits} - \# \text{ Indexbits} = 32 - 5 - 2 = 25$

$\lceil \dots \rceil$  bezeichnet die  
ceil-Funktion, d.h. zur  
nächsten Ganzzahl aufrunden

Eine Adresse wird also folgendermaßen aufgeteilt:



c) Wir untersuchen die Zugriffe, die bei dem gegebenen Stride von 32 Elementen erzeugt werden.

Iteration	Tag	Index	Offset
1	0 ... 0000	00	00000
2	0 ... 0001	00	00000
3	0 ... 0010	00	00000
4	0 ... 0011	00	00000
⋮	⋮	⋮	⋮

ein Schritt von 32 Elementen zu 4 Byte  
entspricht einer Erhöhung der Adresse um  
 $32 \cdot 4 = 128 \text{ Byte}$

$$128_{10} = 10000000_2$$

Aus der Tabelle sehen wir, dass die Indexbits gleich bleiben, d.h. alle Adressen bilden auf das selbe Cacheset ab, es kommt zu Verdrängungen.

$$\#H_1 = \left\lceil \frac{1000}{64} \right\rceil = 16 \quad H_2 = \left\lceil \frac{1000}{32} \right\rceil = 32$$

In der ersten Schleife schauen wir also 16 Elemente an, in Schleife 2 zusätzlich noch 16 weitere Elemente (insgesamt also 32).

i) Conflict Miss: Datum  $T_1$  wurde durch  $T_2$  verdrängt. Ein neuer Zugriff auf  $T_1$  ist ein Conflict Miss

Anzahl in Schleife 1: 0, wir sehen uns kein Element doppelt an.

Anzahl in Schleife 2: 16, die Elemente aus Schleife 1 werden verdrängt und erzeugen Conflict Misses

ii) Cold Miss: Tritt auf wenn das erste Mal auf ein Datum zugegriffen wird

Anzahl in Schleife 1: 16, das Cache ist anfangs leer und wir greifen auf jedes Element das erste Mal zu

Anzahl in Schleife 2: 16, auf die anderen 16 Elemente wurde schon in Schleife 1 zugegriffen.