

Lösungsvorschläge zur Klausur K01609 vom 20.3.2010

Aufgabe 1 (8 Punkte)

Lösung: Die Aussagen b, e, g und h sind richtig.

Aufgabe 2 (14 Punkte)

Lösung: Siehe Einsendeaufgabe 1.10

Aufgabe 3 (5 Punkte):

Lösung: Der dritte und vierte Zustandsgraph von oben sind korrekt. Im Kurstext wird eine Codierung benutzt, die sich an der Notation 1 für taken und 0 für not taken orientiert (dritter Zustandsgraph). Man beachte jedoch, dass die Zustands-codierung beliebig gewählt werden kann. Im vierten Zustandsgraphen sind die einzelnen Zustandsbit invertiert. Insgesamt gibt es $4!$, also 24, mögliche Codierungen.

Aufgabe 4 (15 Punkte)

a) Lösung: DM

b) Lösung:

| | Tag-Teil | Index-Teil | Wortadresse |
|----|----------|------------|-------------|
| DM | 27 | 3 | 2 |
| A2 | 28 | 2 | 2 |
| VA | 30 | - | 2 |

c) Lösung:

Im Folgenden werden die einzelnen Ergebnisse ausführlich hergeleitet.

c1) Zunächst ermitteln wir das Verhalten des DM-Cache:

| Hexadez. Adresse | Tag | Index | Wort | Hit/ Miss | Cache-Block |
|------------------|-----|-------|------|-----------|-------------|
| C6 | 110 | 001 | 10 | Miss | 001 |
| 89 | 100 | 010 | 01 | Miss | 010 |
| A7 | 101 | 001 | 11 | Miss | 001 |
| D3 | 110 | 100 | 11 | Miss | 100 |
| C2 | 110 | 000 | 10 | Miss | 000 |
| C4 | 110 | 001 | 00 | Miss | 001 |
| A3 | 101 | 000 | 11 | Miss | 000 |
| D0 | 110 | 100 | 00 | Hit | 100 |
| DD | 110 | 111 | 01 | Miss | 111 |
| C3 | 110 | 000 | 11 | Miss | 000 |

Die beiden Bits zur Wortauswahl sind nicht von Bedeutung. Der Index-Teil der Adresse ($A_4A_3A_2$) legt den Cache-Block fest. In der rechten Spalte tragen wir nach einem Miss die Blöcke ein, die in den Cache geladen werden. Bei jeder neuen Adresse wird geprüft, ob der zugehörige Cache-Block gültig ist. Ein Hit liegt vor, wenn die Tag-Bits des zuletzt geladenen passenden Cache-Blocks mit den entsprechenden Bits der aktuellen Adresse (A_7, A_6, A_5) übereinstimmen, sonst liegt ein Miss vor und der Cache-Block samt Tag-Bits wird geladen.

c2) Betrachten wir nun den A2-Cache:

| Hexadez. Adresse | Tag | Index | Wort | Hit/ Miss | Cache-Block |
|------------------|------|-------|------|-----------|-------------|
| C6 | 1100 | 01 | 10 | Miss | A-01 |
| 89 | 1000 | 10 | 01 | Miss | A-10 |
| A7 | 1010 | 01 | 11 | Miss | B-01 |
| D3 | 1101 | 00 | 11 | Miss | A-00 |
| C2 | 1100 | 00 | 10 | Miss | B-00 |
| C4 | 1100 | 01 | 00 | Hit | A-01 |
| A3 | 1010 | 00 | 11 | Miss | A-00 |
| D0 | 1101 | 00 | 00 | Miss | B-00 |
| DD | 1101 | 11 | 01 | Miss | A-11 |
| C3 | 1100 | 00 | 11 | Miss | A-00 |

Jeder Cache-Block wird über einen 2-Bit Index selektiert. Die beiden Sätze werden durch die Buchstaben A und B unterschieden. In beiden Sätzen wird gleichzeitig bei allen gültigen Cache-Blöcken geprüft, ob die Tag-Bits

($A_7 \dots A_4$) mit den zugehörigen Adressleitungen übereinstimmen. Ist dies der Fall, so liegt ein Treffer vor, sonst ein Miss und es wird in einen freien Cache-Block des Satzes geschrieben bzw. ein belegter nach LRU ersetzt.

c3) Schließlich kommen wir zum AV-Cache:

| Hexadez. Adresse | Tag | Wort | Hit/Miss | Cache-Block |
|------------------|--------|------|----------|-------------|
| C6 | 110001 | 10 | Miss | 000 |
| 89 | 100010 | 01 | Miss | 001 |
| A7 | 101001 | 11 | Miss | 010 |
| D3 | 110100 | 11 | Miss | 011 |
| C2 | 110000 | 10 | Miss | 100 |
| C4 | 110001 | 00 | Hit | 000 |
| A3 | 101000 | 11 | Miss | 101 |
| D0 | 110100 | 00 | Hit | 011 |
| DD | 110111 | 01 | Miss | 110 |
| C3 | 110000 | 11 | Hit | 100 |

Um zu prüfen, ob ein Hit vorliegt, wird der Tag-Teil der Adresse ($A_7 \dots A_2$) mit den Tags aller gültigen Cache-Blöcke verglichen. Es werden also 8 Vergleiche mit 6 Bit Wortbreite benötigt. Stimmt kein Tag mit dem Tag-Teil der anliegenden Adresse überein, so liegt ein Miss vor und der Cache-Block (Cache-Line) wird vom Hauptspeicher geladen. Für die gegebene Adressfolge musste kein gültiger Cache-Block verdrängt werden.

Aufgabe 5 (9 Punkte):

a) Lösung:

$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{f \cdot T(1) + \frac{1}{n} \cdot (1-f)T(1)} = \frac{1}{f + \frac{1-f}{n}}$$

b) Lösung:

$$4 = \frac{1}{f + \frac{1-f}{8}}$$

$$4 \cdot f + \frac{4}{8}(1-f) = 1$$

$$\frac{7}{2} \cdot f = \frac{1}{2}$$
$$f = \frac{1}{7}$$

c) Lösung:

$$S(32) = \frac{1}{\frac{1}{7} + \frac{6}{7 \cdot 32}} = \frac{7 \cdot 32}{38} = 5,89$$

Der Aufwand lohnt sich nicht, da die Effizienz gegenüber von $E(4) = 0,50$ auf $E(32) = 0,18$ abfällt. Selbst bei massiver Steigerung der Prozessoranzahl wird $S(n) < \frac{1}{f} = 7$ bleiben.