

Aufgabe 4: Cache-Musterlösung

Gegeben seien drei kleine Cachespeicher DM, A2 und AV, die jeweils 8 Cacheblöcke enthalten, wobei jeder Cacheblock vier Datenworte umfaßt. Cache DM ist als direct mapped Cache organisiert, Cache A2 als 2-fach assoziativer Cache und Cache AV ist voll assoziativ. Bei A2 und AV soll die LRU-Ersetzungsstrategie (LRU = least recently used = am längsten unbenutzt) angewendet werden. Der Speicher ist wort-adressiert. D.h. zu jeder Adresse gehört ein Wort. Nehmen Sie nun an, die Caches seien zu Beginn leer und es soll eine Serie von einzelnen Datenworten mit den folgenden 32-Bit-Adressen gelesen werden:

294928070, 294928009, 294928039, 294928083, 294928066, 294928068, 294928035, 294928080, 294928093, 294928067

Hinweis: 294928000d = 0001 0001 1001 0100 0011 1110 1000 0000b. Für die immer wieder vorkommende Zeichenfolge "29492800" bzw. "0001 0001 1001 0100 0011 1110" können sie das Zeichen "*" verwenden.

- a) Geben Sie zunächst für alle drei Cachespeicher die Aufteilung in Offset, Index und Tag an. *Da der Speicher wort-adressiert ist, gibt es keinen Byte-Offset.*
- DM: 2 Bit Block-Offset, 3 Bit Index, 27 Bit Tag
 A2: 2 Bit Block-Offset, 2 Bit Index, 28 Bit Tag
 AV: 2 Bit Block-Offset, 30 Bit Tag
- b) Geben sie in der untenstehenden Tabelle an welche Adress-Bereiche bei den oben beschriebenen einzelnen Lese-Zugriffen in welche Cache-Blöcke kopiert werden. Bezeichnen sie die Cache-Blöcke folgendermassen: "Satznummer-Blocknummer". Bsp: 5-010 kennzeichnet den dritten Block im fünften Satz eines hypothetischen teilassoziativen Caches. Kennzeichnen sie Hits durch ein "H". Verwenden sie folgende Annahme: Wenn mehrere freie Cache-Blöcke zur Verwendung zur Auswahl stehen, wird stets der Block mit der niedrigsten Bezeichnung (= niedrigste Satznummer) gewählt.

Dezimale Adresse	Binäre Adresse	(Dezimaler) Adress-Bereich	DM-Block	A2-Block	AV-Block
*70	* 1100 0110	*68 - *71	001	1-01	1
*09	* 1000 1001	*08 - *11	010	1-10	2
*39	* 1010 0111	*36 - *39	001	2-01	3
*83	* 1101 0011	*80 - *83	100	1-00	4
*66	* 1100 0010	*64 - *67	000	2-00	5
*68	* 1100 0100	*68 - *71	001	H	H
*35	* 1010 0011	*32 - *35	000	1-00	6
*80	* 1101 0000	*80 - *83	H	2-00	H
*93	* 1101 1101	*92 - *95	111	1-11	7
*67	* 1100 0011	*64 - *67	000	1-00	H

- c) Welche Vor- und Nachteile hat die Erhöhung der Assoziativität in Caches während gleichzeitig die Speicherkapazität unverändert bleibt? (xxx Punkte)
- Vorteile: *Selteneres Überschreiben der Daten im Cache, weniger Cacheflattern, dadurch höhere Hit-Rate.*
- Nachteile: *Mehr Hardwareaufwand durch mehr parallele Tag-Vergleiche.*
- d) Welche zwei in der Vorlesung besprochenen Vorgehensweisen gibt es, mit durch Schreibzugriffe veränderten Cache-Inhalten umzugehen? Beschreiben sie diese kurz und gehen sie dabei auch auf Vor- und Nachteile ein. (xxx Punkte)
- Write Through: *Direkt beim Verändern auch in den Hauptspeicher schreiben,*

dadurch langsamer (falls kein Puffer verwendet wird).

Copy Back: Zurückschreiben beim Ersetzen im Cache (angezeigt durch Dirty-Bit).

Bei Write-Through wird nur das veränderte Datum in den Hauptspeicher geschrieben, bei Copy-Back der ganze Block, was je nach Speicher langsamer sein kann. Dafür wird bei mehrfachen Änderungen des gleichen Datums bei Write-Through jedesmal auch der Hauptspeicher verändert, bei Copy-Back nur einmal beim Ersetzen.