



UNIVERSITÄT KARLSRUHE  
Fakultät für Informatik  
Industrielle Anwendungen der Informatik und Mikrosystemtechnik (IAIM)  
Prof. Dr. R. Dillmann

## Aufgabenblätter

zur Klausur „Technische Informatik I/II“

am 03. März 2004, 14.00 - 16.00 Uhr

- Beschriften Sie bitte gleich zu Beginn jedes Lösungsblatt deutlich lesbar mit Ihrem Namen und Ihrer Matrikelnummer.
- Diese Aufgabenblätter werden nicht abgegeben. Tragen Sie Ihre Lösung deshalb ausschließlich in die für jede Aufgabe vorgesehenen Bereiche der Lösungsblätter ein. Lösungen auf separat abgegebenen Blättern werden nicht gewertet.
- Außer Schreibmaterial sind während der Klausur keine Hilfsmittel zugelassen. Täuschungsversuche durch Verwendung unzulässiger Hilfsmittel führen unmittelbar zum Ausschluss von der Klausur und zur Note „nicht bestanden“.
- Soweit in der Aufgabenstellung nichts anderes angegeben ist, tragen Sie in die Lösungsblätter bitte nur die Endergebnisse ein. Die Rückseiten der Aufgabenblätter können Sie als Konzeptpapier verwenden. Weiteres Konzeptpapier können Sie auf Anfrage während der Klausur erhalten.
- Halten Sie Begründungen oder Erklärungen bitte so kurz wie möglich. (Der auf den Lösungsblättern für eine Aufgabe vorgesehene Platz steht übrigens in keinem Zusammenhang mit dem Umfang einer korrekten Lösung!)
- Die Gesamtpunktzahl beträgt 90 Punkte. Zum Bestehen der Klausur sind mindestens 40 Punkte zu erreichen.

*Viel Erfolg und viel Glück !*

## Aufgabe 1 *Schaltfunktionen* (11 Punkte)

Gegeben ist die unvollständig definierte Schaltfunktion  $f_1(c, b, a)$ :

$$f_1(c, b, a) = \text{MINt}(1, 2, 4, 7)$$

$$f_1(c, b, a) = \text{MAXt}(5, 6)$$

1. Geben Sie die konjunktive Normalform (KNF) der Schaltfunktion  $f_1$  an. 1 P.
2. Welche der folgenden Aussagen sind richtig, welche sind falsch (bitte im Lösungsblatt ankreuzen)? Zur Ermittlung der Punktzahl in dieser Teilaufgabe werden von den richtig angekreuzten Aussagen die falsch angekreuzten Aussagen abgezogen; ein negativer Übertrag in andere Teilaufgaben erfolgt nicht. Nicht angekreuzte Aussagen zählen nicht und gehen somit nicht in die Bewertung ein. 4 P.
3. Ermitteln Sie alle Primimplikanten der Funktion  $f_1$  mit Hilfe des Nelson-Verfahrens. 2 P.
4. Die disjunktive Minimalform von  $f_1$  soll unter ausschließlicher Verwendung von NAND-Gattern realisiert werden. Gehen Sie davon aus, dass alle Eingangsvariablen sowohl bejaht als auch negiert zur Verfügung stehen. 2 P.

Zeichnen Sie das resultierende Schaltnetz. Geben Sie den Lösungsweg an.

Gegeben sei das in Bild 1 dargestellte Schaltnetz:

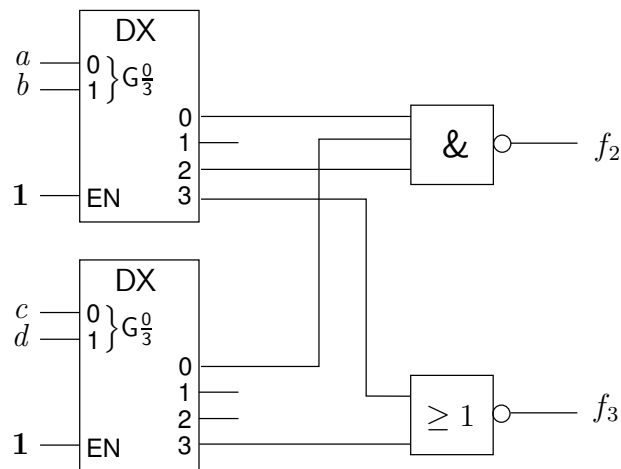


Bild 1: Schaltnetz

5. Geben Sie die Schaltfunktion  $f_2$  in minimierter Form an. 1 P.
6. Geben Sie die Schaltfunktion  $f_3$  in konjunktiver Form an. 1 P.

## Aufgabe 2 Laufzeiteffekte (8 Punkte)

Eine Schaltfunktion  $y = g(c, b, a)$  ist durch das Schaltnetz in Bild 2 realisiert.

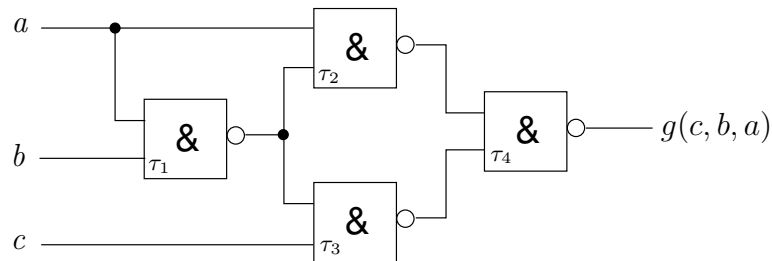


Bild 2: Schaltnetz der Schaltfunktion  $g(c, b, a)$

1. Geben Sie das endgültige Totzeitmodell des Schaltnetzes an, indem Sie jedem Gatter seinen Verzögerungswert zuweisen und alle Totzeiten zum Eingang des Schaltnetzes verschieben. Tragen Sie die Pfadvariablen in Ihrer Lösung ein und geben Sie die Werte der Pfadverzögerungen an. Die verwendeten Gatter besitzen die Totzeiten  $\tau_1$ ,  $\tau_2$ ,  $\tau_3$  und  $\tau_4$ . 3 P.
2. Übertragen Sie die Schaltfunktion  $y = g(c, b, a)$  in das im Lösungsblatt vorbereitete KV-Diagramm. 1 P.
3. Untersuchen Sie folgende Übergänge auf Funktionshasards: 2 P.
  - Übergang 1  $(c, b, a) : (0, 1, 0) \rightarrow (1, 1, 0)$
  - Übergang 2  $(c, b, a) : (0, 0, 1) \rightarrow (1, 1, 0)$

Tragen Sie die Übergänge in das KV-Diagramm aus Aufgabenteil 2 ein. Geben Sie für jeden Übergang an, ob er mit einem Funktionshasard behaftet ist oder nicht. Begründen Sie Ihre Antwort.

4. Nehmen Sie an, dass der Übergang  $(c, b, a) : (1, 0, 0) \rightarrow (1, 1, 1)$  mit einem dynamischen Strukturhasard behaftet ist. Tragen Sie den Übergang wiederum in das KV-Diagramm ein. Geben Sie ein Schaltnetz in disjunktiver Form an, bei dem der obige Übergang frei von dynamischen Strukturhasards ist. Begründen Sie Ihre Antwort. 2 P.

### Aufgabe 3 Schaltwerke I

(9 Punkte)

1. In Bild 3 ist ein flankengesteuertes JK-Flipflop mit einem asynchronen „low“-aktiven Rücksetzeingang dargestellt. 2 P.

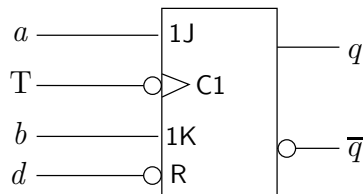


Bild 3: Das JK-Flipflop

Vervollständigen Sie das im Lösungsblatt angegebene Zeitdiagramm, indem Sie den Verlauf von  $q$  angeben.

2. Gegeben ist das in Bild 4 dargestellte Schaltwerk. Das Schaltwerk ist mit flankengesteuerten D-Flipflops realisiert. Es besitzt den Eingang  $x$  und den Ausgang  $y$ . 3 P.

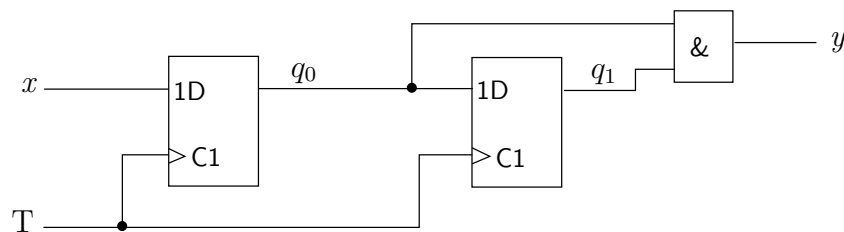


Bild 4: Das Schaltwerk

Vervollständigen Sie die Verläufe der Signale  $q_0$ ,  $q_1$  und  $y$  für den im Lösungsblatt angegebenen Verlauf von  $x$ .

3. Ein weiteres Schaltwerk ist in Bild 5 dargestellt. 4 P.

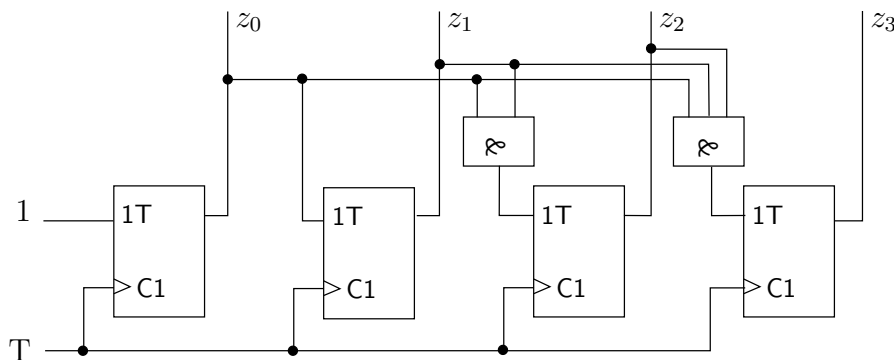


Bild 5: Das Schaltwerk II

Vervollständigen Sie das im Lösungsblatt angegebene Zeitdiagramm, indem Sie die Verläufe der Signale  $z_0$ ,  $z_1$ ,  $z_2$  und  $z_3$  einzeichnen.

## Aufgabe 4 *Schaltwerke II* (8 Punkte)

Es soll ein 3-Bit Zähler als synchrones Schaltwerk mit flankengesteuerten T-Flipflops entworfen werden. Eine Eingangsvariable  $x$  legt den Zählmodus fest.

- Für  $x = 0$  arbeitet das Schaltwerk als Dualzähler und zählt in der Reihenfolge

000, 001, 010, 011, 100, 101, 110, 111

- Für  $x = 1$  zählt das Schaltwerk im Gray-Code in der Reihenfolge

000, 001, 011, 010, 110, 111, 101, 100

1. Geben Sie den Automatengraphen des Schaltwerks an. 3 P.
2. Stellen Sie die kodierte Ablaufabelle auf. Verwenden Sie die Zustandsvariablen mit  $q_i, i = 0, 1, 2, \dots, n$ . 3 P.
3. Kurz vor Fertigungsbeginn des obigen Schaltwerkes stellt man fest, dass im Lager nur noch D-Flipflops, UND-, ODER- und Inverter-Gatter vorhanden sind. Ihre Aufgabe ist es jetzt, ein Schaltnetz zu entwerfen, das sich in Verbindung mit dem D-Flipflop wie ein T-Flipflop verhält. 2 P.

## Aufgabe 5 *Rechnerarithmetik* (10 Punkte)

**Hinweis:** Geben Sie in dieser Aufgabe *immer* den Lösungsweg an.

1. Welche Dezimalzahl wird durch das 32-Bit Wort, interpretiert als IEEE-754-Gleitkommazahl, dargestellt? 1 P.

1 1 0 0 0 0 1 0 0 1 1 0 . . . 0 0

2. Geben Sie die 16-Bit Darstellung der Zahlen  $+127_{10}$  und  $-23_{10}$  in: 2 P.

- Vorzeichen-Betrag-Form
- Zweierkomplement-Form

3. Gegeben sei ein 4-Bit-Carry-Lookahead-Addierer (siehe Bild 6).

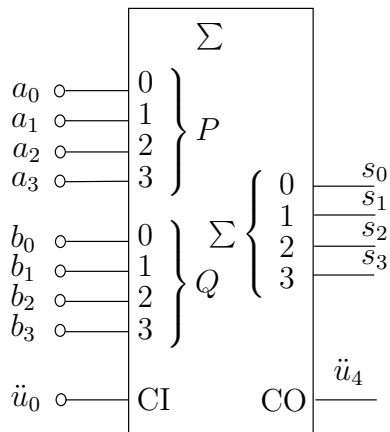


Bild 6: 4-Bit-Carry-Lookahead-Addierer

- (a) Geben Sie eine Schaltung zur Subtraktion von 4-Bit Zweierkomplementzahlen an. Dabei stehen Ihnen ein 4-Bit-Carry-Lookahead-Addierer und vier Inverter zur Verfügung. 2 P.

Für zwei 4-Bit Zahlen  $X := (x_3x_2x_1x_0)$  und  $Y := (y_3y_2y_1y_0)$  soll die Differenz  $D := Y - X$  berechnet werden. Beschriften Sie die Ein- und Ausgänge Ihrer Schaltung, so dass diese Operation ausgeführt wird.

- (b) Entwerfen Sie ein 16-Bit-Addierer aus 4-Bit-Carry-Lookahead-Addierern. Aus Ihrer Schaltung soll deutlich hervorgehen, in welcher Art und Weise die 4-Bit-Carry-Lookahead-Addierer miteinander verbunden sind. Vergessen Sie nicht, die Ein- und Ausgänge der Gesamtschaltung geeignet zu beschriften. 2 P.

4. Gegeben sind die beiden Codewörter 3 P.

- Codewort 1: **1 0 1 0 0 1 0 0 0 1 1**
- Codewort 2: **1 0 0 0 1 0 0 0 0 1 0**

Prüfen Sie beide Codewörter auf Ein-Bit-Fehler. Geben Sie die zugehörigen Datenwörter an.

## Aufgabe 6 Mikroprozessor

(6 Punkte)

In Bild 7 ist der prinzipielle Aufbau eines Mikroprozessors dargestellt.

6 P.

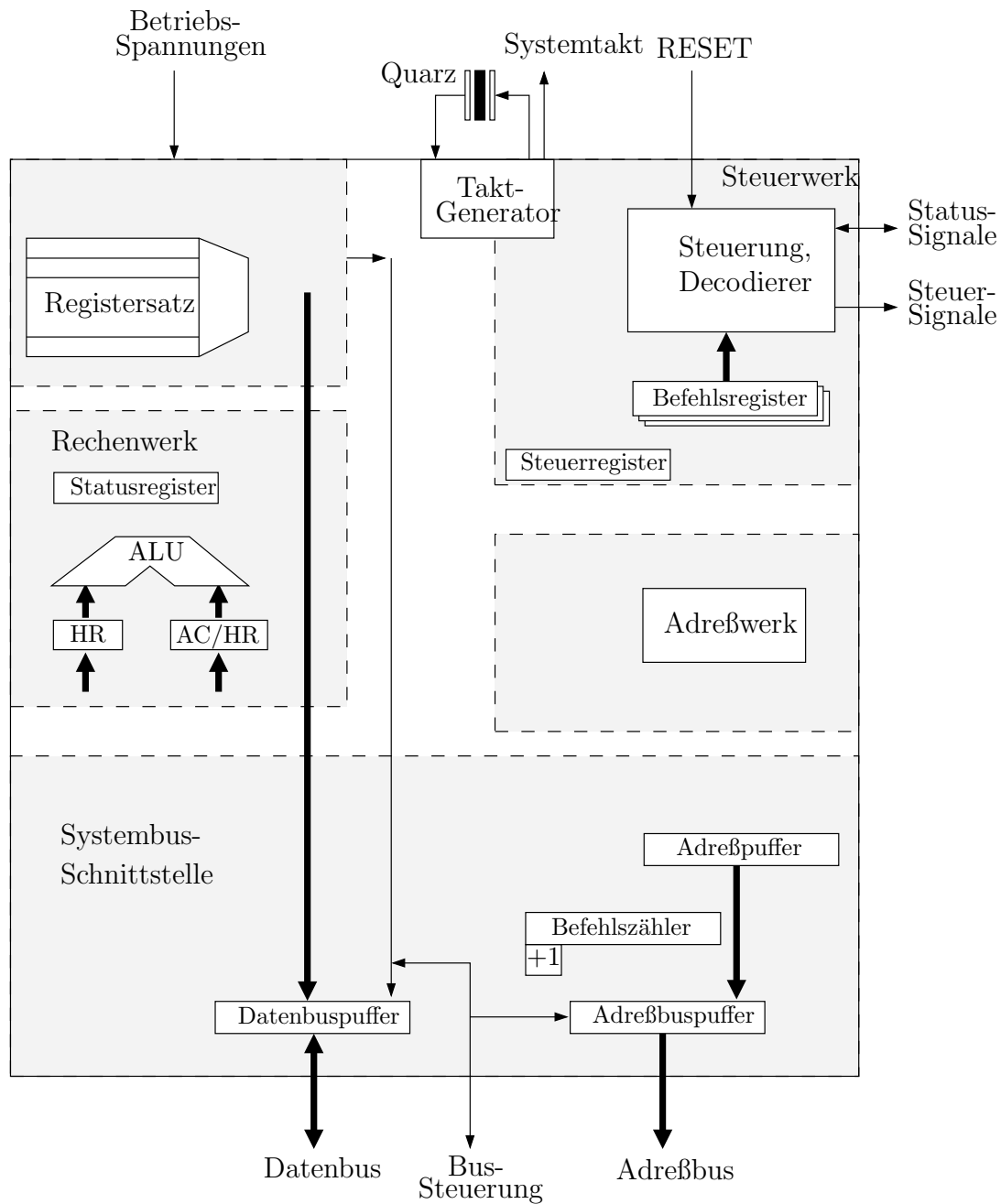


Bild 7: Prinzipieller Aufbau eines Mikroprozessors

Vervollständigen Sie die Architektur des Mikroprozessors, indem Sie alle Verbindungen des internen Busses **im Lösungsblatt** einzeichnen.

## Aufgabe 7 MIMA-Architektur (8 Punkte)

Die MIMA ist die Ihnen aus der Vorlesung bekannte mikroprogrammierte Minimalmaschine (siehe Beiblatt: **Architektur der MIMA**), die nach dem von-Neumann-Prinzip aufgebaut ist, d. h. Maschinenbefehle werden sequentiell abgearbeitet. In der Lese-Phase wird ein über IAR adressierter Befehl aus dem Speicher gelesen und im IR abgelegt. Die Lese-Phase dauert 5 Taktzyklen. Im 6. Taktzyklus wird der Befehl dekodiert (Dekodier-Phase). Die Ausführungsphase beginnt im 7. Taktzyklus. Nach der Ausführung des Befehls folgt ein Zugriff auf den nächsten Befehl.

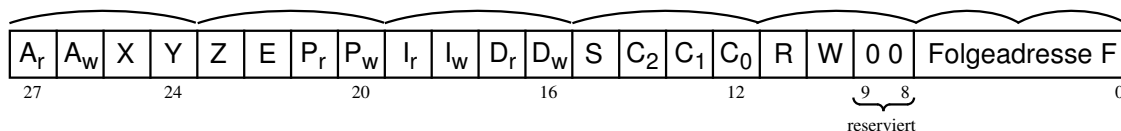
Nehmen Sie an, dass ein Hauptspeicherzugriff (Lesen und Schreiben) drei Takte dauert und währenddessen  $R = 1$  bzw.  $W = 1$  sein muss. Eine ALU-Operation sei nach einem Takt abgeschlossen.

Das Mikroprogramm für die Lese-Phase besteht aus fünf Mikrobefehlen:

- |          |                        |                      |         |   |            |
|----------|------------------------|----------------------|---------|---|------------|
| 1. Takt: | $IAR \rightarrow SAR;$ | $IAR \rightarrow X;$ | $R = 1$ | } | Lese-Phase |
| 2. Takt: | $Eins \rightarrow Y;$  | $R = 1$              |         |   |            |
| 3. Takt: | $ALU$ auf Addieren;    | $R = 1$              |         |   |            |
| 4. Takt: | $Z \rightarrow IAR$    |                      |         |   |            |
| 5. Takt: | $SDR \rightarrow IR$   |                      |         |   |            |

1. Kodieren Sie das oben angegebene Mikroprogramm für die Lese-Phase. Das Mikroprogramm soll bei der 8-Bit-Adresse 0x00 beginnen. Verwenden Sie das folgende 28-Bit-Mikrobefehlsformat:

4 P.



Beispiel:  $0x77:$   
 $0x78:$  7 0 0 0 0 7 9  $A_w = X = Y = 1$  (Akku  $\rightarrow$  X; Akku  $\rightarrow$  Y)  
 $0x79:$  Adresse des nächsten Befehls ist 0x79

2. Geben Sie die Mikroprogramme für die Ausführungsphasen der folgenden Maschinenbefehle an (jeweils ab dem 7. Takt, also nach der Lese-Phase und der Dekodier-Phase):

4 P.

LDV, STV, EQL, JMP

**Beispiel:**

AND:

- |           |                       |         |
|-----------|-----------------------|---------|
| 7. Takt:  | $IR \rightarrow SAR;$ | $R = 1$ |
| 8. Takt:  | $Akku \rightarrow X;$ | $R = 1$ |
| 9. Takt:  | $R = 1$               |         |
| 10. Takt: | $SDR \rightarrow Y$   |         |
| 11. Takt: | $ALU$ auf AND         |         |
| 12. Takt: | $Z \rightarrow Akku$  |         |

**Aufgabe 8** *Pipelining*

(12 Punkte)

1. Erläutern Sie die Aufgaben der einzelnen Pipeline-Stufen der DLX-Pipeline für

4 P.

- arithmetisch-logische Befehle
- Lade-/Speicher-Befehle

2. Gegeben sei folgendes Programm, das auf einem Prozessor mit DLX-Pipeline ohne Forwarding ausgeführt werden soll:

```

S1:   addi  $t1, $zero, 10
S2:   sll   $t2, $t1, 4
S3:   or    $t3, $t1, $t2
S4:   addi  $t4, $zero, 5
S5:   sll   $t5, $t4, 4
S6:   or    $t6, $t4, $t5
S7:   or    $t7, $t3, $t6
S8:   and   $t8, $t3, $t6

```

**Hinweise:** `sll` = *shift logical left*

(a) Bestimmen Sie alle echten Datenabhängigkeiten im Programmstück.

3 P.

(b) Der aus älteren Klausuren bekannte TI-Student hat folgende NOP-Befehle in das Programm eingefügt, um die Pipelinekonflikte zu beheben.

4 P.

```

S1:   addi  $t1, $zero, 10
      NOP
      NOP
S2:   sll   $t2, $t1, 4
      NOP
      NOP
S3:   or    $t3, $t1, $t2
S4:   addi  $t4, $zero, 5
      NOP
      NOP
S5:   sll   $t5, $t4, 4
      NOP
      NOP
S6:   or    $t6, $t4, $t5
      NOP
      NOP
S7:   or    $t7, $t3, $t6
S8:   and   $t8, $t3, $t6

```

Verringern Sie die Anzahl der NOP-Befehle durch Umordnen der Befehle, ohne das Ergebnis zu verändern.

(c) Was steht nach der Ausführung des Programms in den Registern `$t7` und `$t8`?

1 P.

## Aufgabe 9 Cache-Speicher (10 Punkte)

Gegeben sei ein direkt-abgebildeter Cache (*direct mapped*, Abkürzung: **DM**) mit einer Speicherkapazität von 128 Byte und einer Blockgröße von 16 Bytes. Als Aktualisierungsstrategie wird das Rückschreib-Verfahren (*write back*) verwendet. Die Hauptspeicheradresse ist 32 Bit breit.

1. Skizzieren Sie die Unterteilung der Hauptspeicheradresse. 1 P.
2. Zur Verwaltung eines Cacheblocks werden zwei Statusbits verwendet: ein *Valid*-Bit (Abkürzung: *V*) und ein *Dirty*-Bit (Abkürzung: *D*). Berechnen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers. Geben Sie den Lösungsweg an. 2 P.

Der Zustand des Cache-Speichers sei durch Tabelle 1 angegeben.  $V = 1$  kennzeichnet einen gültigen Eintrag im Cache.  $D = 1$  kennzeichnet einen Eintrag im Cache, der gegenüber seiner Originalkopie verändert wurde.

Cache-Speicher			
Zeile	<i>D</i> -Bit	<i>V</i> -Bit	Tag
0	0	1	1
1	0	1	1
2	0	0	4
3	0	1	5
4	1	1	0
5	0	1	3
6	1	1	0
7	0	0	1

Tabelle 1: Anfangsbelegung des Cache-Speichers

Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die in hexadezimaler Schreibweise angegebenen Hauptspeicheradressen:

Adresse	0x44	0xA0	0xC3	0x9E	0x66	0x2D	0x6B	0x49
read/write	w	r	w	r	r	w	r	w

3. Geben Sie an, ob es sich beim Zugriff auf die jeweiligen Adressen um einen Cache-Miss oder einen Cache-Hit handelt. Verwenden Sie dabei „—“ für Cache-Miss und „×“ für Cache-Hit. Geben Sie an, ob der entsprechende Cacheblock in den Hauptspeicher zurückkopiert werden muss (**ja**) oder nicht (**nein**). 4 P.
4. Skizzieren Sie den schematischen Aufbau des Cache-Speichers, indem Sie die im Lösungsblatt vorbereitete Skizze vervollständigen. Aus Ihrer Zeichnung sollen die Unterteilung der Hauptspeicheradresse, die Daten- und Adress-Speicher, die Statusbits, alle Vergleicher, Multiplexer/Dekoder und das Hit-Signal sowie die Breiten der Daten- und Adressleitungen deutlich sein. 3 P.

## Aufgabe 10 *Verschiedenes*

(9 Punkte)

1. Was sind *Tristate*-Treiber? 1 P.
2. Was ist ein *Translation-Lookaside-Buffer*? 2 P.
3. Nennen Sie zwei programmierbare und zwei nicht programmierbare Systemsteuerbausteine. 2 P.
4. Was versteht man unter einer Speicherhierarchie? Welche grundlegenden Annahmen liegen dem Konzept der Speicherhierarchie zugrunde? 2 P.
5. Es soll ein 128-KByte-Cache-Speicher eines Rechners mit einer Speicherwortbreite von 32-Bit konzipiert werden. 2 P.
  - (a) Welche Speicherbausteine würden Sie verwenden? Begründen Sie Ihre Antwort.
  - (b) Es stehen Speicherbausteine zur Verfügung, die als  $8K \times 4$  Bit organisiert sind. Wie viele Speicherbausteine dieser Art sind zur Realisierung des Cache-Speichers notwendig?