

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 09. August 2019, 12:30 – 14:30 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 Endergebnisse und Rechenweg 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 so kurz und präzise wie möglich. Der auf den Lösungsblättern für eine Aufgabe vorgesehene Platz lässt nicht auf den Umfang einer korrekten Lösung schließen.
- Die Gesamtpunktzahl beträgt 90 Punkte. Zum Bestehen der Klausur sind mindestens 40 Punkte zu erreichen.

Viel Erfolg und viel Glück!

Aufgabe 1 *Schaltfunktionen*

(10 Punkte)

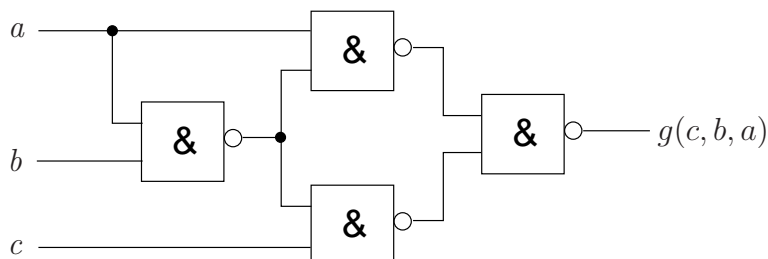
1. Bestimmen Sie die konjunktive Normalform (KNF) der Funktion
- $f(w, x, y, z)$

2 P.

$$f(w, x, y, z) = \bar{y}(x\bar{z} \vee z) \vee w\bar{x}(y \vee \bar{z}) \vee \bar{x}yz$$

2. Bestimmen Sie
- algebraisch*
- die disjunktive Minimalform (DMF) der Funktion
- $g(c, b, a)$
- , die durch das Schaltnetz in Abbildung 1 realisiert ist.

3 P.

Abbildung 1: Schaltnetz der Funktion $g(c, b, a)$

Gegeben sei die Überdeckungstabelle einer weiteren Funktion z mit den Nullstellen a, b, c, d und e sowie allen Primimplikaten A, B, C, D und E der Funktion z (Siehe Tabelle 1). Die Realisierungskosten eines Primimplikaten sind umgekehrt proportional zur Anzahl der von ihm überdeckten Nullstellen.

Prim- implikate	Nullstellen				
	a	b	c	d	e
A		\times	\times		
B	\times				\times
C	\times			\times	
D	\times	\times			\times
E		\times	\times		\times

Tabelle 1: Überdeckungstabelle der Funktion z

3. Vereinfachen Sie die Tabelle, indem Sie den/die Kern-Primimplikat(e) ermitteln und die davon überdeckten Nullstellen in Tabelle 1 streichen.

1 P.

Zeichnen Sie die so reduzierte Tabelle.

4. Vereinfachen Sie Ihre Tabelle aus Aufgabenteil 3 mit den Regeln der Spalten-
-
- dominanz.

1 P.

Zeichnen Sie die so reduzierte Tabelle.

5. Ergänzen Sie Ihre Tabelle im Aufgabenteil 4 um eine Kostenspalte und vereinfachen Sie die Tabelle dann mit den Regeln der Zeilendominanz.

2 P.

Zeichnen Sie die so reduzierte Tabelle.

6. Geben Sie die minimale Form der Funktion
- z
- an.

1 P.

Aufgabe 2 *CMOS-Technologie*

(5 Punkte)

In Abbildung 2 ist eine Teilrealisierung einer Schaltfunktion $y = f(x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ in der CMOS-Technologie dargestellt.

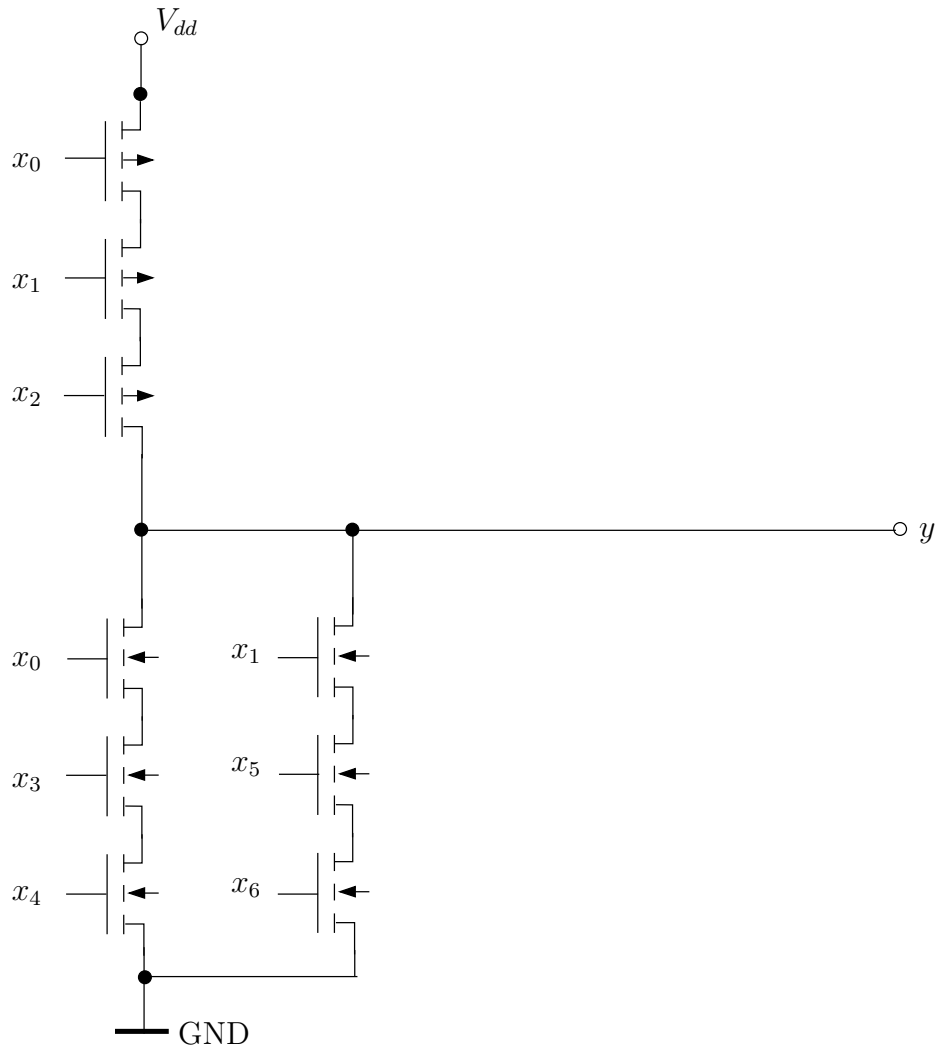


Abbildung 2: Teilrealisierung einer CMOS-Schaltung

1. Ergänzen Sie die Schaltung im p- und n-Teil so, dass eine Realisierung der Schaltfunktion y in der CMOS-Technologie entsteht.
2. Welche Schaltfunktion haben Sie realisiert?

4 P.

1 P.

Aufgabe 3 Laufzeiteffekte

(6 Punkte)

Gegeben ist das in Abbildung 3 dargestellte Schaltnetz. NAND-, NOR- und OR-Gatter haben eine Totzeit von 5 ns , das XOR-Gatter von 7 ns und der Inverter von 2 ns . Die Eingangsvariablen a , b und c wechseln zum Zeitpunkt $t = 0$ gleichzeitig von 0 auf 1.

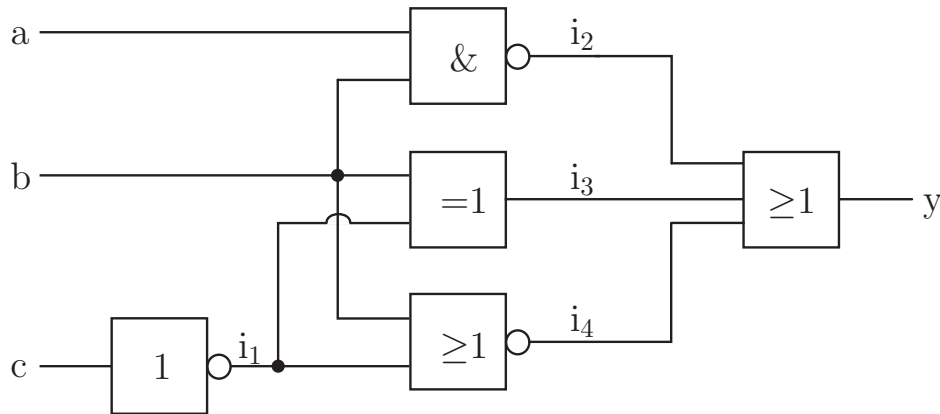


Abbildung 3: Schaltnetz

1. Geben Sie die Verläufe der Signale i_1 , i_2 , i_3 , i_4 und y an, indem Sie das im Lösungsblatt angegebene Zeitdiagramm vervollständigen. 3 P.
2. Treten im Zeitverlauf von y Hasardfehler auf? Falls ja, um welchen Typ handelt es sich bei dem zu Grunde liegenden Hasard? Begründen Sie Ihre Antwort. 3 P.

Aufgabe 4 *Schaltwerke*

(12 Punkte)

1. Gegeben ist das in Abbildung 4 dargestellte Schaltwerk.

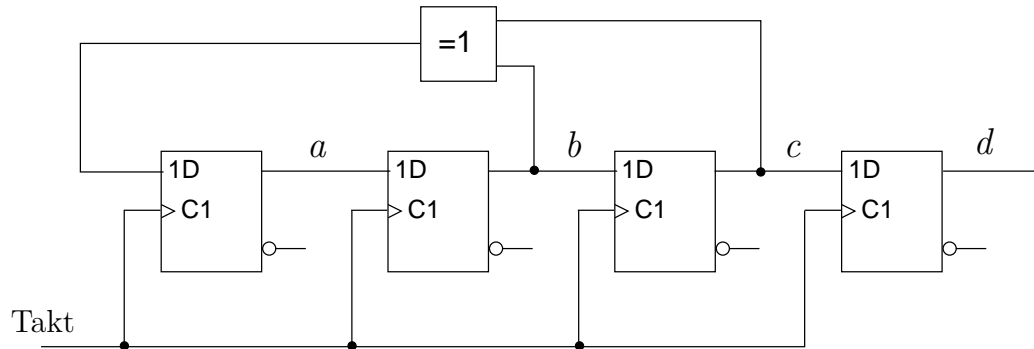


Abbildung 4: Schaltwerk

- (a) Ist das Schaltwerk synchron oder asynchron? 1 P.
- (b) Wie viele Zustände kann das Schaltwerk maximal annehmen? 1 P.
- (c) Vervollständigen Sie die Verläufe der Signale a , b , c und d im Lösungsblatt. 4 P.

2. Es soll ein synchroner modulo-8-Rückwärtszähler mit flankengesteuerten T-Flipflops entworfen werden.

- (a) Geben Sie den Automatengraphen des Zählers an. 2 P.
- (b) Stellen Sie die kodierte Ablaufabelle des Zählers auf. Verwenden Sie hierzu die im Lösungsblatt vorbereitete Tabelle. Die Zustände des Zählers seien mit Hilfe der Zustandsvariablen q_2 , q_1 und q_0 dual kodiert. 1 P.
- (c) Geben Sie die Ansteuerfunktionen der verwendeten Flipflops in minimaler Form an. 1 P.
- (d) Zeichnen Sie das Schaltbild des Zählers. 2 P.

Aufgabe 5 *Rechnerarithmetik*

(12 Punkte)

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

1. Finden Sie die Basen
- r
- und
- s
- so dass gilt:

$$12_r = 111_s$$

1 P.

2. Geben Sie den dezimalen Wert der größten Zahl an, die repräsentiert werden kann mit

2 P.

- 12 binären Stellen
- 4 hexadezimalen Stellen

3. Geben Sie die Darstellung der Zahl
- 2005_{10}
- im

3 P.

- 32-Bit Zweierkomplement-Format
- 32-Bit IEEE-754-Gleitkomma-Format.

an.

4. Ein BCD-Addierer für eine Tetrade soll aus zwei 4-Bit-Volladdierern und einem Schaltnetz zur Pseudotetraden- und Übertragskorrektur realisiert werden (siehe Bild 5).

3 P.

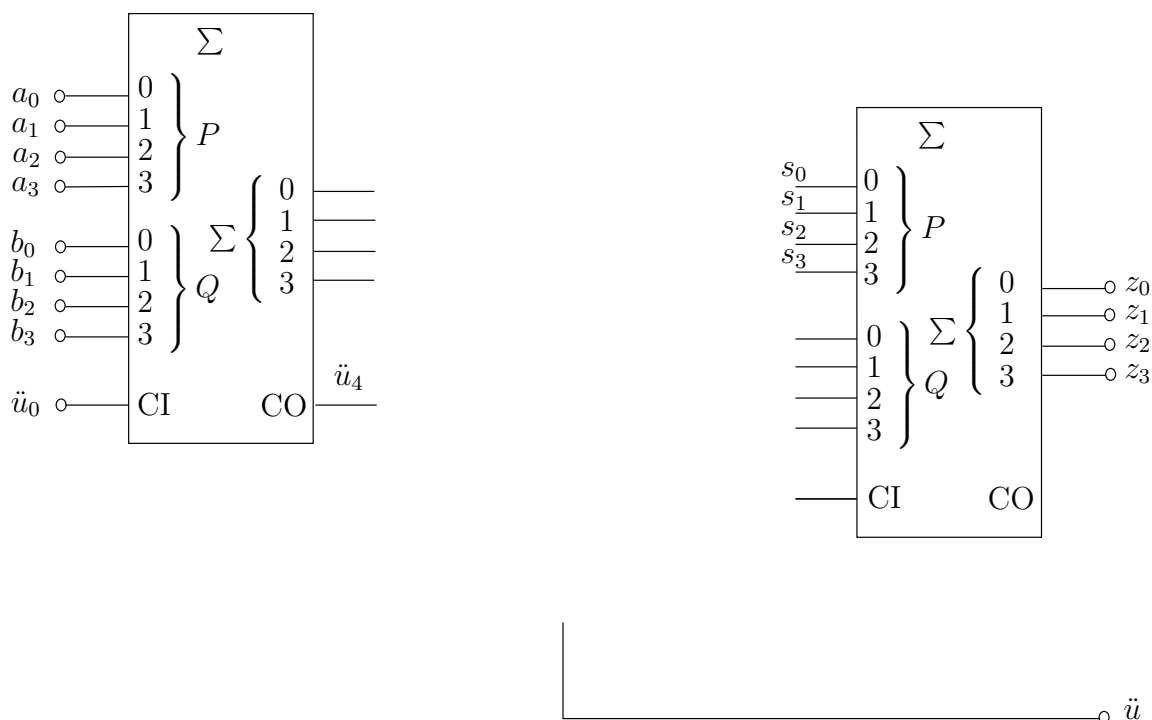


Abbildung 5: BCD-Addierer für eine Tetrade

Ergänzen Sie das im Lösungsblatt wiederholt angegebene Schaltbild. Dabei stellt $(s_3 s_2 s_1 s_0)$ das Dual-Ergebnis der Addition der Tetraden $(a_3 a_2 a_1 a_0)$ und $(b_3 b_2 b_1 b_0)$ dar; \ddot{u}_4 ist der sich dabei ergebende Übertrag. \ddot{u} ist der Übertrag der BCD-Addition der beiden Tetraden.

5. Der Hauptspeicher eines Rechners mit 8-Bit Datenwortbreite unterstützt eine Einzelfehler-Korrektur(Hamming-Code). Aus dem Speicher erhält man die beiden Codewörter:

3 P.

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

Prüfen Sie beide Codewörter auf Fehler, die beim Übertragen oder Speichern entstanden sein könnten und korrigieren Sie diese (falls vorhanden). Geben Sie die zugehörigen Datenwörter an.

Aufgabe 6 *Mikroprozessor* (6 Punkte)

In Abbildung 6 ist das interne Bussystem eines Mikroprozessors dargestellt. Bei dieser Bus-Architektur ist nur eine geringe Parallelität bei der Befehlsbearbeitung möglich.

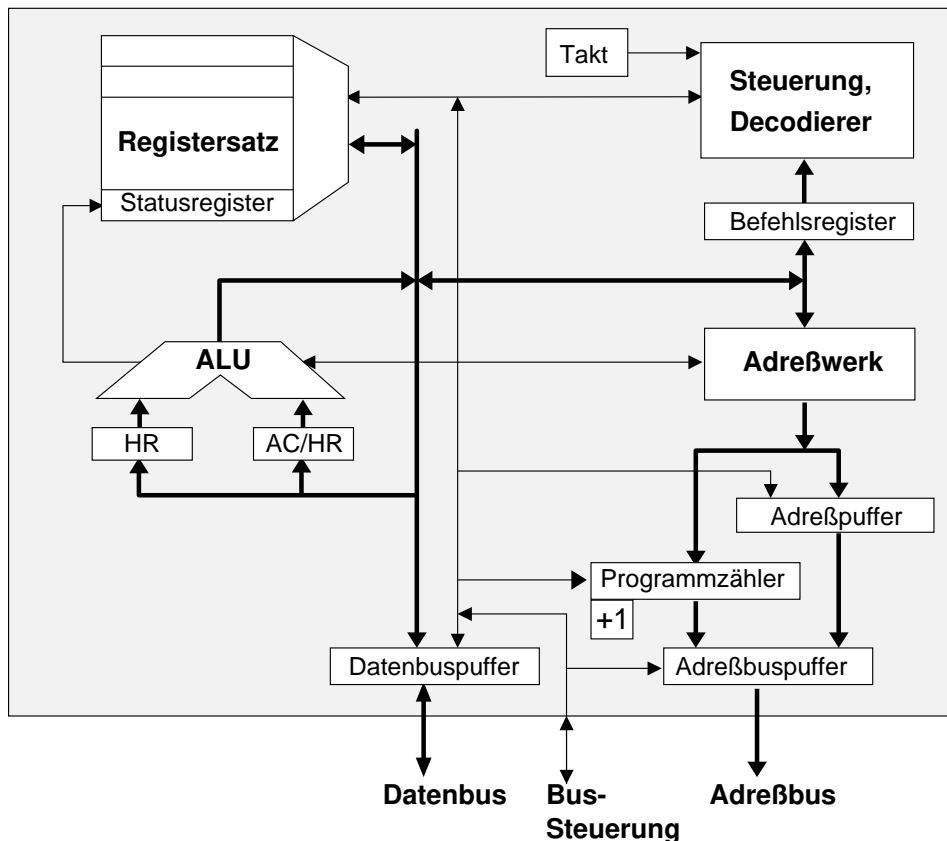


Abbildung 6: Das interne Bussystem eines Mikroprozessors

1. Nehmen Sie an, dass das Befehlsregister als Warteschlange (*OpCode prefetch queue*) realisiert ist.

Ändern Sie die Bus-Architektur so, dass parallel zu Aktionen im Rechenwerk und Registersatz ein *OpCode Prefetching* möglich wird.

2 P.

2. Während das Ergebnis der zuletzt ausgeführten Operation im Rechenwerk in den Registersatz transportiert wird, soll ein Hilfsregister der ALU mit dem nächsten Operanden aus dem Datenbuspuffer oder aus dem Registersatz (sofern nicht das gleiche Register hierfür benötigt wird) geladen werden.

Ändern Sie die Bus-Architektur so, dass diese Parallelisierung im Rechenwerk möglich wird.

2 P.

3. Ändern Sie die Bus-Architektur so, dass zwei Operanden parallel an die Eingänge der ALU geführt werden können. Dabei soll der eine Operand immer aus dem Registersatz stammen, während der andere Operand wahlweise aus dem Registersatz oder dem Datenbuspuffer stammt.

2 P.

Aufgabe 7 *MIPS-Assembler*

(12 Punkte)

1. Übersetzen Sie die folgenden C-Kontrollstrukturen in MIPS-Assembler. Sie dürfen dabei **nur** die MIPS-Befehle `add`, `lw`, `beq`, `bne` und `j` verwenden.

Nehmen Sie an, dass die Variablen `i`, `j` und `k` in den Registern `$s3`, `$s4` und `$s5` stehen und dass sich die Anfangsadresse von `A` im Register `$s6` befindet. Dabei ist `A` ein Feld aus 32-Bit Integerzahlen. Verwenden Sie die Register `$t0` und `$t1` zur Speicherung temporärer Variablen.

(a) do-while-Schleife:

3 P.

```
do
i = i + j;
while ( A[i] == k )
```

(b) if-else-Anweisungen:

5 P.

```
if (i == j)
i = i + A[k];
else if (i == k)
i = i + A[j];
else
i = j + k;
```

2. Was ist der Unterschied zwischen Maschinensprache und Assemblersprache?

1 P.

3. Was bewirkt die Assemblerdirektive `.align 2`?

1 P.

4. Beim MIPS-Befehlssatz sind die höchstwertigen 6 Bits (Bits: 31 ... 26) im Befehlsformat für den OpCode reserviert. Mit 6 Bits lassen sich 64 verschiedene Kombinationen bilden. Bedeutet dies, daß der MIPS-Befehlssatz aus maximal 64 Instruktionen besteht? Erläutern Sie Ihre Antwort.

2 P.

Aufgabe 8 *Pipelining*

(9 Punkte)

1. Bestimmen Sie alle Datenabhängigkeiten im folgenden Programmstück. Geben Sie jeweils die Art der Abhängigkeit und das betroffene Register an.

3 P.

```
S1:      addi   $t1, $t0, 1
S2:      srli   $t2, $t1, 2
S3:      or     $t3, $t1, $t2
S4:      srli   $t1, $t3, 4
S5:      add    $t4, $t2, $t3
```

2. Das folgende Programmstück soll auf einem Prozessor mit einer DLX-Pipeline ohne Forwarding ausgeführt werden. Fügen Sie in das Programmstück möglichst wenige NOP-Befehle, damit keine Konflikte auftreten.

4 P.

```
S1:      anfang: andi $t2, $t1, 1
S2:                      beqz $t2, weiter
S3:                      subi $t1, $t1, 1
S4:                      j anfang
S5:      weiter:  srli $t1, $t1, 1
S6:                      j anfang
S7:                      addi $t3, $t0, 1
```

3. Warum können Ausgabeabhängigkeiten (*output dependence*) und Gegenabhängigkeiten (*anti-dependence*) in der DLX-Pipeline nicht zu Konflikten führen?

2 P.

Aufgabe 9 *Cache-Speicher*

(10 Punkte)

1. Gegeben sei ein 2-fach-satzassoziativer Cache-Speicher (*2-way-set-associative cache*) mit der folgenden Unterteilung der Hauptspeicheradresse

31	16	15	5	4	0
Tag			Index		Byte-Offset

- (a) Wie viele Bytes enthält ein Cache-Block? 1 P.
- (b) Wie groß ist die Kapazität des Cache-Speichers? 1 P.
- (c) Bestimmen Sie den insgesamt erforderlichen Speicherbedarf für die Realisierung des Cache-Speichers? Nehmen Sie dabei an, dass zwei Statusbits (*Valid* und *Dirty*) zur Verwaltung eines Cacheblocks verwendet werden. 2 P.
- (d) Der Prozessor greift auf die Speicheradresse 0x00EF1A34 zu. Mit wie vielen und welchen Zeilen im Cache wird ein Vergleich durchgeführt, um herauszufinden, ob ein Cache-Hit vorliegt? 2 P.
2. Gegeben sei ein direkt abgebildeter Cache-Speicher (*direct mapped cache*) 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. Zur Verwaltung eines Cacheblocks werden zwei Statusbits verwendet: ein *Valid*-Bit *V* und ein *Dirty*-Bit *D*. 4 P.

Der Zustand des Cache-Speichers sei durch Tabelle 2 angegeben. Dabei kennzeichnet $V = 1$ einen gültigen Eintrag im Cache und $D = 1$ 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 2: Anfangsbelegung des Cache-Speichers

Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die angegebenen Hauptspeicheradressen, welche nacheinander von links nach rechts erfolgen:

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

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**).

Aufgabe 10 Virtuelle Speicherverwaltung (8 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 1 KByte, 8 virtuellen Seiten und 4 physikalischen Seiten (Frames). Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 3 angegeben.

Virtuelle Seitennummer	Physikalische Seitennummer
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

Tabelle 3: Seitentabelle

1. Skizzieren Sie die Unterteilung der 32 Bit breiten virtuellen Adresse. 1 P.
2. Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen: 4 P.

1023, 1024, 4204, 6200

Zur Beschleunigung der Adressberechnung soll ein Cache-Speicher als *Translation-Lookaside-Buffer (TLB)* eingesetzt werden, der die letzten 32 Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle speichert.

3. Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen *Translation Lookaside Buffer (TLB)* erreicht? 1 P.
4. Wie breit ist der *Tag* eines Cache-Eintrags? Gehen Sie dabei von einer n Bit breiten virtuellen Adresse, einer m Bit breiten physikalischen Adresse und einer Seitengröße von 4 KByte aus. 2 P.