



UNIVERSITÄT KARLSRUHE (TH)
Fakultät für Informatik
Institut für Rechnerentwurf und Fehlertoleranz (IRF)
Prof. Dr. W. Karl

Aufgabenblätter

zur Klausur „Technische Informatik I/II“
am 29. Januar 2005, 10.00 - 12.00 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

(10 Punkte)

1. Eine unvollständig definierte Schaltfunktion $y = f(d, c, b, a)$ sei durch ihre Eins- und *don't care*-Stellen (Abkürzung d) gegeben:

$$y = \text{MINt}(0, 1, 7, 8, 15) \vee d(4, 9)$$

- (a) Tragen Sie alle Primimplikanten der Funktion ins KV-Diagramm im Lösungsblatt ein und geben Sie eine disjunktive Minimalform (DMF) der Funktion f an. 2 P.

- (b) Tragen Sie alle Primimplikate der Funktion ins KV-Diagramm im Lösungsblatt ein und geben Sie eine konjunktive Minimalform (KMF) der Funktion f an. 2 P.

2. Gegeben sei die Überdeckungstabelle einer Schaltfunktion $h(e, d, c, b, a)$ mit den Mintermen 0, 1, 4, 6, 7, 13, 15, 19, 31. Die Primimplikanten der Schaltfunktion seien A, B, C, D, E und F.

	0	1	4	6	7	13	15	19	31
A	×	×	×						
B		×							
C			×	×	×				
D					×	×	×		
E					×			×	
F					×		×	×	×

Es seien:

$$\begin{array}{lll} A = (0\ 0\ -\ 0\ -) & B = (-\ 0\ -\ 0\ 1) & C = (0\ 0\ 1\ -\ -) \\ D = (0\ -\ 1\ -\ 1) & E = (-\ 0\ 1\ 0\ 1) & F = (-\ -\ 1\ 1\ 1) \end{array}$$

- (a) Welche Primimplikanten sind Kern-, Wahl- und welche sind entbehrliche Primimplikanten? 1 P.

- (b) Bestimmen Sie die disjunktive Minimalform der Funktion $h(e, d, c, b, a)$. Beschreiben Sie Ihre Vorgehensweise. 2 P.

3. Beweisen Sie *schaltalgebraisch* die Identität 3 P.

$$xu \vee \bar{x}w = xu \vee \bar{x}w \vee uw$$

Aufgabe 2 CMOS-Technologie

(8 Punkte)

1. Geben Sie die Schaltfunktion $g(d, c, b, a)$ an, die durch das in Bild 1 dargestellte CMOS-Schaltnetz realisiert wird. 2 P.

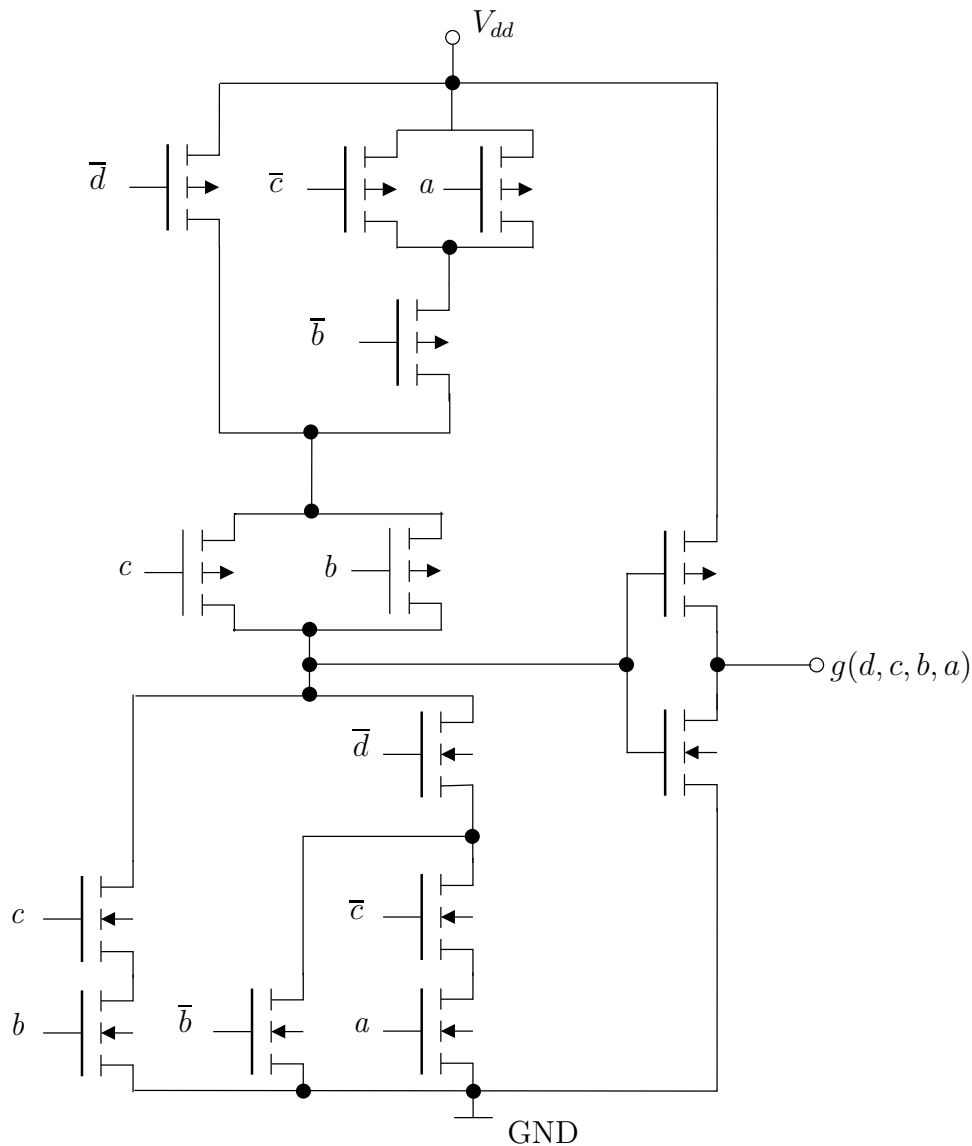


Bild 1: CMOS-Schaltnetz

2. Geben Sie die Funktionstabelle und das Transistor-Schaltbild eines *Tri-State*-Inverters an. 3 P.
3. Skizzieren Sie den Aufbau eines nMOS-Transistors. Aus Ihrer Zeichnung müssen die unterschiedlich dotierten Zonen und die Anschlüsse des Transistors eindeutig erkennbar sein. 3 P.

Aufgabe 3 Schaltwerke

(11 Punkte)

1. Gegeben sei ein synchrones Schaltwerk mit 2 flankengesteuerten JK-Flipflops, einer Eingangsvariablen x und einer Ausgangsvariablen y (siehe Bild 2).

4 P.

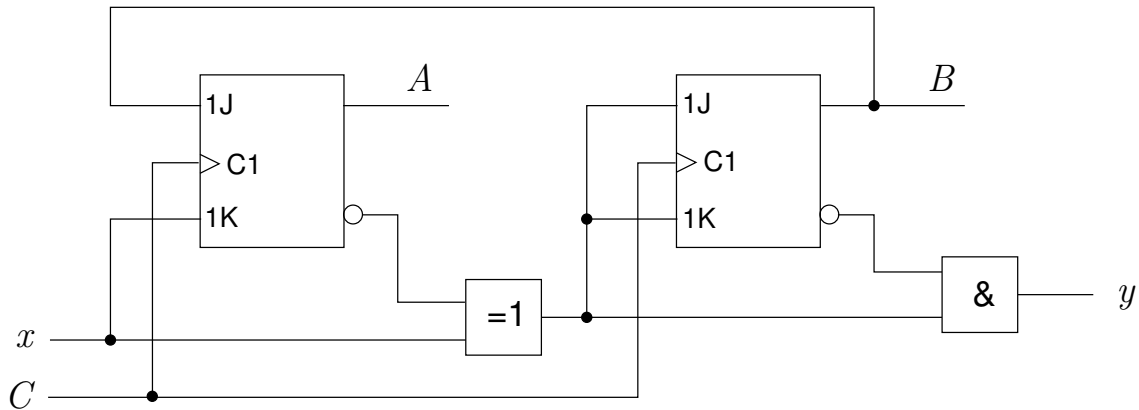


Bild 2: Schaltwerk I

Stellen Sie die kodierte Ablaufabelle des Schaltwerks auf. Stellen Sie Ihre Lösung schrittweise dar.

Hinweis: Bestimmen Sie die Ansteuerfunktionen der Flipflops, die Zustandsübergangsgleichungen und die Ausgabefunktion des Schaltwerks.

2. In Bild 3 ist der Automatengraph eines zweiten synchronen Schaltwerks mit der Eingangsvariablen e , der Ausgangsvariablen a und den Zuständen Z_0, Z_1, Z_2 und Z_3 dargestellt.

3 P.

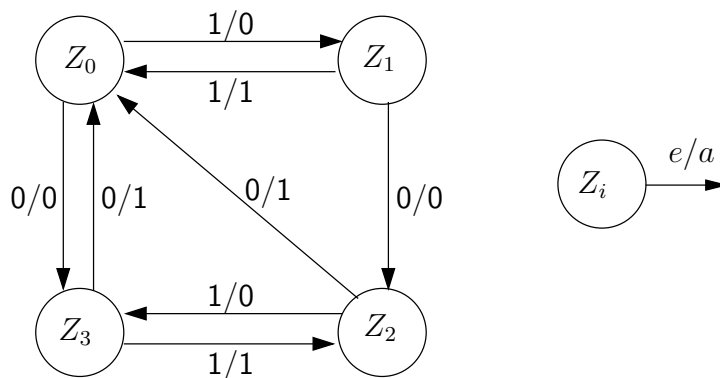


Bild 3: Automatengraph vom Schaltwerk II

Nehmen Sie an, dass sich das Schaltwerk am Anfang im Zustand Z_0 befindet. Vervollständigen Sie die im Lösungsblatt angegebene Tabelle der Zustands-, Eingabe- und Ausgabefolgen des Schaltwerks.

3. In Bild 4 ist ein weiteres Schaltwerk dargestellt.

4 P.

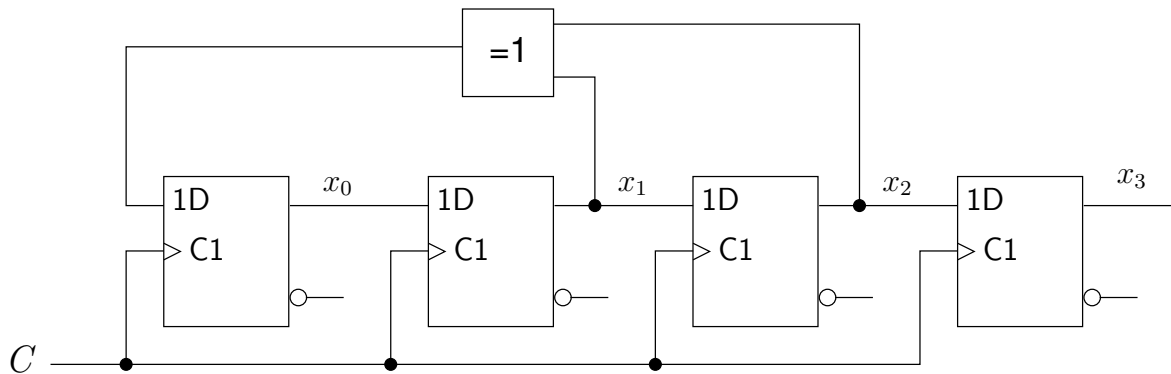


Bild 4: Schaltwerk III

Vervollständigen Sie die Verläufe der Signale x_0, x_1, x_2 und x_3 im angegebenen Zeitdiagramm im Lösungsblatt.

Aufgabe 4 Rechnerarithmetik

(12 Punkte)

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

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

1 P.

$$12_r = 111_s$$

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 auf der nächsten Seite).

3 P.

Ergänzen Sie das im Lösungsblatt wiederholt angegebene Schaltbild. Dabei stellt $(s_3s_2s_1s_0)$ das Dual-Ergebnis der Addition der Tetraden $(a_3a_2a_1a_0)$ und $(b_3b_2b_1b_0)$ dar; \ddot{u}_4 ist der sich dabei ergebende Übertrag. \ddot{u} ist der Übertrag der BCD-Addition der beiden Tetraden.

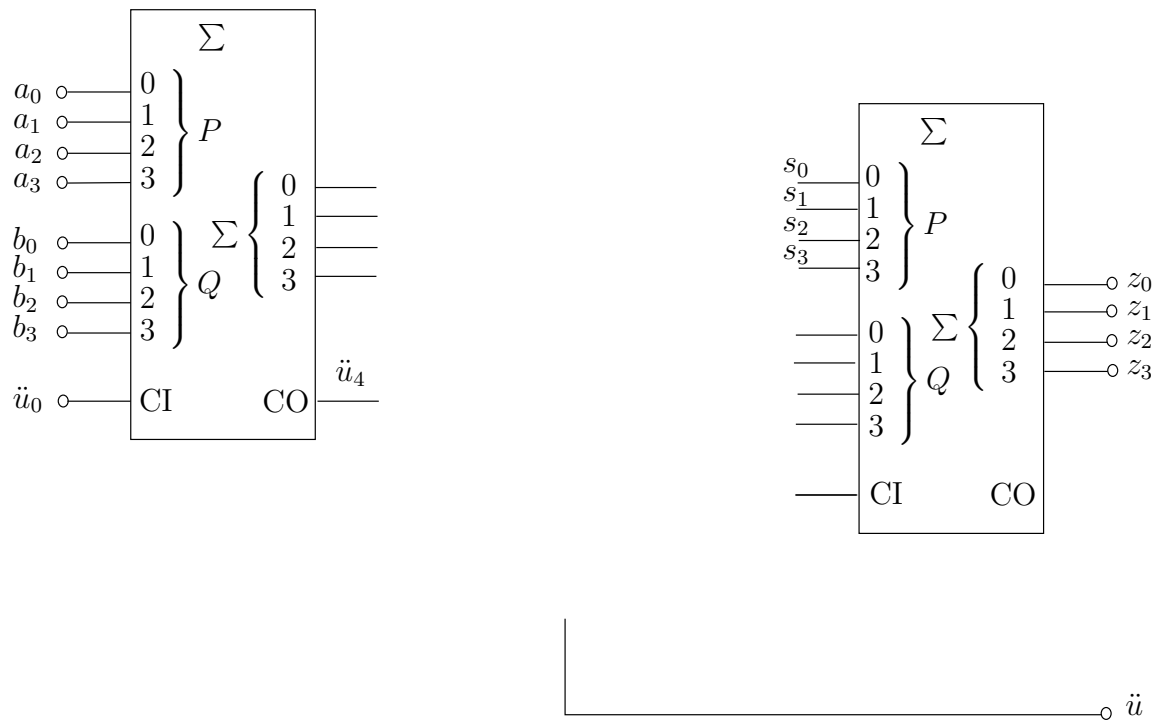


Bild 5: BCD-Addierer für eine Tetrade

5. Der Hauptspeicher eines Rechners mit 8-Bit Datenwortbreite unterstützt eine Einzelfehler-Korrektur. 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 5 *Multiple Choice* (4 Punkte)

Kreuzen Sie bitte für jede der Behauptungen im Lösungsblatt an, ob sie Ihrer Meinung nach richtig oder falsch ist. Nicht angekreuzte Behauptungen zählen nicht und gehen somit nicht in die Bewertung ein. Zur Ermittlung der Punktzahl werden von den richtig angekreuzten Behauptungen die falsch angekreuzten Behauptungen abgezogen; ein negativer Übertrag in andere Teilaufgaben erfolgt nicht.

Aufgabe 6 *MIMA-Architektur* (5 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.

Geben Sie das Mikroprogramm für die Lese-Phase (Fetch-Phase) in Register-Transfer-Schreibweise an, d. h. in der Form

5 P.

- 1. Takt: $IR \rightarrow SAR; \quad R = 1$
- 2. Takt:
- ⋮

Aufgabe 7 *MIPS-Assembler* (12 Punkte)

1. Schreiben Sie die folgenden C-Kontrollstrukturen in MIPS-Assembler um.

3 P.

```
for (i = 0; i < 100; i++) j = j * i;
```

Die Variablen i und j stehen in den Registern $\$a0$ und $\$a1$. Verwenden Sie das Register $\$v0$ zur Speicherung temporärer Variablen.

2. Das folgende Programmstück soll die Summe der Elemente eines Arrays aus 32-Bit Integer-Zahlen in Zweierkomplement-Form berechnen. Das Register $\$a0$ sei mit der Adresse des Arrays initialisiert; das Register $\$a1$ sei mit der Anzahl der Array-Elemente initialisiert. Alle anderen Register seien nicht initialisiert.

3 P.

```
array_sum:      lw     $t0, 0($a0)
                add   $v0, $v0, $t0
                addi  $a0, $a0, 1
                addi  $a1, $a1, -1
                bltz  $a0, array_sum
```

Leider haben sich bei der Implementierung einige Fehler eingeschlichen. Finden Sie diese Fehler und korrigieren Sie das Programm, so dass es korrekt arbeitet.

3. Geben Sie für das folgende MIPS-Programmstück den Inhalt des Zielregisters in hexadezimaler Schreibweise nach der Ausführung des jeweiligen Befehls an.

3 P.

```
ori    $s1, $zero, 0x10
slr    $s2, $s1, 3
slti   $s3, $s2, 100
sub    $s4, $s3, $s2
lui    $s5, 0x20
```

4. Gegeben sei das folgende MIPS-Programmstück:

```
.data
vec:   .word 8, 13, 17, 19, 23, 29, 31, 37, 41, 43

.text
main:  lw $t1, vec
       lw $t2, vec+0x18
       lw $t3, vec($t1)
       lw $t4, vec+0x14($t1)
```

- (a) Geben Sie die Inhalte der Register `$t1`, `$t2`, `$t3` und `$t4` in hexadezimaler Schreibweise nach der Ausführung des obigen Programmcodes an.
- (b) Geben Sie den MIPS-Code an, mit dem man die Adresse von `vec` im Register `$s0` speichert.

2 P.

1 P.

Aufgabe 8 *Cache-Speicher*

(12 Punkte)

1. Bei einem Cache-Speicher mit einer Speicherkapazität von 512 KByte ist die Hauptspeicheradresse in ein 16 Bit Tag-Feld, ein 10 Bit Index-Feld und einen 6 Bit Byte-Offset unterteilt. Geben Sie bei der Beantwortung der folgenden Fragen den Lösungsweg an.
- (a) Bestimmen Sie die Blockgröße in Bytes.
- (b) Wieviele Einträge besitzt der Cache-Speicher?
- (c) Wie ist der Cache-Speicher organisiert?
2. Es soll ein 5-fach-assoziativer (*5-way set associative cache*) Cache-Speicher mit 128 Sätzen und einer Blockgröße von 8 Byte realisiert werden. Nehmen Sie an, dass die Hauptspeicheradresse 32 Bit breit ist. Zur Verwaltung eines Cacheblocks wird nur ein Statusbit (*Valid*-Bit: V) verwendet.

1 P.

1 P.

2 P.

- (a) Welche Bits der 32-Bit-Adresse bilden Offset, Tag und Index? Skizzieren Sie hierzu die Unterteilung der Hauptspeicheradresse. 1 P.
- (b) Bestimmen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers. 3 P.
3. Bild 6 zeigt ein Mikrorechnersystem, das neben dem Prozessor mit Cache einen DMA-Controller ohne Cache als weiteren Master aufweist. Der DMA-Controller hat wie der Prozessor einen direkten Zugriff auf den Hauptspeicher, in dem sich beide Master einen Speicherbereich teilen.

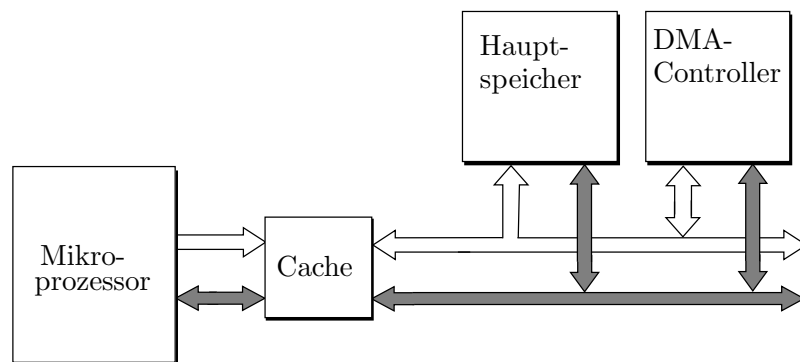


Bild 6: Mikrorechnersystem mit DMA-Controller

- (a) Welches Daten-Inkonsistenz-Problem tritt auf, wenn der Cache ein Durchschreib-Verfahren (*write-through*) verwendet? 1 P.
- (b) Welches Daten-Inkonsistenz-Problem tritt auf, wenn der Cache ein Rückschreib-Verfahren (*write-back*) verwendet? 1 P.
- (c) Geben Sie zwei Lösungsvorschläge zur Gewährleistung der Datenkonsistenz an. 2 P.

Aufgabe 9 Speicherverwaltung (6 Punkte)

Die Speicherverwaltung in einem Rechnersystem geschieht zweistufig über eine Segmenttabelle und eine Seitentabelle. Die Unterteilung der virtuellen und der physikalischen Adresse ist in Bild 7 (siehe nächste Seite) dargestellt.

1. Geben Sie die Größe des maximal verfügbaren virtuellen Adressraums in Byte an. In wieviele Segmente wird der virtuelle Adressraum unterteilt? 2 P.
2. Wieviel Seiten können in einem Segment im virtuellen Adressraum gespeichert werden? Geben Sie die Größe einer Seite in Byte an. 2 P.
3. Was ist der Vorteil einer solchen zweistufigen Adressumsetzung gegenüber einer reinen Seitenverwaltung? 2 P.

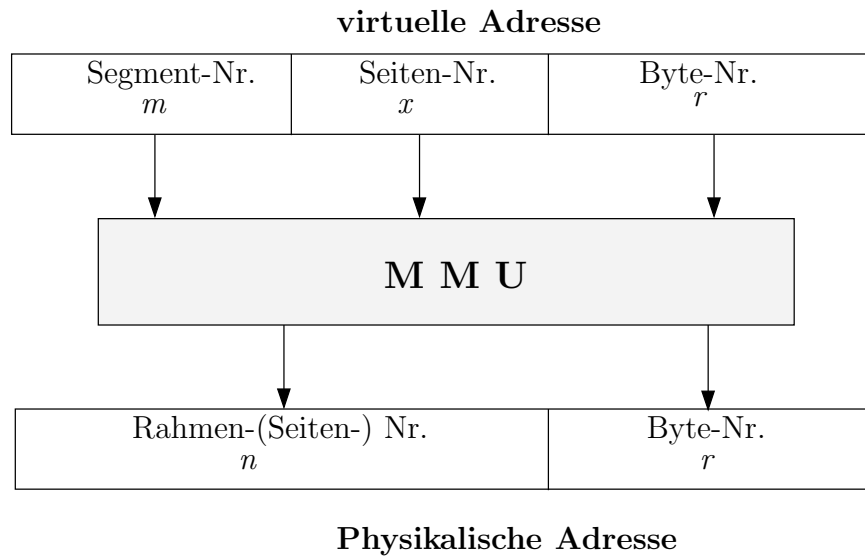


Bild 7: Zweistufige Adressberechnung

Aufgabe 10 *Speicher*

(10 Punkte)

1. Wieviele Adressleitungen sind erforderlich bei einem Speicherbaustein mit einer Kapazität von 4096 Bits und einer 512×8 -Organisation? Begründen Sie Ihre Antwort. 1 P.
2. Wieviele RAM-Bausteine der Organisation $8k \times 1$ sind notwendig, um einen Speicher mit einer Kapazität von 8k Wörter und einer Wortbreite von 16 Bit zu realisieren? Begründen Sie Ihre Antwort. 1 P.
3. Wie ist ein ROM-Baustein mit der Speicherkapazität von 2048 Bits und 7 Adressleitungen organisiert? Begründen Sie Ihre Antwort. 1 P.
4. Welche vier typischen Probleme müssen beim Entwurf einer Speicherhierarchie berücksichtigt werden? Nennen Sie eine Lösungsmöglichkeit zu jedem Problem. 4 P.
5. Vervollständigen Sie das Timing-Diagramm eines DRAM-Bausteins, indem Sie Timing-Parameter, d. h.
 - die Zykluszeit (t_{RC}),
 - die RAS-Zugriffszeit (t_{RAC}),
 - die CAS-Zugriffszeit (t_{CAC}),
 - die RAS-CAS-Delay (t_{RCD}) und
 - die RAS-Precharge-Time (t_{RP}).

eintragen.