



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

## Aufgabenblätter

zur Klausur „Technische Informatik I/II“  
am 02. September 2004, 9.00 - 11.00 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)

Eine unvollständig definierte Schaltfunktion  $y = f(d, c, b, a)$  ist gegeben durch die folgenden Gleichungen:

$$y = \text{MINt}(3, 4, 5, 6, 10, 12, 13, 14)$$

$$y = \text{MAXt}(0, 1, 2, 7, 8, 9)$$

1. Tragen Sie die Schaltfunktion  $f$  in das KV-Diagramm im Lösungsblatt ein. Zeichnen Sie alle Prim-Einsblöcke klar und eindeutig ein. Geben Sie die zugehörigen Primimplikanten an. Unterstreichen Sie alle Kernprimimplikanten. 4 P.
2. Geben Sie eine disjunktive Minimalform (DMF) von  $f$  an. 1 P.

Im Folgenden sei die Überdeckungstabelle einer Schaltfunktion  $z = g(x_n, \dots, x_0)$  mit den Mintermen  $a, b, c, d$  und  $e$  sowie den Primimplikanten  $A, B, C$  und  $D$  gegeben (siehe Tabelle 1).

Primimplikanten	Minterme				
	$a$	$b$	$c$	$d$	$e$
A	×	×			
B	×	×			×
C			×		
D				×	×

Tabelle 1: Überdeckungstabelle der Schaltfunktion  $z = g(x_n, \dots, x_0)$

3. Ist die Schaltfunktion  $z$  vollständig oder unvollständig definiert? Begründen Sie Ihre Antwort. 2 P.
4. Welche Primimplikanten sind Kernprimimplikanten? 1 P.
5. Geben Sie die Überdeckungsfunktion  $\ddot{u}_g$  für die gegebene Überdeckungstabelle an. Formen Sie  $\ddot{u}_g$  in eine disjunktive Form um. 3 P.

## Aufgabe 2 *Schaltnetze* (10 Punkte)

Gegeben sei das in Bild 1 dargestellte Schaltnetz der Schaltfunktion  $y = f(d, c, b, a)$

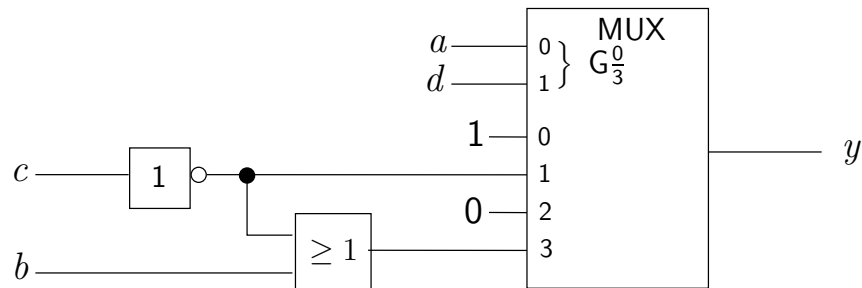


Bild 1: Schaltnetz der Funktion  $y = f(d, c, b, a)$

1. Geben Sie die Schaltfunktion  $y = f(d, c, b, a)$  an. 1 P.
2. Formen Sie die Schaltfunktion  $y$  mit Hilfe der Regeln der Schaltalgebra in eine Minimalform um. 1 P.
3. Die Schaltfunktion  $y$  soll unter ausschließlicher Verwendung von NAND-Gattern realisiert werden. Formen Sie die im letzten Aufgabenteil gefundene Minimalform entsprechend um und zeichnen Sie das zugehörige Schaltnetz. Die Eingangsvariablen stehen nur nicht negiert zur Verfügung. 3 P.
4. Die Schaltfunktion 2 P.

$$g = \text{NAND}_3(x_1, x_2, x_3) = \overline{x_1 \wedge x_2 \wedge x_3}$$

soll unter ausschließlicher Verwendung von NAND-Gattern mit zwei Eingängen realisiert werden. Wandeln Sie die Schaltfunktion entsprechend um. Zeichnen Sie das Schaltbild.

5. Geben Sie eine CMOS-Realisierung des in Bild 2 dargestellten 2:1-Multiplexers an. Es stehen Ihnen dabei zwei Transmission-Gates und ein Inverter zur Verfügung. 3 P.

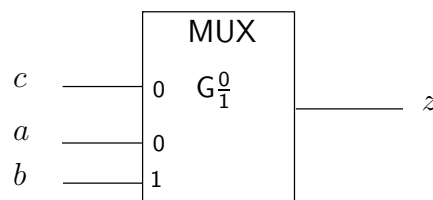


Bild 2: 2:1-Multiplexer

### Aufgabe 3 Laufzeiteffekte

(7 Punkte)

Eine Schaltfunktion  $y = f(d, c, b, a)$  sei durch das Schaltnetz im Bild 3 realisiert.

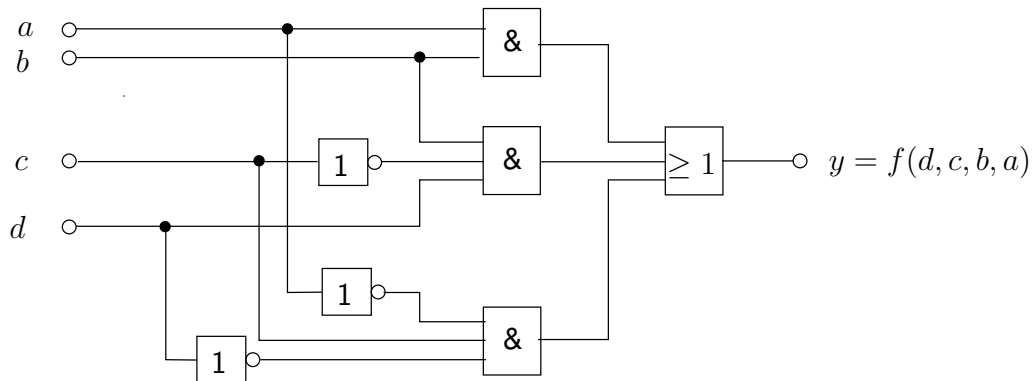


Bild 3: Schaltnetz der Schaltfunktion  $y = f(d, c, b, a)$

1. Geben Sie das endgültige Totzeitmodell des Schaltnetzes an, indem Sie jedem Gatter seinen Verzögerungswert zuweisen und alle Totzeiten zum Eingang des Schaltnetzes verschieben. Geben Sie die Werte der Pfadverzögerungen an. Die verwendeten Gatter NOT, OR und AND besitzen die Totzeiten  $\tau_{NOT}$ ,  $\tau_{OR}$  und  $\tau_{AND}$ . 3 P.
2. Übertragen Sie die Schaltfunktion  $y = f(d, c, b, a)$  in das im Lösungsblatt vorbereitete KV-Diagramm und kennzeichnen Sie die Einsblöcke, die bei der Realisierung durch das Schaltnetz im Bild 3 verwendet wurden. 1 P.
3. Ist das Schaltnetz frei von allen statischen Strukturhasards? Begründen Sie Ihre Antwort. 1 P.
4. Der Übergang  $(1, 1, 0, 1) \rightarrow (0, 1, 1, 1)$  weist einen dynamischen Strukturhasard auf. Geben Sie eine Maßnahme zur Behebung des zugehörigen Hasardfehlers an. Begründen Sie Ihre Antwort. 2 P.

### Aufgabe 4 Schaltwerke

(10 Punkte)

1. In Bild 4 ist ein flankengesteuertes JK-Flipflop mit einem asynchronen „low“-aktiven Rücksetzeingang dargestellt. 1 P.

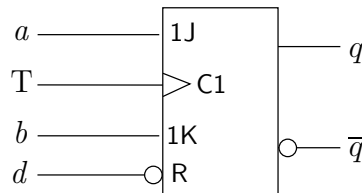


Bild 4: Das JK-Flipflop

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

2. Gegeben ist das in Bild 5 dargestellte Schaltwerk. Das Schaltwerk ist mit flankengesteuerten D-Flipflops realisiert. Es besitzt den Eingang  $x$  und den Ausgang  $y$ . 3 P.

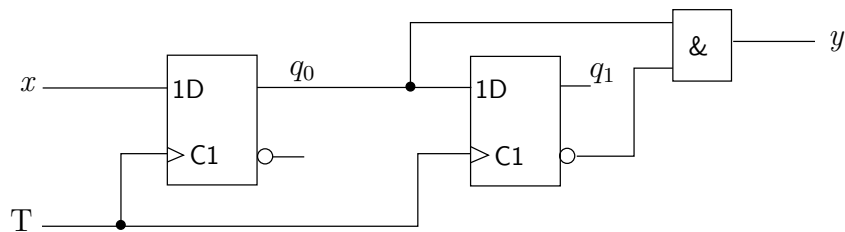


Bild 5: Das Schaltwerk

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

3. Gegeben sei ein synchrones Schaltwerk mit einer Eingangsvariablen  $x^t$  und vier Zuständen, die durch die Zustandsvariablen  $q_1^t, q_0^t$  dual kodiert sind. Die Zustandsübergangsgleichungen sind gegeben durch:

$$q_0^{t+1} = \bar{q}_1^t x^t \vee \bar{q}_0^t \bar{q}_1^t \vee q_0^t q_1^t \bar{x}^t$$

$$q_1^{t+1} = q_1^t x^t \vee \bar{q}_0^t x^t$$

Zur Speicherung der Zustandsvariablen  $q_0$  wird ein T-Flipflop verwendet werden, während  $q_1$  mit einem D-Flipflop realisiert ist.

- (a) Geben Sie die disjunktiven Minimalformen (DMF) der Ansteuerfunktionen der Flipflops an. Geben Sie dabei den Lösungsweg an. 4 P.
- (b) Zeichnen Sie das Schaltwerk. 2 P.

**Aufgabe 5** *Rechnerarithmetik*

(7 Punkte)

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

1. Wandeln Sie  $512_{10}$  in eine 32-Bit-Zweierkomplementzahl um. 1 P.
2. Wandeln Sie  $-1023_{10}$  in eine 32-Bit-Zweierkomplementzahl um. 1 P.
3. Welche Dezimalzahl wird durch das 32-Bit Wort, interpretiert als IEEE-754-Gleitkommazahl, dargestellt? 1 P.

1100 0000 1010 0000 0000 0000 0000 0000

4. Gegeben sei die folgende Darstellung einer Festkommazahl  $N = x_0x_1 \dots x_7$  4 P.

$$N = \sum_{i=0}^7 x_i 2^{-i}$$

Dabei stellt  $x_0$  das Vorzeichenbit und  $x_1 \dots x_n$  den Betrag der Zahl dar. Es sollen zwei Zahlen miteinander multipliziert werden. Hierzu stehen zwei 8-Bit Register M und Q für den Multiplikanden und Multiplikator sowie ein 8-Bit Akkumulator A zur Verfügung.

Berechnen Sie  $1001\ 0101 \times 1100\ 1001$  mit Hilfe der Addiere-Schiebe-Methode. Der Rechenweg soll eindeutig erkennbar sein. Verwenden Sie das im Lösungsblatt angegebene Schema.

## Aufgabe 6 *Allgemeines*

(8 Punkte)

1. Was versteht man unter einem *von-Neumann-Flaschenhals*? 1 P.
2. Das Statusregister enthält Bits (*Flags*), die das Ergebnis einer arithmetischen Operation widerspiegeln. Nennen Sie 4 verschiedene Bits und erklären Sie Ihre Bedeutung. 2 P.
3. Nennen Sie 4 Aufgaben von Schnittstellenbausteinen. 2 P.
4. Skizzieren Sie den Aufbau einer statischen CMOS-Speicherzelle. Aus Ihrer Zeichnung muss die Ankoppelung der Speicherzelle an die Daten- und Adressleitungen erkennbar sein. 3 P.

**Aufgabe 7** MIPS-Assembler

(9 Punkte)

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

6 P.

```
(a)  a = b = c = 0;
      if (i < 5) {
          a = 1;
          b = 2;
          c = 3;
      }
      d = 5;
```

```
(b)  a = b = c = 0;
      if (i < 5) {
          a = 1;
          b = 2;
          c = 3;
      }
      else {
          a = 4;
          b = 5;
          c = 6;
      }
      d = 5;
```

```
(c)  int a[100];
      ...
      sum = 0;
      for (i = 0; i < 100; i++)
          sum = sum + a [i];
```

2. Beschreiben Sie die Funktion der folgenden MIPS-Befehle:

2 P.

- (a) `addu $t3, $t2, $t1`
- (b) `andi $t3, $t2, 0x2000`
- (c) `slt $t3, $t2, $t1`
- (d) `lui $t3, 0x2000`

3. In welchem Register wird die Rücksprungadresse beim Unterprogrammaufruf gesichert?

1 P.

**Aufgabe 8** *Pipelining*

(10 Punkte)

1. Erläutern Sie die Aufgaben der einzelnen Pipeline-Stufen der DLX-Pipeline für bedingte Sprünge mit PC-relativer Adressierung. 2 P.
2. Das folgende MIPS-Programmstück soll auf einem Prozessor mit DLX-Pipeline ausgeführt werden.

```
S1:    lw    $t1, 1000($zero)
S2:    lw    $t2, 1004($zero)
S3:    add   $t3, $t2, $t1
S4:    addi  $t1, $t2, 8
S5:    subi  $t4, $zero, 2
S6:    and   $t5, $t3, $t2
S7:    sw    $t4, 1000($zero)
S8:    sw    $t5, 1004($zero)
S9:    sw    $t1, 1008($zero)
```

- (a) Bestimmen Sie alle Datenabhängigkeiten im Programmstück. 5 P.
- (b) Nehmen Sie an, dass keine *Forwarding*-Techniken implementiert sind und die auftretenden Pipelinekonflikte durch Einfügen von NOP (*No Operation*) Befehlen behoben werden müssen. 3 P.

Ergänzen Sie das obige Programm, so dass es korrekte Ergebnisse liefert. Sie dürfen dabei die Reihenfolge der Befehle **nicht** ändern und so wenig NOP-Befehle wie möglich einfügen.

**Aufgabe 9** *Cache-Speicher*

(11 Punkte)

1. Ein 32-Bit-Prozessor besitzt einen Daten-Cache-Speicher mit einer Speicherkapazität von 128 KByte und einer Blockgröße von 16 Byte. Der Cache-Speicher ist als direkt-abgebildeter Cache (*direct mapped cache*) organisiert. Die Hauptspeicheradresse ist 32 Bit breit. Zur Verwaltung eines Cacheblocks werden zwei zusätzliche Statusbits (*Valid*-Bit: V und *Dirty*-Bit: D) verwendet.

(a) Welche Bits der 32-Bit-Adresse bilden Byte-Offset, Index und Tag? Skizzieren Sie die Unterteilung der Hauptspeicheradresse für diesen Cache-Speicher. 1 P.

(b) Wie viele Zeilen hat der Cache-Speicher? 1 P.

(c) In welche Cache-Zeile wird die Hauptspeicheradresse 2004 geladen? Geben Sie Ihren Rechenweg an. 2 P.

2. Es soll ein 2-fach-satzassoziativer Cache-Speicher (*2-way-set-associative cache*) realisiert werden, der 128 KByte Daten zwischenspeichern kann. Die Blockgröße beträgt 16 Byte. Nehmen Sie an, dass die Hauptspeicheradresse 32 Bit breit ist. Zur Verwaltung eines Cacheblocks werden zwei Statusbits (*Valid*-Bit und *Dirty*-Bit) verwendet.

(a) Wie breit ist der *Tag*? 1 P.

(b) Bestimmen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers. 2 P.

3. Gegeben sei ein Prozessor mit getrennten Befehls- und Datencachespeicher. Die Fehlzugriffsrates des Befehlscachespeichers beträgt 20%, die des Datencachespeichers beträgt 50%. Der Zugriff auf den Befehlscachespeicher erfolgt in 10 ns, der Zugriff auf den Datencachespeicher erfolgt in 20 ns. Ein Hauptspeicherzugriff dauert 80 ns. 4 P.

Auf dem Prozessor läuft ein Programm `test`, dessen Speicherzugriffe zu 80% aus Zugriffen auf Befehle und zu 20% aus Zugriffen auf Daten bestehen. Gehen Sie davon aus, dass die Cachespeicher und der Hauptspeicher gleichzeitig adressiert werden und dass das Datum beim Laden aus dem Hauptspeicher direkt dem Prozessor zugeführt wird.

Berechnen Sie die mittlere Zugriffszeit auf den Speicher für das Programm `test`. Geben Sie Ihren Rechenweg an.

**Aufgabe 10** Virtuelle Speicherverwaltung (7 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU). Der virtuelle Speicher ist in 8 Seiten mit je 1 KByte unterteilt. Der physikalischen Speicher hat eine Kapazität von 4 KByte. Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 2 angegeben.

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

Tabelle 2: 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: 5 P.  
2100, 4095, 5620, 6200, 1023
3. Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen *Translation Lookaside Buffer (TLB)* erreicht? 1 P.