

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation
und

Technische Informatik I/II

am 23. Februar 2018, 11:00 – 13:00 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 $y = f(d, c, b, a)$ sei durch ihre Eins- und *don't care*-Stellen (Abkürzung d) gegeben:

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

1. 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.
2. Tragen Sie alle Primimplikate der Funktion ins KV-Diagramm im Lösungsblatt ein und geben Sie eine konjunktive Minimalform (KMF) der Funktion f an. 3 P.

Gegeben sei eine Schaltfunktion $z = g(d, c, b, a)$, von der man weiß, dass $\bar{c} b \bar{a}$ und $\bar{d} \bar{c} \bar{a}$ *Kernprimimplikanten* dieser Funktion sind.

3. Welche im Lösungsblatt angegebenen Produktterme können definitiv **keine** Primimplikanten der Funktion z sein? Tragen Sie in diesem Fall ein **X** in der Tabelle im Lösungsblatt. Geben Sie zu jedem Produktterm eine Begründung Ihrer Antwort an. (Keine Punkte bei fehlender Begründung) 2 P.

Hinweis: KV-Diagramm verwenden.

Die Funktionen f_1 , f_2 , f_3 und f_4 sollen mit Hilfe eines PLA-Bausteins realisiert werden.

$$\begin{aligned} f_1(c, b, a) &= \text{MINt}(3, 6, 7) \\ f_2(c, b, a) &= \text{MINt}(0, 1, 4, 5, 6) \\ f_3(c, b, a) &= \text{MINt}(2, 3, 4) \\ f_4(c, b, a) &= \text{MINt}(2, 3, 4, 7) \end{aligned}$$

4. Personalisieren Sie den im Lösungsblatt angegebenen PLA-Baustein, indem Sie geeignete Leitungskreuzungen der UND- und der ODER-Matrix markieren. 4 P.

Aufgabe 2 *Spezielle Bausteine*

(8 Punkte)

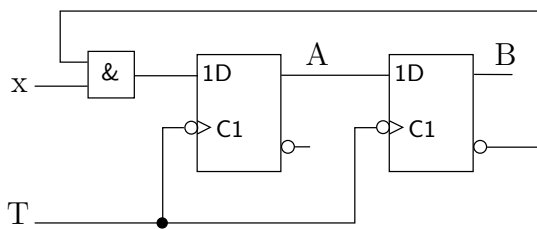
Analysieren Sie die in Abbildung 1 dargestellten Schaltwerke und geben Sie an, ob sie

8 P.

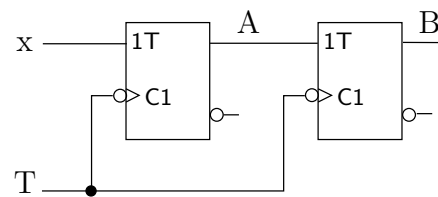
- vorwärts zählen
- rückwärts zählen
- synchron sind
- bei *jedem* Zählerstand mit Hilfe der Eingangsvariablen x angehalten werden können.

Verwenden Sie die im Lösungsblatt vorbereitete Tabelle und tragen Sie ein **X** an der richtigen Stelle ein.

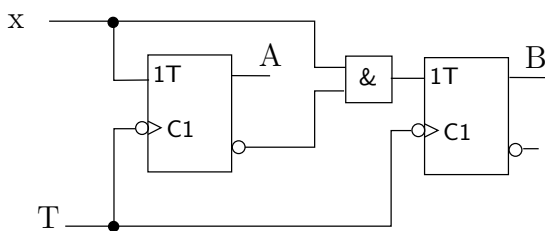
Hinweis: A stellt das niedrigstwertige Bit (LSB) und B das höchstwertige Bit (MSB) dar.



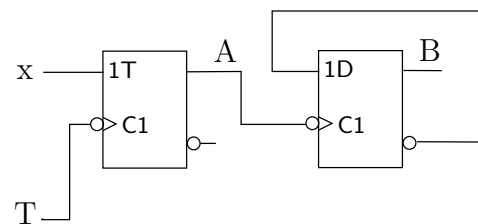
Schaltwerk 1



Schaltwerk 3



Schaltwerk 2



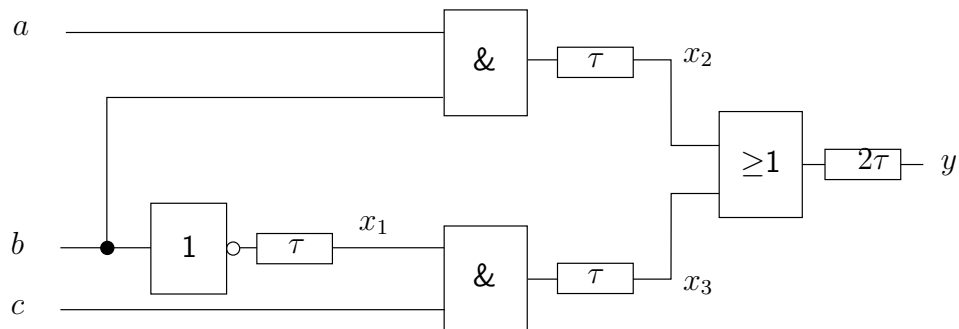
Schaltwerk 4

Abbildung 1: Schaltwerke

Aufgabe 3 Laufzeiteffekte

(7 Punkte)

Eine Schaltfunktion $y = f(c, b, a)$ sei durch das Schaltnetz in Abbildung 2 mit den angegebenen Verzögerungszeiten realisiert. Betrachten Sie den **im Lösungsblatt** angegebenen zeitlichen Verlauf der Eingangsvariablen. Zu Beginn liegen alle Eingabevariablen stabil an.

Abbildung 2: Schaltnetz der Schaltfunktion $y = f(c, b, a)$

1. Zeigen Sie anhand eines Zeitdiagramms, ob die folgenden Eingabewechsel einen Hasardfehler auslösen. Die Variablenreihenfolge sei (c, b, a) . 3 P.
 - (a) b wechselt auf 1, d. h. Übergang $B_5 \rightarrow B_7$ zum Zeitpunkt t_0
 - (b) b wechselt auf 0 zurück, d. h. Übergang $B_7 \rightarrow B_5$ zum Zeitpunkt t_1
2. Falls Sie Hasardfehler im letzten Aufgabenteil gefunden haben, dann geben Sie an, um welchen Typ von Hasardfehlern es sich handelt und wie sie behoben werden könnten. 2 P.
3. Geben Sie einen Übergang an, der mit einem statischen 1-Funktionshasard behaftet ist. Begründen Sie Ihre Antwort. 2 P.

Aufgabe 4 *CMOS*

(10 Punkte)

1. Die Schaltfunktion

5 P.

$$y = (a, b, c) = \bar{a} \vee \bar{b}c$$

soll in der CMOS-Technologie realisiert werden. Es stehen Ihnen zwei NAND-Gatter und ein Inverter-Gatter zur Verfügung. Formen Sie die Schaltfunktion entsprechend um und geben Sie das resultierende Transistor-Schaltbild an.

2. Was ist der Unterschied zwischen einem n-Kanal- und einem p-Kanal-MOSFET? Warum werden pMOS-Transistoren in CMOS-Schaltungen so angeordnet, dass sie eine logische Eins an den Ausgang durchschalten, während nMOS-Transistoren eine logische Null durchschalten?

2 P.

3. Skizzieren Sie den Aufbau eines pMOS-Transistors. Beschriften Sie alle Bestandteile. Aus Ihrer Zeichnung müssen die unterschiedlich dotierten Zonen und die Anschlüsse des Transistors eindeutig erkennbar sein.

3 P.**Aufgabe 5** *Verschiedenes*

(9 Punkte)

1. Was ist der Unterschied zwischen einem Halbaddierer und einem Volladdierer?

1 P.

2. Zeichnen Sie das Schaltnetz eines 1-Bit-Volladdierers. Beschriften Sie die Eingänge und Ausgänge Ihrer Schaltung und geben Sie Ihre Bedeutung an.

3 P.

3. Wie viele Prüfbits sind für eine Einzelbit-Fehlerkorrektur in 20-Bit-Datenwörtern erforderlich? Geben Sie den Rechenweg an.

1 P.

4. Was ist die physikalische Ursache für das Auftreten von Hasardfehlern bei einem Übergang in einem Schaltnetz?

1 P.

5. Was ist der Unterschied zwischen einem PAL-Baustein und einem PLA-Baustein?

1 P.

6. Schieberegister lassen sich in vielfältiger Weise einsetzen. Geben Sie vier unterschiedliche Einsatzmöglichkeiten an.

2 P.

Aufgabe 6 *MIPS-Assembler*

(10 Punkte)

1. Schreiben Sie die folgenden Kontrollstrukturen in MIPS-Assembler um. Die Variablen *i*, *j* und *k* stehen in den Registern *\$s3*, *\$s4* und *\$s5*.

3 P.

(a) **if** (*i*==*j*)
 k = *i*+*j*;

(b) **if** (*i*!=*j*)
 k = *i*+*j*;
 else
 k = *i*-*j*;

(c) **if** (*i* < *j*)
 k = 1;
 else
 k = 0;

2. Bei der MIPS-Architektur können nur 16-Bit Operanden durch MIPS-Befehle in die Register geladen werden. Geben Sie eine Folge von MIPS-Befehlen an, mit welcher der 32-Bit Operand 1111 0000 0011 1101 0000 1001 0000 1001 ins Register *\$s0* geladen werden kann.

2 P.

Hinweis: Sie dürfen den Pseudobefehl *li \$s0, Imm* nicht verwenden.

3. Welche Werte haben die 2 niedrigstwertigen Bits einer Wortadresse?

1 P.

4. Führen Sie den folgenden MIPS-Kode aus und geben Sie die Änderungen in den Register- und Speicherinhalten an. Verwenden Sie die im Lösungsblatt angegebenen Tabellen.

4 P.

| Registersatz | | | Hauptspeicher | | |
|--------------|------------------|--------|---------------|--------|------|
| | Register | Inhalt | Adresse | Inhalt | |
| addi | \$t3, \$0, 0x20 | \$t0 | 0x10 | \$0x20 | 0x22 |
| lw | \$t1, 0x04(\$t3) | \$t1 | 0x14 | \$0x24 | 0x30 |
| add | \$t4, \$t3, \$t0 | \$t2 | 0x16 | \$0x28 | 0x40 |
| sw | \$t4, 0x08(\$t3) | \$t3 | 0x28 | \$0x2C | 0x50 |
| | | \$t4 | 0x1234 | \$0x30 | 0x60 |

Aufgabe 7 *Pipelining*

(10 Punkte)

Das folgende Programmstück soll in einer DLX-Pipeline ausgeführt werden.

```
S1:  lw    $t1, 1000($t0)
S2:  lw    $t2, 1004($t0)
S3:  add   $t3, $t2, $t1
S4:  addi  $t1, $t2, 8
S5:  subi  $t4, $t0, 2
S6:  and   $t5, $t3, $t2
S7:  sw    $t4, 1000($t0)
S8:  sw    $t5, 1004($t0)
S9:  sw    $t1, 1008($t0)
```

1. Bestimmen Sie alle Datenabhängigkeiten im Programmstück. 5 P.
2. 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 nur so wenig NOP-Befehle wie möglich einfügen.

3. Geben Sie die Anzahl der zur Ausführung des Programms notwendigen Taktzyklen an, für 1 P.
 - die sequentielle Ausführung und
 - die korrekte Ausführung in einer DLX-Pipeline ohne Forwarding-Techniken

Nehmen Sie für die sequentielle Ausführung an, dass die Ausführung jedes Befehls fünf Taktzyklen benötigt.

Für die Ausführung auf einer DLX-Pipeline analysieren Sie das modifizierte Programmstück, dass Sie im vorhergehenden Aufgabenteilen erstellt haben.

4. Was sind Struktur- oder Ressourcenkonflikte? Können diese bei der Ausführung des Programmstücks in der DLX-Pipeline auftreten? Begründen Sie Ihre Antwort. 1 P.

Aufgabe 8 *Cache*

(10 Punkte)

Gegeben sind ein direkt-abgebildeter Cache (*direct mapped*, Abkürzung: DM), ein 2-fach satzassoziativer Cache (*2-way-set-associativ*, Abkürzung: A2) und ein vollassoziativer Cache (*fully-associativ*, Abkürzung: AV). Die drei Cache-Speicher haben jeweils eine Speicherkapazität von 32 Byte und werden in Blöcken von je 4 Byte geladen. Die Hauptspeicheradresse ist 32 bit breit. Falls notwendig, wird die »Least Recently Used«-Ersetzungsstrategie verwendet.

Betrachten Sie die Folge der Lesezugriffe auf die folgenden, in hexadezimaler Schreibweise angegebenen Hauptspeicheradressen:

0x0B, 0x2B, 0x07, 0x0C, 0x1E, 0x0A, 0x1A, 0x05, 0x04, 0x29

1. Skizzieren Sie die Unterteilung der Hauptspeicheradresse für die drei Cache-Architekturen. 3 P.
2. Geben Sie die Anzahl der erforderlichen Vergleiche für jede der drei Cache-Architekturen an. 1 P.
3. Nehmen Sie an, die Caches seien zu Beginn leer. Kennzeichnen Sie in der vorbereiteten Tabelle im Lösungsblatt für jeden Cache-Speicher, ob es sich beim Lesezugriff auf die jeweiligen Adressen um einen Treffer (Cache-Hit) oder um keinen Treffer (Cache-Miss) handelt. Verwenden Sie dabei »×« für Cache-Hit und »–« für Cache-Miss. 6 P.

Aufgabe 9 *Speicherverwaltung*

(8 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 2 KByte, 8 virtuellen Seiten und 4 physikalischen Seiten (Frames). Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 1 angegeben.

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

Tabelle 1: 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.

1024, 2047, 2048, 2102, 4095, 4096, 8192, 8202

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.

Aufgabe 10 *Verschiedenes*

(7 Punkte)

1. Aus welchen Komponenten besteht ein einfaches Rechnermodell? 1 P.
2. Beantworten Sie die folgenden Fragen zu Standard-Prozessoren. 2 P.
 - (a) Welches Register enthält den aktuell ausgeführten Befehl?
 - (b) Aus welchem Register entnimmt das Steuerwerk die Information über das Ergebnis einer arithmetisch logischen Operation im Prozessor?
 - (c) Wo steht die Adresse des nächsten auszuführenden Befehls?
 - (d) Welche Einheit des Mikroprozessors führt logische Operationen aus?
3. Beantworten Sie die folgenden Fragen zu RISC-Prozessoren. 2 P.
 - (a) Was macht die Dekodierschaltung in einem RISC-Prozessor einfach?
 - (b) Was bedeutet *Load/Store*-Architektur?
 - (c) Wie ist das Steuerwerk implementiert?
 - (d) Was ist eine *Harvard*-Architektur?
4. Was versteht man bei einem superskalaren Prozessor unter „in-order“ und „out-of-order“? 2 P.