

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation

und

Technische Informatik I/II

am 24. Februar 2017, 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 *Minimierungsverfahren* (9 Punkte)

Eine vollständig definierte Schaltfunktion $y = f(d, c, b, a)$ ist gegeben durch

$$y = \text{MAXt}(0, 4, 8, 9, 10, 12, 13, 14).$$

1. Tragen Sie die Funktion f in das KV-Diagramm im Lösungsblatt ein. Zeichnen Sie *alle* Prim-Nullblöcke klar und eindeutig ein und geben Sie die zugehörigen Primimplikate an. 3 P.
2. Geben Sie *alle* konjunktiven Minimalformen (KMF) von f an. 1 P.
3. Geben die kürzeste Gleichung für f an, die als Ausgangspunkt für die Gewinnung *aller* Primimplikanten mit Hilfe des Nelson-Verfahrens geeignet ist. 1 P.

Eine weitere, vollständig definierte Schaltfunktion $z = g(d, c, b, a)$ ist gegeben durch den Würfel ihrer Einstellen \mathcal{C}_1 . Die Variablenreihenfolge im Würfel ist d, c, b, a .

$$\mathcal{C}_1 = \{(0, 0, -, 1), (0, 0, -, 0), (0, 1, -, 0), (1, 0, 0, -), (1, 1, 0, 0)\}$$

4. Bestimmen Sie mit Hilfe des Consensus-Verfahrens die Menge aller Primimplikanten. Die prinzipielle Vorgehensweise bei der Anwendung des Consensus-Verfahrens soll aus der Lösung ersichtlich sein. Verwenden Sie hierzu die im Lösungsblatt vorbereitete Tabelle. 4 P.

Aufgabe 2 *Laufzeiteffekte* (6 Punkte)

Gegeben ist das in Abbildung 1 dargestellte Schaltnetz. Alle Gatter haben eine Totzeit von 5 ns . Zu Beginn liegen alle Signale stabil an (siehe Zeitdiagramm). Die Eingangssignale D_0 und S wechseln *gleichzeitig* von 0 auf 1. D_1 bleibt aber stabil auf 0. Die Verläufe der Signale a und d sind schon im Zeitdiagramm dargestellt.

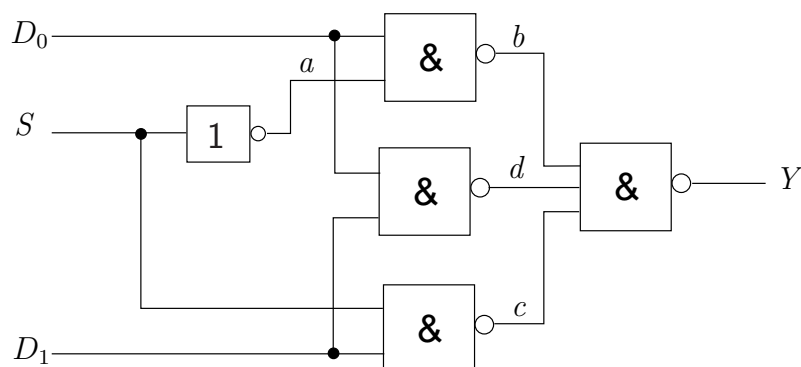


Abbildung 1: Schaltnetz

1. Geben Sie die Verläufe der Signale b, c und Y an, indem Sie das im Lösungsblatt angegebene Zeitdiagramm vervollständigen. 3 P.
2. Der Verlauf des Ausgangssignals Y weist einen Hasardfehler auf. Um welchen Typ von Hasards handelt es sich hierbei? Begründen Sie Ihre Antwort durch eine ausführliche Analyse. 3 P.

Aufgabe 3 *Schaltwerke* (10 Punkte)

1. Gegeben ist das in Abbildung 2 dargestellte Schaltwerk.

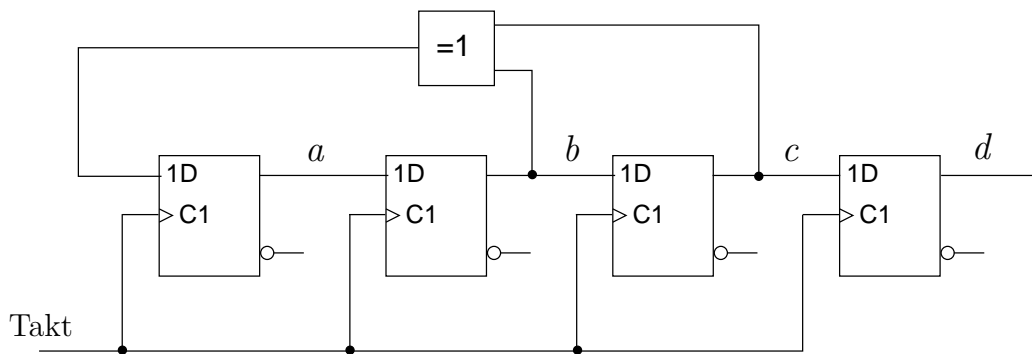


Abbildung 2: Schaltwerk

- (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. 2 P.
2. Es soll ein synchroner modulo-8-Rückwärtszähler mit flankengesteuerten D-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. 2 P.
 - (c) Geben Sie die Ansteuerfunktionen der verwendeten Flipflops in minimaler Form an. 2 P.

Aufgabe 4 *Spezielle Bausteine*

(8 Punkte)

1. Die Schaltfunktion

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

4 P.

soll in der CMOS-Technologie realisiert werden. Es stehen Ihnen ein NOR-Gatter, ein NAND-Gatter, und ein Inverter-Gatter zur Verfügung. Geben Sie das Transistor-Schaltbild an.

2. Für eine Fehlererkennung in einem 4-Bit-Code wird ein Schaltnetz benötigt, welches die anliegenden Eingangsvariablen auf gerade Parität überprüft, d.h die Funktion

4 P.

$$p = \text{even}(w, x, y, z) = w \oplus x \oplus y \oplus z \oplus 1$$

realisiert. Dabei bezeichnet \oplus den Quersummen-Operator.

Realisieren Sie das Schaltnetz unter ausschließlicher Verwendung eines 8:1-Multiplexers und eines Inverters. Zeichnen Sie die Schaltung.

Aufgabe 5 *Rechnerarithmetik & Codes*

(12 Punkte)

1. Addieren Sie die beiden BCD-Zahlen:

2 P.

$$\begin{array}{r} 0011 \ 1000 \\ + \ 0110 \ 0100 \\ \hline \end{array}$$

Der Rechenweg soll klar ersichtlich dargestellt werden.

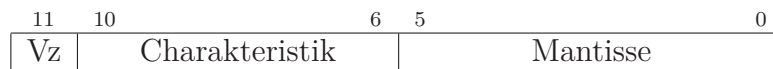
2. Welche Vor- und Nachteile hat die BCD-Arithmetik gegenüber der Dual-Arithmetik?

1 P.

3. Wie viele Prüfbits müssen bei Verwendung des Hamming-Codes mindestens zu einem 32-Bit-Datenwort hinzugefügt werden, um einen Ein-Bit-Fehler korrigieren zu können?

1 P.

4. Gegeben sei das folgende 12-Bit-Fließkommaformat:



Dabei gilt:

- Basis: $b = 2$
- $Vz = 0$ für positive Zahlen, $Vz = 1$ für negative Zahlen.
- Der Exponent Exp ist in Exzess-15-Darstellung (Charakteristik = $Exp + 15$). Die Werte der Charakteristik 0 und 31 dienen zur Darstellung der Null und ∞ .
- Die Mantisse ist, wie beim IEEE-Format, in normalisierter Form mit einer impliziten 1 links vom Komma repräsentiert.

- (a) Stellen Sie die Zahl $-1,375_{10}$ im obigen Format dar. 1 P.
- (b) Geben Sie die kleinste darstellbare normalisierte positive Zahl in diesem Format an. Welchen Dezimalwert hat sie? (Sie brauchen die Zweierpotenzen nicht explizit auszurechnen.) 1 P.
- (c) Geben Sie die kleinste darstellbare normalisierte negative Zahl in diesem Format an. Welchen Dezimalwert hat sie? (Sie brauchen die Zweierpotenzen nicht explizit auszurechnen.) 1 P.
- (d) Wie viele verschiedene Darstellungen der Zahl 0 gibt es in diesem Format? 1 P.
- (e) Addieren und multiplizieren Sie die folgenden, im obigen Format angegebenen Zahlen: 4 P.

$$A = \boxed{0 \mid 1 \ 0 \ 0 \ 0 \ 1 \mid 0 \ 1 \ 1 \ 0 \ 1 \ 1}$$

$$B = \boxed{1 \mid 0 \ 1 \ 1 \ 1 \ 1 \mid 1 \ 0 \ 1 \ 0 \ 1 \ 0}$$

Stellen Sie die Ergebnisse in normalisierter Form dar und tragen Sie sie in die vorbereiteten Felder auf dem Lösungsblatt ein. Geben Sie den Rechenweg an.

Hinweis: Bei einem Überlauf der Mantisse wird nicht gerundet, sondern nur abgeschnitten.

Aufgabe 6 *Allgemeines*

(7 Punkte)

1. Beantworten Sie die folgenden Fragen:

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?

2. 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?

3. Was versteht man unter dem *Y-Diagramm*? Zeichnen Sie das Y-Diagramm von Gajski und beschriften Sie die „Achsen“ und die verschiedenen Entwurfsebenen.

3 P.

Aufgabe 7 *MIPS-Assembler*

(12 Punkte)

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

3 P.

```

andi  $s1, $zero, 40
sra   $s2, $t1, 2
slt   $s3, $zero, $t2
sub   $s4, $t3, $t2
xori  $s5, $t4, -1

```

Die Register seien vor der Ausführung folgendermaßen belegt:

| Register | Inhalt | Register | Inhalt |
|----------|-------------|----------|-------------|
| \$s1 | 0x0000 0000 | \$t1 | 0x0000 0040 |
| \$s2 | 0x0000 0020 | \$t2 | 0x0000 0040 |
| \$s3 | 0xFFFF FFFF | \$t3 | 0x0000 0042 |
| \$s4 | 0x0000 0142 | \$t4 | 0xFFFF FFFB |
| \$s5 | 0x0000 0020 | \$t5 | 0x0000 00B5 |

2. Was ist ein Pseudobefehl? Und was ist eine Assemblerdirektive?

1 P.

3. Wie ist die Trennung von Programmen und Daten bei der Ihnen bekannten MIPS-R2000-Architektur realisiert?

1 P.

4. Gegeben sei das folgende MIPS-Programmstück:

```

        .data
vec:    .word 20, 16, 12, 0, -3, 0, -4, 16, 20, 2014

        .text
main:   lw $t1, vec
        lw $t2, vec+4
        lw $t3, vec($t1)
        lw $t4, vec+4($t1)

```

(a) Geben Sie die Inhalte der Register `$t1`, `$t2`, `$t3` und `$t4` in hexadezimaler Schreibweise nach der Ausführung des obigen Programmcodes an.

2 P.

(b) Geben Sie den MIPS-Code an, mit dem man die Adresse von `vec` im Register `$s0` speichert.

1 P.

(c) Schreiben Sie eine Programmschleife, welche für jedes Element aus `vec` die Summe aller vorhergehenden Elemente ausgibt. Die ausgegebenen Zahlen sollen dabei durch ein Leerzeichen getrennt sein.

4 P.

Hinweis: In Tabelle 1 sind die Nummern und die Argumente der benötigten Systemaufrufe angegeben.

| Funktion | Systemaufruf | Argumente |
|---------------------------|--------------|---|
| <code>print_int</code> | 1 | Integer in <code>\$a0</code> |
| <code>print_string</code> | 4 | Adresse einer mit Null terminierten Zeichenkette in <code>\$a0</code> |

Tabelle 1: Ausgabe-Systemaufrufe (*system calls*)

Aufgabe 8 *Pipelining*

(8 Punkte)

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

4 P.

```

S1:      addi  $t1, $t0, 1
S2:      srl   $t2, $t1, 2
S3:      or    $t3, $t1, $t2
S4:      srl   $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 ein, so dass keine Konflikte auftreten.

4 P.

```

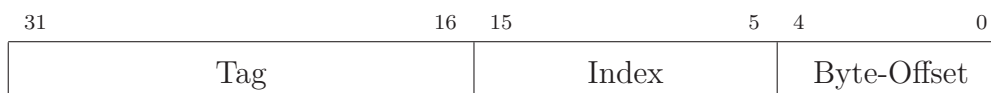
S1:      anfang:  andi $t2, $t1, 1
S2:              beqz $t2, weiter
S3:              subi $t1, $t1, 1
S4:              j   anfang
S5:      weiter:  srl  $t1, $t1, 1
S6:              j   anfang
S7:              addi $t3, $t0, 1

```

Aufgabe 9 *Cache-Speicher*

(10 Punkte)

1. Gegeben sei ein 3-fach-satzassoziativer Cache-Speicher (*3-way-set-associative cache*) mit der folgenden Unterteilung der Hauptspeicheradresse



- (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 `0x00FE13A4` zu. Mit wie vielen und welchen Zeilen im Cache wird ein Vergleich durchgeführt, um herauszufinden, ob ein Cache-Hit vorliegt? 2 P.

4 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 .

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 | D -Bit | V -Bit | Tag |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 1 | 4 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 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:

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

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 *Speicherverwaltung*

(8 Punkte)

Die Speicherverwaltungseinheit in einem Rechnersystem bildet den virtuellen Adressraum der Größe von 4 GByte auf den physikalischen von 32 MByte ab. Die Seitengröße beträgt 4 KByte.

1. Wie viele Bits umfasst die virtuelle Adresse? 1 P.
2. Wie viele Bits umfasst die physikalische Adresse? 1 P.
3. Wie viele Bits umfasst die virtuelle Seitennummer? 1 P.
4. Wie viele Bits umfasst die physikalische Seitennummer? 1 P.

Der Arbeitsspeicher eines Rechners sei bis auf zwei Lücken der Größen 1700 Bytes und 1600 Bytes vollständig belegt (siehe Abbildung 3).



Abbildung 3: Arbeitsspeicher

Die nachfolgenden Speicheranforderungen benötigen Segmente der Größen 1200 Bytes, 1300 Bytes und 450 Bytes.

5. Wenden Sie die *first-fit*-Zuweisungsstrategie (beginnend bei niedrigen Adressen) und die *best-fit*-Zuweisungsstrategie auf diese Folge von Speicheranforderungen an. Welche der beiden Strategien ist günstiger? 2 P.
6. Vergleichen Sie beide Strategien bezüglich ihres Suchaufwands und der Fragmentierung. 2 P.

Lösungsblätter zur Klausur

Digitaltechnik und Entwurfsverfahren & Rechnerorganisation

und

Technische Informatik I/II

am 24. Februar 2017, 14:00 – 16:00 Uhr

| | | |
|-------|----------|-----------------|
| Name: | Vorname: | Matrikelnummer: |
|-------|----------|-----------------|

| Digitaltechnik und Entwurfsverfahren/TI-1 | |
|--|----------------|
| Aufgabe 1 | von 9 Punkten |
| Aufgabe 2 | von 6 Punkten |
| Aufgabe 3 | von 10 Punkten |
| Aufgabe 4 | von 8 Punkten |
| Aufgabe 5 | von 12 Punkten |

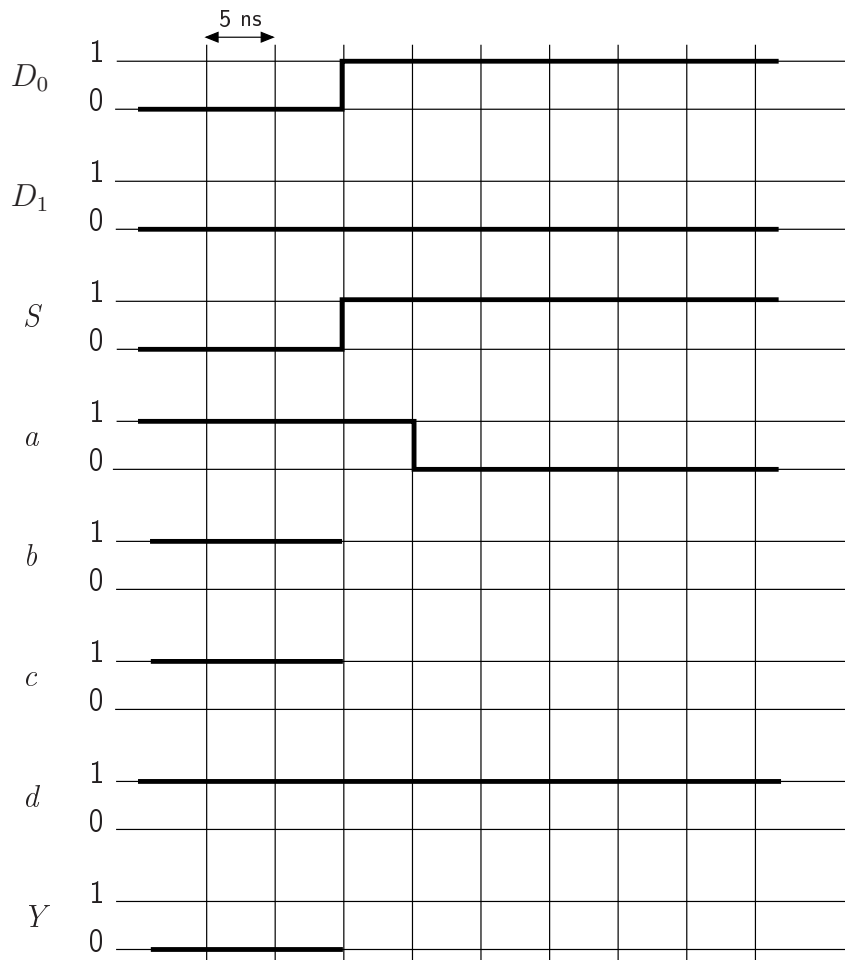
| Rechnerorganisation/TI-2 | |
|---------------------------------|----------------|
| Aufgabe 6 | von 7 Punkten |
| Aufgabe 7 | von 12 Punkten |
| Aufgabe 8 | von 8 Punkten |
| Aufgabe 9 | von 10 Punkten |
| Aufgabe 10 | von 8 Punkten |

| | |
|-------------------------|--|
| Gesamtpunktzahl: | |
|-------------------------|--|

| | |
|--|--------------|
| | Note: |
|--|--------------|

Aufgabe 2

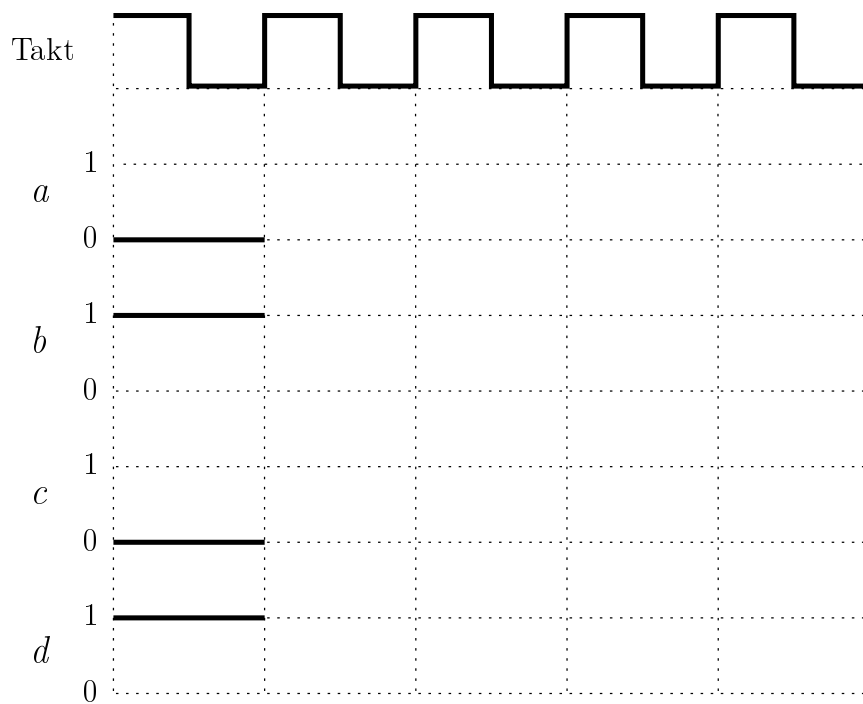
1. Zeitdiagramm:



2. Hasardfehler und Analyse:

Aufgabe 3

1. (a) Das Schaltwerk ist
- (b) Maximale Anzahl der Zustände ist:
- (c) Verläufe der Signale a, b, c und d :



2. (a) Automatengraph:

(b) Kodierte Ablaufabelle:

| q_2^t | q_1^t | q_0^t | q_2^{t+1} | q_1^{t+1} | q_0^{t+1} |
|---------|---------|---------|-------------|-------------|-------------|
| 0 | 0 | 0 | | | |
| 0 | 0 | 1 | | | |
| 0 | 1 | 0 | | | |
| 0 | 1 | 1 | | | |
| 1 | 0 | 0 | | | |
| 1 | 0 | 1 | | | |
| 1 | 1 | 0 | | | |
| 1 | 1 | 1 | | | |

(c) Minimalformen der Ansteuerfunktionen der Flipflops:

Aufgabe 4

1. CMOS-Transistor-Schaltbild von $f(c, b, a)$:

2. Schaltnetz von $p = \text{even}(w, x, y, z)$:

Name:

Vorname:

Matr.-Nr.:

8

4.b) Kleinste darstellbare normalisierte positive Zahl:

| | | |
|--|--|--|
| | | |
|--|--|--|

Dezimalwert:

4.c) Kleinste darstellbare normalisierte negative Zahl:

| | | |
|--|--|--|
| | | |
|--|--|--|

Dezimalwert:

4.d) Verschiedene Darstellungen der Null:

Name:

Vorname:

Matr.-Nr.:

9

4.e)

$$A + B = \boxed{} \boxed{} \boxed{}$$

$$A * B = \boxed{} \boxed{} \boxed{}$$

Aufgabe 6

1. (a)

(b)

(c)

(d)

2. (a)

(b)

(c)

(d)

3. Y-Diagramm:

Aufgabe 7

1. Inhalte der Zielregister:

| Befehl | Zielregister = (z. B. \$s6 = 0x0000 F00A) |
|-------------------------------------|---|
| <code>andi \$s1, \$zero, 40</code> | |
| <code>sra \$s2, \$t1, 2</code> | |
| <code>slt \$s3, \$zero, \$t2</code> | |
| <code>sub \$s4, \$t3, \$t2</code> | |
| <code>xori \$s5, \$t4, -1</code> | |

2. Pseudobefehl:

Assemblerdirektive:

3. Trennung von Programmen und Daten bei der MIPS-R2000-Architektur:

4. (a) Registerinhalte:

| Register | Inhalt |
|----------|--------|
| \$t1 | |
| \$t2 | |
| \$t3 | |
| \$t4 | |

(b) MIPS-Code zur Speicherung der Adresse von `vec` im Register `$s0`:

(c) Programmschleife zur Ausgabe von Partialsummen aus `vec`:

Name:

Vorname:

Matr.-Nr.:

13

Aufgabe 8

1. Datenabhängigkeiten:

2. Beseitigung der Konflikte:

Aufgabe 9

1. (a) Größe eines Cache-Blocks in Byte:

(b) Kapazität des Cache-Speichers:

(c) Der insgesamt erforderliche Speicherbedarf:

(d) Zugriff auf die Adresse 0x00FE13A4:

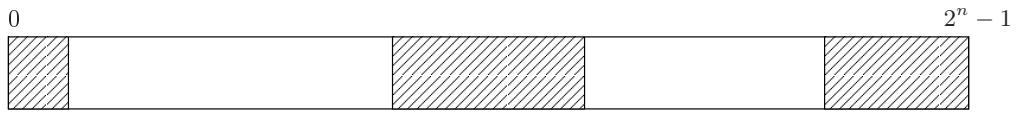
2.

| Adresse | 0x44 | 0xA0 | 0xC3 | 0x9E | 0x66 | 0x2D | 0x6B | 0x49 |
|-------------|------|------|------|------|------|------|------|------|
| read/write | w | w | r | r | r | r | r | r |
| Hit/Miss | | | | | | | | |
| write back? | | | | | | | | |

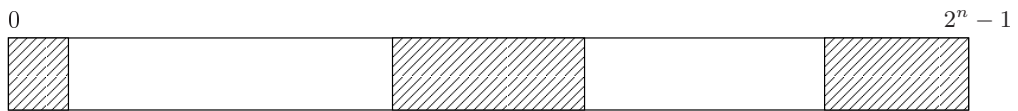
Aufgabe 10

1. Anzahl der Bits der virtuellen Adresse:
2. Anzahl der Bits der physikalischen Adresse:
3. Anzahl der Bits der virtuellen Seitennummer:
4. Anzahl der Bits der physikalischen Seitennummer:

5. *first-fit*-Strategie:



best-fit-Strategie:



6. Vergleich der Strategien: