

# Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 13. August 2024, 08:00 – 10:00 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 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** *Rechnerarithmetik und Codes* (12 Punkte)

In der Vorlesung haben Sie verschiedene Möglichkeiten zur Kodierung von Zahlen- und Zeichendarstellungen kennengelernt. Die Gray-Kodierung benutzt hierfür eine *einschrittige* Kodierung.

1. Was versteht man unter einer einschrittigen Kodierung? 1 P.
2. Geben Sie die Dezimalzahl an, die durch die folgende Gray-Kodierung dargestellt wird:  $1011_G$ . 2 P.
3. Geben Sie die Gray-Kodierung mit 4 Bits der Dezimalzahl 7 an. 2 P.
4. Geben Sie einen Vorteil der Gray-Kodierung an. 1 P.
5. Geben Sie einen Nachteil der Gray-Kodierung an. 1 P.

Für die Darstellung von Gleitkommazahlen wird oft der IEEE-754-Standard verwendet.

6. Wie viele Bits werden als Mantisse bei der Darstellung von Gleitkommazahlen im IEEE-754-Standard für 32-Bit-Zahlen gespeichert? Wie viele für 64-Bit-Zahlen? 1 P.
7. Geben Sie die dezimale Zahl an, die durch die folgende 32-Bit-Darstellung im IEEE-754-Standard dargestellt wird:  $0100\ 0000\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000_{IEEE}$ . 2 P.

In der Vorlesung haben Sie zur Fehlerkorrektur von Datenübertragungen den Hamming-Code kennengelernt.

8. Geben Sie das Codewort des Hamming-Codes für die 4-Bit-Datenfolge 1011 an. 2 P.

**Aufgabe 2** *Schaltfunktion*

(8 Punkte)

Gegeben sei die Schaltfunktion  $h$ :

$$h(d, c, b, a) = a \leftrightarrow b \leftrightarrow c \leftrightarrow d$$

1. Tragen Sie die Schaltfunktion  $h$  in das bereitgestellte KV-Diagramm ein und bestimmen Sie alle Primimplikanten. 5 P.
2. Zeichnen Sie das Schaltnetz, das die Schaltfunktion  $h$  realisiert. Verwenden Sie ausschließlich UND- oder ODER-Gatter mit je 4 Eingängen. Sie können davon ausgehen, dass Eingangsvariablen sowohl negiert als auch nicht negiert vorliegen. Konstante Eingänge stehen überall zur Verfügung. 3 P.

### Aufgabe 3 Bausteine und CMOS-Technologie (6 Punkte)

1. Gegeben sei der in Abbildung 1 dargestellte 4:1-Multiplexer.

4 P.

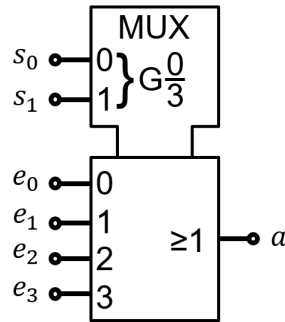


Abbildung 1: 4:1 Multiplexer

Zeichnen Sie eine Realisierung der Schaltfunktion  $a$ , die aus möglichst wenigen 2:1-Multiplexern besteht. Beweisen Sie Ihre Lösung schaltalgebraisch.

2. Welche Schaltfunktion  $z$  wird durch die in Abbildung 2 dargestellte Transistor-Schaltung realisiert? Geben Sie die Schaltfunktion an. Welchem aus der Vorlesung bekannten Bauteil entspricht die Schaltung?

2 P.

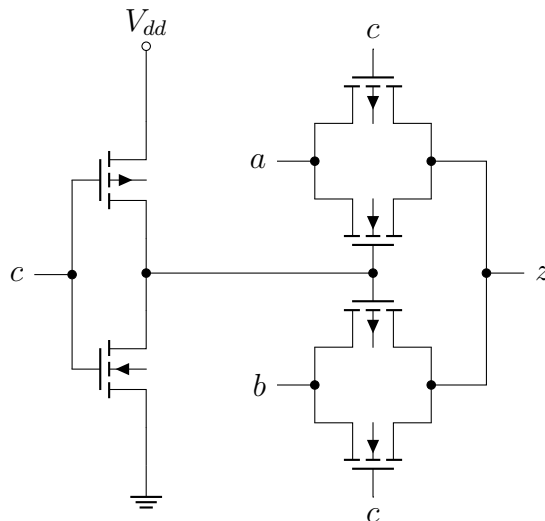


Abbildung 2: Transistor-Schaltung

**Aufgabe 4** Laufzeiteffekte

(10 Punkte)

Gegeben sei das in Abbildung 3 dargestellte Schaltnetz. Die AND- und OR-Gatter haben eine Totzeit von  $4\text{ ns}$ . Das Äquivalenz-Gatter (XNOR) besitzt eine Totzeit von  $5\text{ ns}$ . Die Eingangsvariablen  $a$  und  $b$  wechseln zum Zeitpunkt  $t = 1$  von 0 auf 1. Die Eingangsvariablen  $c$  und  $d$  wechseln zum Zeitpunkt  $t = 4$  von 0 auf 1.

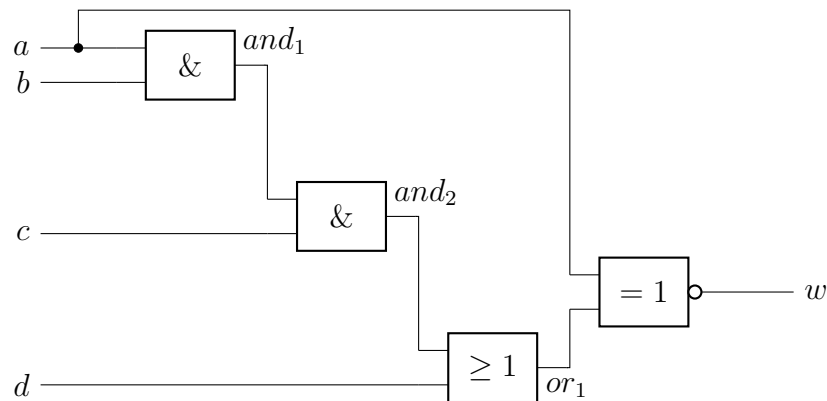


Abbildung 3: Schaltnetz

1. Geben Sie die Verläufe der Signale  $and_1$ ,  $and_2$ ,  $or_1$  und  $w$  an, indem Sie das im Lösungsblatt angegebene Zeitdiagramm vervollständigen. 5 P.
2. Weist der Verlauf von  $w$  Hazardfehler auf? Begründen Sie Ihre Antwort. 1 P.
3. Geben Sie das Totzeitmodell des in Abbildung 3 dargestellten Schaltnetzes an, indem Sie jedem Gatter seinen Verzögerungswert zuweisen und alle Totzeiten zum Eingang des Schaltnetzes verschieben. Geben Sie die Werte der Pfadverzögerungen an. Geben Sie anschließend den Strukturausdruck des Schaltnetzes an. 4 P.

**Aufgabe 5** *Schaltwerke*

(9 Punkte)

Gegeben sei das in Abbildung 4 dargestellte synchrone Schaltwerk mit T-Flipflops. Ziel der Aufgabe ist es, das dargestellte Schaltwerk in ein synchrones Schaltwerk unter Verwendung von D-Flipflops zu überführen.

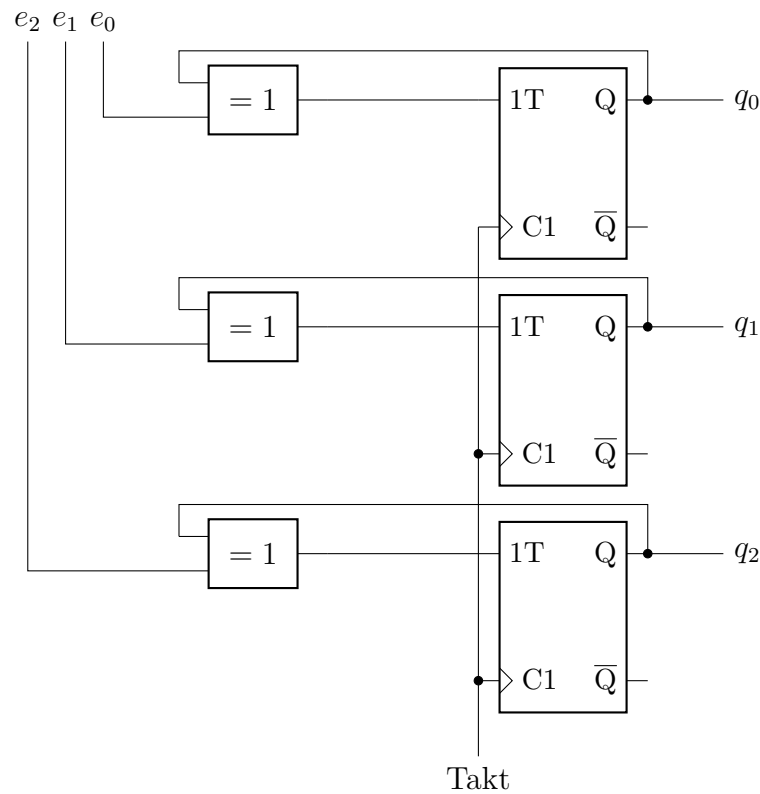


Abbildung 4: Schaltwerk mit T-Flipflops

1. Wie viele Zustände kann das Schaltwerk maximal annehmen? Begründen Sie Ihre Antwort. 1 P.
2. Geben Sie die Ansteuerfunktionen der drei T-Flipflops an. Wie lauten die Zustandsübergangsgleichungen? 3 P.
3. Zeichnen Sie das Schaltwerk unter Verwendung von D-Flipflops. Geben Sie dafür zunächst die Ansteuerfunktionen der D-Flipflops an. 3 P.
4. Entwerfen Sie ein T-Flipflop aus einem JK-Flipflop und möglichst wenigen weiteren Gattern. Zeichnen Sie die resultierende Schaltung. 2 P.

**Aufgabe 6** *RISC-V Assembler*

(7 Punkte)

1. Gegen Sie zu den folgenden Pseudoinstruktionen eine mögliche Umsetzung in RISC-V Assemblersprache an. 3 P.
  - (a) `mv t1, t2`                   # Bewege Registerinhalt von t2 nach t1
  - (b) `neg t1, t2`                   # Negiere Registerinhalt von t2 nach t1
  - (c) `nop`                           # No Operation
  
2. Was bewirken die folgenden Assemblerdirektiven: 2 P.
  - (a) `.data`
  - (b) `.asciz`
  
3. Der Inhalt des Registers `t0` sei ein Integer und soll an den Benutzer ausgegeben werden. Schreiben Sie dazu eine Assemblersequenz (RISC-V), die den Inhalt des Registers `t0` an den Benutzer ausgibt. 2 P.

**Aufgabe 7** *Pipelining*

(9 Punkte)

1. In welche Kategorien lassen sich Datenabhängigkeiten einteilen? 1 P.
2. Bestimmen Sie alle Datenabhängigkeiten im folgenden Programmstück. Geben Sie dabei auch die Art der Abhängigkeit an. 4 P.

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

3. Das folgende Programmstück soll auf einem Prozessor mit der aus der Vorlesung bekannten RISC-V Pipeline mit fünf Stufen ohne Forwarding oder Sprungvorhersage ausgeführt werden. Fügen Sie in das Programmstück möglichst wenige NOP-Befehle ein, sodass keine Konflikte auftreten. Sie können die Marker S1 bis S7 als Abkürzungen der Programmzeile nutzen. 4 P.

```
1      S1:      anfang: andi  t2, t1, 1
2      S2:              beq  t2, zero, weiter
3      S3:              addi t1, t1, -1
4      S4:              j    anfang
5      S5:      weiter: srli  t1, t1, 1
6      S6:              j    anfang
7      S7:              addi t3, t0, 1
```

## Aufgabe 8 Cache-Speicher

(12 Punkte)

1. Bei einem Cache-Speicher mit einer Speicherkapazität von 128 KiByte ist die Hauptspeicheradresse in ein 16 Bit Tag-Feld, ein 11 Bit Index-Feld und ein 5 Bit Byte-Offset unterteilt. Geben Sie bei der Beantwortung der folgenden Fragen den Lösungsweg an.

(a) Bestimmen Sie die Blockgröße in Bytes.

1 P.

(b) Wie viele Einträge besitzt der Cache-Speicher?

1 P.

(c) Wie ist der Cache-Speicher organisiert?

2 P.

2. Es soll ein 5-fach-assoziativer Cache-Speicher (*5-way set associative cache*) mit 4 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.

3 P.

Bestimmen Sie den insgesamt erforderlichen Speicherbedarf zur Realisierung dieses Cache-Speichers.

3. Gegeben sei ein direkt-abgebildeter Cache-Speicher (*direct mapped cache*) mit einer Speicherkapazität von 32 Byte und einer Blockgröße von 8 Byte. Als Aktualisierungsstrategie wird ein Durchschreibverfahren (*write through policy*) mit *write no-allocate* verwendet. Bei dieser Aktualisierungsstrategie wird ein CPU-Datum bei einem *write miss* nur in den Speicher geschrieben. Bei einem *write hit* wird ein CPU-Datum sowohl in den Cache als auch in den Speicher geschrieben.

5 P.

Betrachten Sie die folgenden Lese- und Schreibzugriffe auf die in dezimaler Schreibweise angegebenen Adressen:

| dez. Adresse | 0    | 16 | 48 | 48 | 56 | 8 | 8 | 56 | 32 | 0 |
|--------------|------|----|----|----|----|---|---|----|----|---|
| read/write   | r    | r  | w  | r  | r  | r | r | w  | w  | r |
| dez. Index   | 0    | 2  |    |    |    |   |   |    |    |   |
| dez. Tag     | 0    | 0  |    |    |    |   |   |    |    |   |
| Hit/Miss     | Miss |    |    |    |    |   |   |    |    |   |

Vervollständigen Sie die obige Tabelle im Lösungsblatt. Verwenden Sie dabei **Miss** für Cache-Miss und **Hit** für Cache-Hit.

## Aufgabe 9 Betriebssystemunterstützung (9 Punkte)

1. Nennen Sie zwei verwaltende Aufgaben, die ein Betriebssystem übernimmt. 1 P.
2. Welche zwei Betriebsmodi unterstützt ein Betriebssystem mindestens als Schutzmechanismus? 1 P.
3. In welche zwei Kategorien von Ursachen lassen sich Programmunterbrechungen einteilen? 1 P.

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 2 KiByte, 12 virtuelle Seiten und 8 physikalische Seiten (Frames). Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 1 dargestellt.

| Virtuelle Seite | Physikalische Seite |
|-----------------|---------------------|
| 0               | 3                   |
| 1               | 1                   |
| 2               | -                   |
| 3               | 2                   |
| 4               | 6                   |
| 5               | 0                   |
| 6               | 4                   |
| 7               | -                   |
| 8               | 5                   |
| 9               | -                   |
| 10              | -                   |
| 11              | 7                   |

Tabelle 1: Ausschnitt der Seitentabelle

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

1024, 5666, 12458, 5678, 9004, 8192, 8400, 0

Füllen Sie dazu die auf dem Lösungsblatt bereitgestellte Tabelle aus. Geben Sie die Adressen in dezimaler Schreibweise an. Rechnen Sie dazu das Ergebnis explizit aus.

6. Beschreiben Sie knapp, was bei einem Zugriff auf eine nicht im Hauptspeicher befindliche Seite passiert. 1 P.

## Aufgabe 10 *Allgemeines*

(8 Punkte)

1. Aus welchen Komponenten besteht ein von-Neumann-Rechner? 2 P.
2. 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?
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. Für was steht die Abkürzung *USB*? Um was für eine Form der Verbindungsstruktur handelt es sich? 2 P.

# Lösungsblätter zur Klausur

## Digitaltechnik und Entwurfsverfahren (TI-1)

und

## Rechnerorganisation (TI-2)

am 13. August 2024, 08:00 – 10:00 Uhr

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

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

| <b>Rechnerorganisation (TI-2)</b> |                |
|-----------------------------------|----------------|
| Aufgabe 6                         | von 7 Punkten  |
| Aufgabe 7                         | von 9 Punkten  |
| Aufgabe 8                         | von 12 Punkten |
| Aufgabe 9                         | von 9 Punkten  |
| Aufgabe 10                        | von 8 Punkten  |

| <b>Übungsscheine:</b>                       |               |
|---|---------------|
| Digitaltechnik und Entwurfsverfahren (TI-1) | von 2 Punkten |
| Rechnerorganisation (TI-2)                  | von 2 Punkten |

|                         |  |
|-------------------------|--|
| <b>Gesamtpunktzahl:</b> |  |
|-------------------------|--|

|  |              |
|--|--------------|
|  | <b>Note:</b> |
|--|--------------|

## Aufgabe 1 *Rechnerarithmetik und Codes*

1. Einschrittige Kodierung:

2.  $1011_G$  als Dezimalzahl:

3.  $7_{10}$  in Gray-Kodierung:

4. Vorteil der Gray-Kodierung:

5. Nachteil der Gray-Kodierung:

6. Länge der Mantisse:

- 32-Bit:
- 64-Bit:

Name:

Vorname:

Matr.-Nr.:

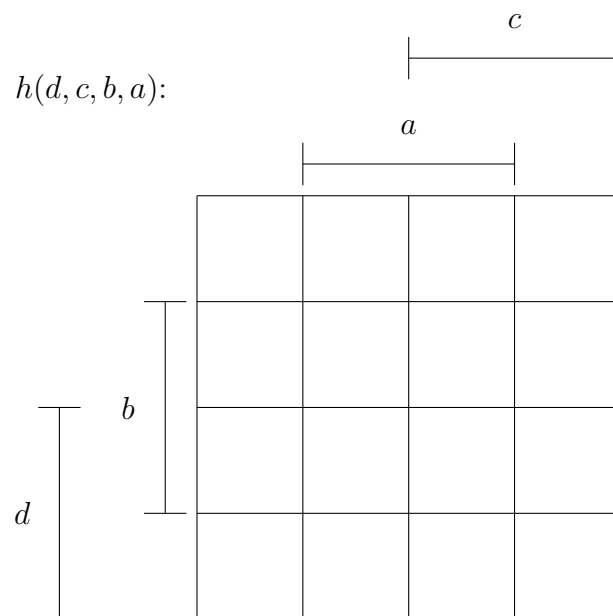
3

7.  $0100\ 0000\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000$ <sub>IEEE</sub> als Dezimalzahl:

8. Hamming-Code für  $1011_2$ :

## Aufgabe 2 *Schaltfunktion*

1. KV-Diagramm:



Primimplikante:

Name:

Vorname:

Matr.-Nr.:

5

2. Schaltnetz:

## Aufgabe 3 *Bausteine und CMOS-Technologie*

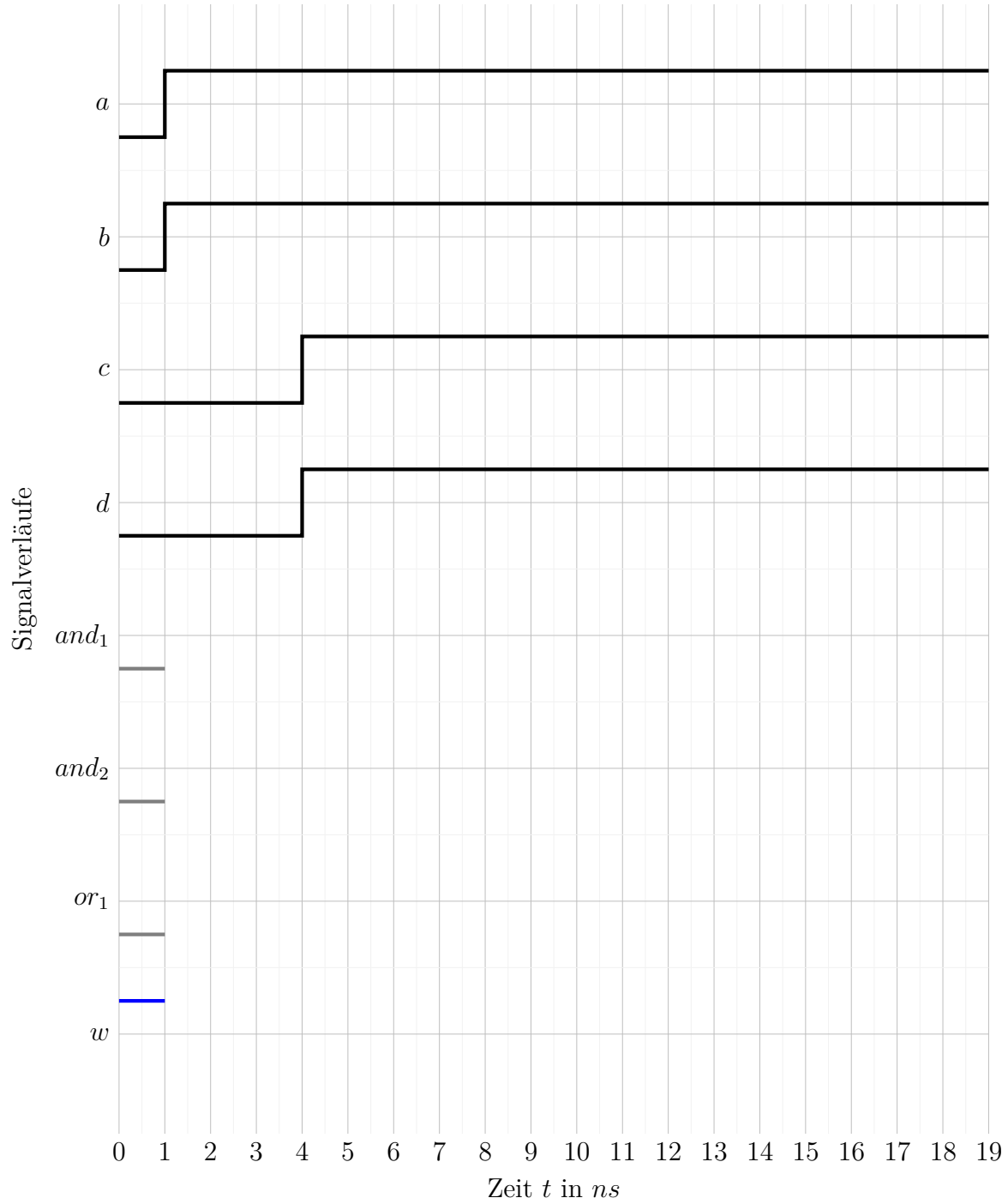
1. Realisierung des 4:1-Multiplexers:

2. • Schaltfunktion  $z$ :

• Bauteil:

# Aufgabe 4 Laufzeiteffekte

1. Laufzeitdiagramm:



2. Hazardfehler:

Name:

Vorname:

Matr.-Nr.:

8

3. • Totzeitmodell:

• Strukturausdruck:

## Aufgabe 5 *Schaltwerke*

1. Anzahl der Zustände mit Begründung:

2. • Ansteuerfunktionen der T-Flipflops:

• Zustandsübergangsgleichungen:

Name:

Vorname:

Matr.-Nr.:

10

3. Schaltwerk mit D-Flipflops:

4. T-Flipflop aus einem JK-Flipflop:

## Aufgabe 6 *RISC-V Assembler*

### 1. Pseudoinstruktionen:

(a) `mv t1, t2`:

(b) `neg t1, t2`:

(c) `nop`

### 2. Assemblerdirektiven:

(a) `.data`:

(b) `.asciz`:

### 3. Assemblersequenz:

Name:

Vorname:

Matr.-Nr.:

12

## Aufgabe 7 *Pipelining*

1. Kategorien der Datenabhängigkeiten:

2. Datenabhängigkeiten:

3. Beseitigung der Konflikte:

## Aufgabe 8 *Cache-Speicher*

1. (a) Blockgröße in Bytes:

(b) Anzahl der Einträge:

(c) Cache-Organisation:

2. Speicherbedarf:



## **Aufgabe 9**    *Betriebssystemunterstützung*

1. Zwei verwaltende Aufgaben:

2. Zwei Betriebsmodi:

3. Zwei Kategorien:

4. Unterteilung der Adresse:

Name:

Vorname:

Matr.-Nr.:

16

5. Physikalische Adressen:

| Virtuelle |              | Physikalische |         |
|-----------|--------------|---------------|---------|
| Adresse   | Seitennummer | Seitennummer  | Adresse |
| 1024      |              |               |         |
| 5666      |              |               |         |
| 12458     |              |               |         |
| 5678      |              |               |         |
| 9004      |              |               |         |
| 8192      |              |               |         |
| 8400      |              |               |         |
| 0         |              |               |         |

6. Zugriff:

## Aufgabe 10 *Allgemeines*

1. Komponenten eines von-Neumann-Rechners:

2. (a)

(b)

(c)

(d)

Name:

Vorname:

Matr.-Nr.:

18

3. (a)

(b)

(c)

(d)

4. *USB*: