

Klausur
„Technische Informatik“
Bachelor Cyber Security & Privacy
Prof. Thomas Breuer

| | | |
|---|----------------------------|-------------|
| Name <small>(bitte in Blockschrift)</small> | <i>Musterlösung</i> | |
| Datum | 16.09.25 | SS 2025 - 2 |

| Multiple Choice | Rechenaufgabe 1 | Rechenaufgabe 2 | Rechenaufgabe 3 | Gesamtpunkte |
|-----------------|-----------------|-----------------|-----------------|--------------|
| | | | | |

Hinweise:

- Bitte beachten Sie die Hinweise auf dem Deckblatt!
- Dieser Klausurteil enthält 11 Seiten.
- Jede Multiple-Choice-Frage hat jeweils nur eine richtige Antwort. Richtige Antworten ergeben je 2 Punkte, bei einer falschen Antwort wird ein Punkt abgezogen. **Nicht beantwortete Fragen werden mit 0 Punkten bewertet.** Es findet kein Übertrag einer negativen Gesamtpunktzahl der Multiple-Choice-Fragen in das Gesamtergebnis statt.
- Bei den Rechenaufgaben muss der Lösungsweg anhand von angegebenen Formeln, Herleitungen, Skizzen oder/und erklärendem Text ersichtlich sein. Kennzeichnen Sie bitte die Endresultate.

Viel Erfolg!

1.) Multiple Choice Fragen (30 Punkte)

| Nr. | Frage | Antwortmöglichkeiten: | | | Antwort A,B oder C ? |
|-----|---|---|--|--|-----------------------------|
| | | A | B | C | |
| 1 | Worauf beruht die 1. Kirchhoffsche Regel (Knotenregel) zur Netzwerkberechnung? | Energieerhaltung | Impulserhaltung | Ladungserhaltung | |
| 2 | Wann ist ein Festkörper elektrisch leitfähig: | Valenzband und Leitungsband überschneiden sich nicht, s.d. Elektronen das Valenzband nicht verlassen können | Valenzband und Leitungsband überschneiden sich, s.d. sich auch im Leitungsband Elektronen befinden | Das Valenzband enthält keine Elektronen, s.d. keine Ladungsträger fest gebunden sind | |
| 3 | Was bewirkt die P-Dotierung eines Halbleitermaterials bei ansonsten konstanter Temperatur? | Eine Abnahme der Leitfähigkeit | Eine geringere Empfindlichkeit gegenüber elektrostatischer Aufladung | Eine Zunahme der Dichte freier Defektelekttronen („Löcher“) | |
| 4 | Warum werden in der Digitaltechnik komplementäre Transistoren (CMOS) eingesetzt? | Mit komplementären Transistoren können verlustarme Schaltungen realisiert werden | Komplementäre Transistoren sind erforderlich, um binäre Rechenoperationen darstellen zu können | Komplementäre Transistoren sind resistenter gegen elektrostatische Aufladungen | |
| 5 | Welche Gleichung ist als „De Morgansches Theorem“ bekannt? | $a + (a \cdot b) = a$ | $\overline{(a \cdot b)} = \bar{a} + \bar{b}$ | $c + \bar{c} \cdot b = c + b$ | |
| 6 | Die Beschaltung eines digitalen Signalausgangs mit mehreren digitalen Eingängen ist ... | ... nur in Bussystem zulässig | ... grundsätzlich in Grenzen immer möglich | ... nur bei speziellen Gattern zulässig | |
| 7 | Die Abkürzung PROM steht für: | Programmable Read Only Memory | Processor Reset Order Memory | Predefined Raw Output Memory | |
| 8 | Wozu dient laut Vorlesung der sogenannte 'Program Counter' (PC)? | Er zählt die ausgeführten Programmschritte um den Rücksprung aus Funktionen zu ermöglichen | Er enthält den nächsten auszuführenden Befehl | Er enthält die Adresse des nächsten auszuführenden Befehls | |
| 9 | Welche Aufgabe hat ein „Stack-Pointer“ ? | Er enthält die Adresse, unter der der nächste Befehl gespeichert ist | Er enthält die Adresse eines als Stack organisierten Speichers, auf die der nächste Schreib-/Lesezugriff erfolgt | Er verwaltet die Rücksprungadresse nach einem Interrupt-Aufruf | |
| 10 | Welche Aussage zum Interrupt-Konzept ist richtig? | Bei einem Ereignis wird eine spezielle Service-Routine aktiviert | Das Interrupt-Konzept berücksichtigt nur prozessorinterne Ereignisse | Das Interrupt-Konzept ist ungeeignet, wenn eine sofortige Reaktion auf ein Ereignis erforderlich ist | |
| 11 | Das Befehlsphasenpipelining moderner Prozessoren arbeitet beispielsweise dann optimal, wenn ... | ... viele Datenabhängigkeiten zwischen den Operanden gegeben sind | ... ein möglichst linearer Programmablauf vorliegt | ... viele bedingte Sprünge im Programmcode auftreten | |

| Nr. | Frage | Antwortmöglichkeiten: | | | Antwort |
|-----|---|--|--|--|--------------|
| | | A | B | C | A,B oder C ? |
| 12 | Welche Eigenschaft eines Prozessors wird mit „Big Endian“ bzw. „Little Endian“ charakterisiert? | Die Reihenfolge, in der die Bytes einer Ganzzahl im Speicher organisiert sind | Die größte positive bzw. kleinste negative Zahl, die von der Verarbeitungseinheit eines Prozessors verarbeitet werden kann | Segmentgröße bei der dynamischen Speicherreservierung | |
| 13 | Welche Aussage zur Codierung alphanumerischer Zeichen ist <u>falsch</u> ? | Der ASCII-Code umfasst 128 Zeichen, weitere 128 sprachspezifische Zeichen sind in speziellen "Codepages" definiert | Der ASCII-Code enthält neben Buchstaben und Zahlen auch Steuerzeichen, z.B. zur Ausgabe eines akustischen Signals | Die "ASCII-Tabelle" ist in heutigen Rechnergenerationen unbrauchbar, da diese vom "Unicode" abgelöst wurde | |
| 14 | Zur Optimierung von Datendurchsatz und Kosten von Rechnersystemen ... | ...verwendet man eine Speicher-Hierarchie | ... verwendet man eine von-Neumann-Architektur | ... verwendet man den Polling-Controller | |
| 15 | Virtueller Speicher wird in einem Rechnersystem nicht verwendet ... | ... zur Trennung der Speicherbereiche der verschiedenen Tasks | ... für die Ablage von 3D-Bilddaten | ... zur Vergrößerung des physikalisch verfügbaren Hauptspeichers | |

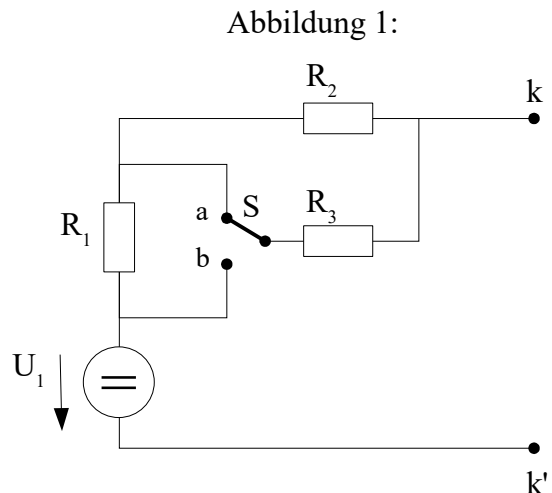
2.) Rechenaufgaben

Aufgabe 1 (20 Punkte)

Gegeben ist eine Schaltung gemäß nebenstehender Abbildung 1 mit der Spannungsquelle $U_1 = 4,5 \text{ V}$, den Widerständen $R_1 = 3,6 \text{ k}\Omega$, $R_2 = 0,6 \text{ k}\Omega$ und $R_3 = 1,2 \text{ k}\Omega$ sowie einem Schalter S .

Der Schalter S befindet sich zunächst in der Position 'a'.

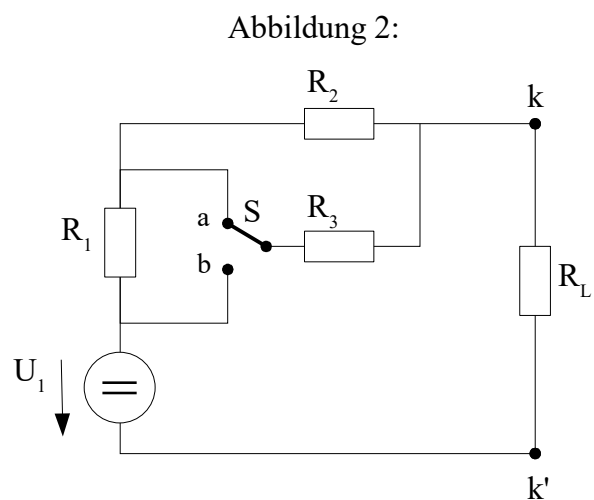
- a) Berechnen Sie den Gesamtwiderstand der Schaltung aus R_1 , R_2 und R_3 !
(5 Punkte)



- b) Welcher Gesamtwiderstand ergibt sich aus R_1 , R_2 und R_3 für die Schalterposition 'b' ?
(5 Punkte)

An die Klemmen k-k' wird nun gemäß Abbildung 2 ein weiterer Widerstand $R_L = 0,6 \text{ k}\Omega$ angeschlossen. Der Schalter befindet sich in der Position 'b'.

- c) Bestimmen Sie den Strom durch Widerstand R_L !
(5 Punkte)

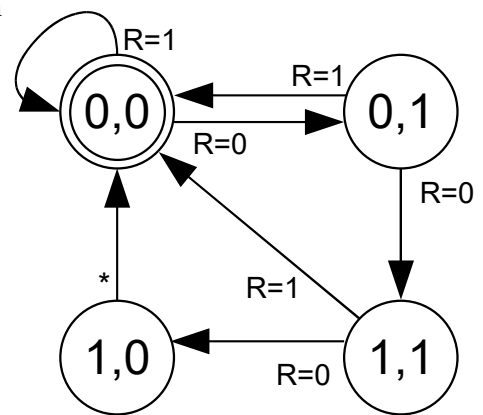


- d) Wie ändert sich der Strom durch Widerstand R_L qualitativ, wenn S wieder auf die Position 'a' umgeschaltet wird?
(5 Punkte)

Aufgabe 2 (20 Punkte)

Die nebenstehende Abbildung zeigt einen Zustandsgraphen mit dem Startzustand (0,0).

Es soll ein Schaltwerk entworfen werden, das diesen Zustandsgraphen realisiert.



*: R=0 oder R=1

- a) Wie viele Flip-Flops sind zur Realisierung erforderlich?
Begründen Sie Ihre Antwort!
(3 Punkte)

- b) Welche besonderen Merkmale weist der Zustandsgraph auf?
(4 Punkte)

- c) Stellen Sie die Zustandsfolgetabelle der Schaltung auf!
(4 Punkte)

| R | $Q_A[n]$ | $Q_B[n]$ | $Q_A[n+1]$ | $Q_B[n+1]$ |
|-----|----------|----------|------------|------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

d) Geben Sie die Zustandsübergangsgleichungen

$$Q_A(n+1) = f(Q_A[n], Q_B[n], R[n])$$

$$Q_B(n+1) = g(Q_A[n], Q_B[n], R[n])$$

in volldisjunktiver Normalform an!

(5 Punkte)

e) Implementieren Sie das Schaltwerk!

(4 Punkte)

Aufgabe 3 (20 Punkte)

Ein gegebener Prozessor verfügt neben dem Program-Counter **PC** über die General-Purpose-Register **R1**, **R2** und **R3**. Die Datenwortbreite beträgt ebenso wie der Adressraum 16-Bit. Die Instruction-Set-Architecture (ISA) umfasst u.a. auch die folgenden Maschinenbefehle:

| | | |
|-----|------------|---|
| NOP | | no operation |
| INC | Rx | inkrementiert Inhalt des Registers Rx |
| DEC | Rx | dekrementiert Inhalt des Registers Rx |
| MOV | Rx, \$addr | kopiert Inhalt der Speicherstelle addr in das Register Rx |
| MOV | Rx, #const | kopiert Konstante const in das Register Rx |
| MOV | \$addr, Rx | kopiert Register Rx an die Speicherstelle addr |
| ADD | Rz, Rx, Ry | addiert Rx und Ry, schreibt Ergebnis nach Rz |
| CMP | Rx, Ry | vergleicht Register Rx mit Ry und setzt ggf. Compare-Flag "less or equal" falls Rx <= Ry |
| JLE | \$addr | Prüft Compare-Flag und springt an die Programm-Adresse addr, falls Flag "less or equal" gesetzt ist |
| JMP | \$addr | Springt unbedingt an die Programm-Adresse addr |

Die in Abbildung 1 gezeigte Codesequenz in der Programmiersprache C wird von einem Compiler in einen Assemblercode übersetzt, dieser ist in der Abbildung 2, Spalte "Instruktion", dargestellt.

Die Variable **x** ist eine vorzeichenlose Ganzzahl mit der Speicheradresse **2000_{hex}**.

Abbildung 1

```
1  x = 0;
2  for(int i=1; i<=2; i++)
3  {
4      x += i;
5  }
```

Abbildung 2

| Programm-Speicher | |
|-------------------|---------------------------------|
| Adresse | Instruktion |
| 8000 | MOV R2, #0 |
| 8004 | <input type="text"/> \$2000, R2 |
| 8008 | MOV R1, #1 |
| 800C | MOV R3, #2 |
| 8010 | JMP <input type="text"/> |
| 8014 | ADD R2, R2, R1 |
| 8018 | MOV \$2000, R2 |
| 801C | INC <input type="text"/> |
| 8020 | CMP R1, R3 |
| 8024 | <input type="text"/> \$8014 |
| 8028 | NOP |

- a) Vervollständigen Sie den Assembler-Code an den grau unterlegten Stellen der Spalte "Instruktion"! (5 Punkte)

Multiple-Choice

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| C | B | C | A | B | B | A | C | B | A | B | A | C | A | B |

Rechenaufgabe 1

a) $R_{ges,a} = R_1 + R_2 \parallel R_3 = (3,6 + \frac{0,6 \cdot 1,2}{0,6+1,2}) k\Omega = 4 k\Omega$

b) $R_{ges,b} = (R_1 + R_2) \parallel R_3 = (\frac{(3,6+0,6) \cdot 1,2}{(3,6+0,6)+1,2}) k\Omega = 0,93 k\Omega$

c) Wegen Schalterposition 'b' liegt R_L in Reihe mit $R_{ges,b}$:

$$I = \frac{U_1}{R_{ges,b} + R_L} = \frac{4,5V}{(0,93+0,6) k\Omega} \approx 3 mA$$

d) Der Gesamtwiderstand wird größer ($R_{ges,a} > R_{ges,b}$), der Strom nimmt daher ab.

Rechenaufgabe 2

a) Benötigte Anzahl n der Flip-Flops muss

$$2^n \geq \text{Anzahl Zustände}$$

erfüllen.

Hier: 4 Zustände im Grafen, also $n = 2$.

b) Unbedingte Zustandsübergänge vorhanden, alle Zustände können erreicht bzw. verlassen werden.

c)

| R | $Q_A [n]$ | $Q_B [n]$ | $Q_A [n+1]$ | $Q_B [n+1]$ |
|-----|-----------|-----------|-------------|-------------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |

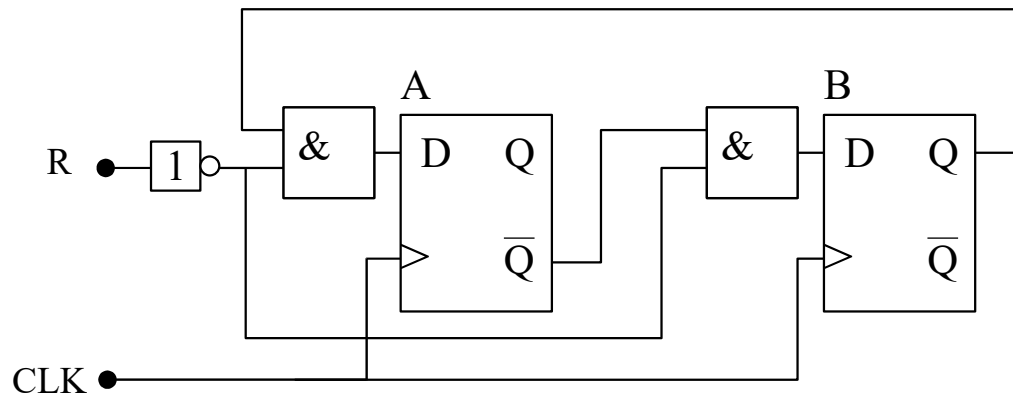
d) $Q_A(n+1) = \bar{R} \cdot \overline{Q_A} \cdot Q_B + \bar{R} \cdot Q_A \cdot Q_B$

$$Q_B(n+1) = \bar{R} \cdot \overline{Q_A} \cdot \overline{Q_B} + \bar{R} \cdot \overline{Q_A} \cdot Q_B$$

e) Zunächst Minimierung:

$$Q_A(n+1) = \bar{R} \cdot Q_B$$

$$Q_B(n+1) = \bar{R} \cdot \bar{Q}_A$$



Rechenaufgabe 2

a)

| Programm-Speicher | |
|-------------------|----------------|
| Adresse | Instruktion |
| 8000 | MOV R2, #0 |
| 8004 | MOV \$2000, R2 |
| 8008 | MOV R1, #1 |
| 800C | MOV R3, #2 |
| 8010 | JMP 8020 |
| 8014 | ADD R2, R2, R1 |
| 8018 | MOV \$2000, R2 |
| 801C | INC R1 |
| 8020 | CMP R1, R3 |
| 8024 | JLE \$8014 |
| 8028 | NOP |

b)

| Register | | | |
|----------|-----|----|-----|
| PC | R1 | R2 | R3 |
| 8004 | --- | 0 | --- |
| 8008 | --- | 0 | --- |
| 800C | 1 | 0 | --- |
| 8010 | 1 | 0 | 2 |
| 8020 | 1 | 0 | 2 |
| 8024 | 1 | 0 | 2 |
| 8014 | 1 | 0 | 2 |
| 8018 | 1 | 1 | 2 |
| 801C | 1 | 1 | 2 |
| 8020 | 2 | 1 | 2 |
| 8024 | 2 | 1 | 2 |
| 8014 | 2 | 1 | 2 |
| 8018 | 2 | 3 | 2 |
| 801C | 2 | 3 | 2 |
| 8020 | 3 | 3 | 2 |
| 8024 | 3 | 3 | 2 |
| 8028 | 3 | 3 | 2 |
| 802C | 3 | 3 | 2 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

x=0

$1 \leq 2 \Rightarrow \text{LE}$

x=1

$2 \leq 2 \Rightarrow \text{LE}$

x=3

$3 > 2 \Rightarrow \text{not LE}$
kein Sprung

c) Zuweisung an Speicheradresse \$2000 (Programmadressen 8004 und 8018) erst nach Verlassen der Schleife (8024).