

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 15. August 2025, 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 Endergebnisse und den 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 & Codes

(6 Punkte)

1. Geben Sie zu den folgenden dezimalen Zahlen die Darstellung im Zweierkomplement mit minimaler Anzahl Bits an.

2 P.

- -1_{10}
- -13_{10}
- 256_{10}
- -17_{10}

2. Geben Sie zu den folgenden Zweierkomplementzahlen die dezimale Darstellung an.

1 P.

- 1111_{ZK}
- 10101_{ZK}

Ein Datenwort soll mithilfe des Hamming-Codes um Prüfbits ergänzt werden, sodass Einzelfehler erkannt und korrigiert werden können.

3. Geben Sie die Ungleichung an, die die Anzahl der Prüfbits k in Abhängigkeit von der Länge des Datenwortes m beschreibt.

1 P.

4. Berechnen Sie die Länge des Codewortes n bei gegebenem Datenwort mit der Länge $m = 11$.

1 P.

5. Erläutern Sie die Bedeutung der in Aufgabe 1.3 genannten Ungleichung. Erläutern Sie unter anderem, was die beiden Seiten der Ungleichung bedeuten.

1 P.

Aufgabe 2 Minimierungsverfahren (12 Punkte)

Eine unvollständig definierte Schaltfunktion $f(x_3, x_2, x_1, x_0)$ ist gegeben durch die Minterme (MINT) und die Don't-Care-Stellen (D):

$$f(x_3, x_2, x_1, x_0) = \text{MINT}(3, 7, 12, 13, 14, 15) \\ \vee \text{D}(0, 1).$$

Es soll das Quine-McCluskey-Verfahren angewandt werden, um disjunktive Minimalformen (DMF) der Funktion $f(x_3, x_2, x_1, x_0)$ zu bestimmen.

1. Erstellen Sie zunächst die 1. Quinesche Tabelle 0. Ordnung, indem Sie die Minterme und Don't-Care Stellen entsprechend ihrer Gewichtung sortieren und in die Tabelle eintragen. 1 P.
2. Ermitteln Sie alle Primimplikanten der Funktion $f(x_3, x_2, x_1, x_0)$ mit Hilfe des Quine-McCluskey-Verfahrens durch Bildung der Tabellen 1. und folgender Ordnungen. 6 P.
3. Stellen Sie die Überdeckungstabelle (2. Quinesche Tabelle) auf. Bestimmen Sie alle Kernprimimplikanten der Funktion $f(x_3, x_2, x_1, x_0)$. 2 P.
4. Ermitteln Sie die konjunktive Form der Überdeckungsfunktion. Überführen Sie zunächst die Überdeckungstabelle in ihre reduzierte Form, indem Sie die Kernprimimplikanten nutzen, um gegebenenfalls Zeilen und Spalten zu streichen. Verzichten Sie auf die Anwendung der Regeln der Spalten- oder Zeilendominanz. 2 P.
5. Geben Sie alle disjunktiven Minimalformen (DMF) der Funktion $f(x_3, x_2, x_1, x_0)$ an. 1 P.

Aufgabe 3 Boolesche Algebra & Schaltnetze (7 Punkte)

1. Zeigen Sie, dass die Operatorenmenge $\{\oplus, \wedge\}$ (XOR, UND) zusammen mit der Konstanten 1 ein vollständiges Operatorensystem bildet. 3 P.

Eine Schaltfunktion $g(c, b, a)$ sei durch das Schaltnetz in Abbildung 1 gegeben.

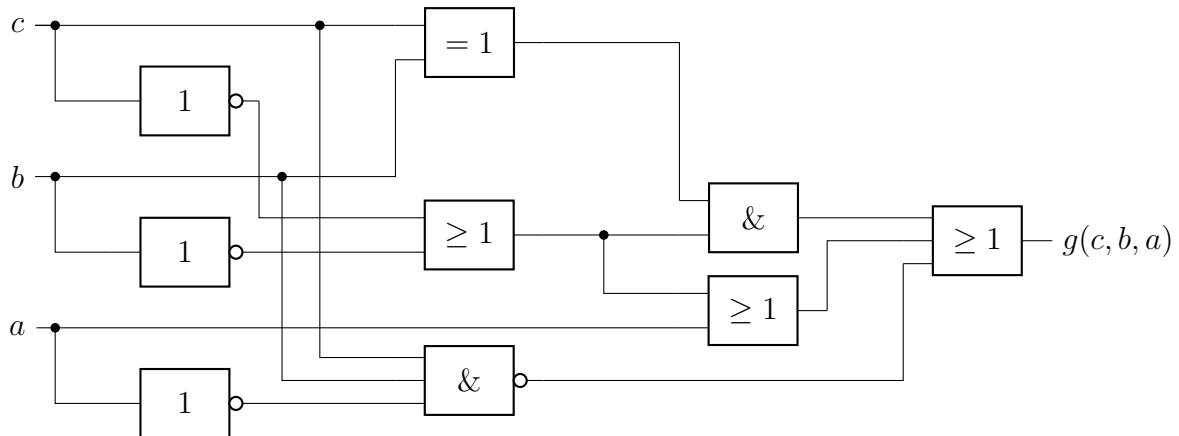


Abbildung 1: Schaltnetz der Funktion $g(c, b, a)$

2. Geben Sie zu dem in Abbildung 1 dargestellten Schaltnetz die Schaltfunktion $g(c, b, a)$ an. 1 P.
3. Geben Sie die zur Schaltfunktion $g(c, b, a)$ gehörende Wahrheitstabelle an. 2 P.
4. Geben Sie eine konjunktive Minimalform (KMF) der Schaltfunktion $g(c, b, a)$ an. 1 P.

Aufgabe 4 Laufzeiteffekte (6 Punkte)

Gegeben ist das in Abbildung 2 dargestellte Schaltnetz. Die beiden Gatter haben jeweils eine Laufzeit von 1 ns . Der Verlauf des Eingangssignals E ist in Abbildung 3 dargestellt.

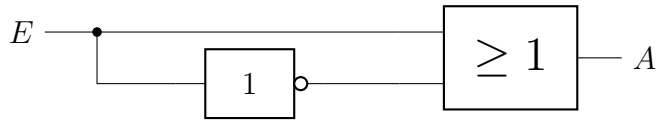


Abbildung 2: Schaltnetz

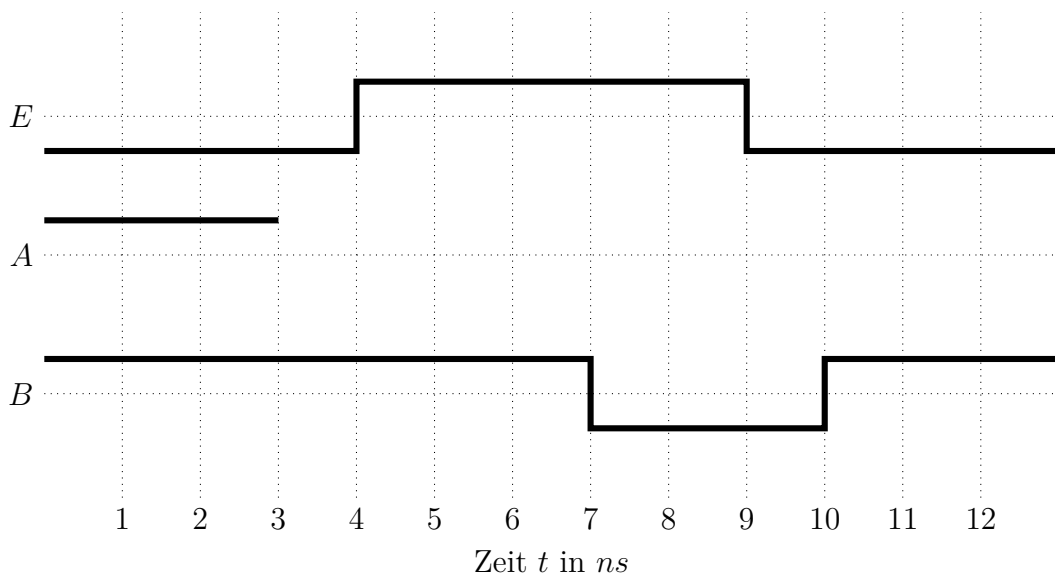


Abbildung 3: Verlauf der Signale A , B und E

1. Vervollständigen Sie auf dem Lösungsblatt den Verlauf des Ausgangssignals A im Zeitintervall zwischen 3 ns und 12 ns . 2 P.
2. Analysieren Sie den Verlauf des Ausgangssignals A auf Hazardfehler. Falls ein Hazardfehler auftritt, wodurch ist dieser Fehler bedingt? Begründen Sie Ihre Antwort. 2 P.
3. Zeichnen Sie ein Schaltnetz, das am Ausgang das Signal B erzeugt, wenn das Signal E am Eingang anliegt. Als Gatter stehen Ihnen zwei Inverter und ein NAND-Gatter mit zwei Eingängen zur Verfügung. Die Gatter haben jeweils eine Laufzeit von 1 ns . 2 P.

Aufgabe 5 Schaltwerke

(14 Punkte)

Es soll ein Mealy-Schaltwerk entworfen werden, welches eine beliebig lange Bitfolge einliest (Variable e) und eine Sequenz von drei oder mehr Einsen erkennt. Das Schaltwerk soll am Ausgang a eine 1 ausgeben, wenn eine solche Sequenz auftritt. Sie können davon ausgehen, dass das Schaltwerk sich zu Beginn im Startzustand z_0 befindet.

1. Wie viele Zustände sind mindestens erforderlich? Geben Sie den Automatengraphen des Schaltwerkes an. 4 P.

2. Das Schaltwerk soll mit flankengesteuerten D-Flipflops realisiert werden. Geben Sie die kodierte Ablaufabelle des Schaltwerkes an. Die Zustände sind dual mit den Zustandsvariablen q_n, q_{n-1}, \dots, q_0 kodiert. 2 P.

3. Ermitteln Sie mit Hilfe der im Lösungsblatt angegebenen KV-Diagramme die disjunktiven Minimalformen 5 P.
 - (a) der Zustandsübergangsgleichungen $z^{t+1} = \delta_i(z^t, e^t)$,
 - (b) der Ansteuerfunktionen der Flipflops $D_i^t = f_i(z^t, e^t)$ und
 - (c) der Ausgangsgleichung $a^t = \omega(z^t, e^t)$.

4. Zeichnen Sie die resultierende Schaltung des Schaltwerkes. 3 P.

Aufgabe 6 RISC-V-Architektur (10 Punkte)

Gegeben ist ein vereinfachter Datenpfad eines RISC-V Prozessors, der nur die folgenden Befehlen unterstützt:

- `add rd, rs1, rs2`
- `lw rd, offset(rs1)`
- `sw rs2, offset(rs1)`
- `beq rs1, rs2, offset`

Die zentrale Steuereinheit erzeugt die Steuersignale basierend auf dem Opcode des Befehls.

1. Skizzieren Sie den vereinfachten Datenpfad, der folgende Komponenten enthalten muss: 5 P.

- PC (Program Counter)
- Befehlsspeicher (Instruction Memory)
- Registerspeicher (Register File)
- ALU (Arithmetic Logic Unit)
- Datenspeicher (Data Memory)
- Multiplexer (MUX) (falls notwendig)
- Vorzeichen-Erweiterung (Immediate Generation)
- Steuerleitungen zu den Komponenten (Control Signals)

Beschriften Sie alle relevanten Signale und Datenwege.

2. Erklären Sie die Rolle folgender Steuersignale im Datenpfad und nennen Sie jeweils die Befehle bei denen sie relevant sind: 3 P.

- RegWrite
- PCSrc
- ALUSrc

3. Beschreiben Sie den Ablauf der Instruktion `lw rd, offset(rs1)` im Datenpfad. Gehen Sie dabei auf die beteiligten Komponenten, Signale und den Datenfluss ein. 2 P.

Aufgabe 7 RISC-V Assembler

(6 Punkte)

Betrachten Sie den folgenden RISC-V Assemblercode:

```
1  .data
2  array: .word 3, 7, -2, 5, -1, 9
3
4  .text
5  la t0, array      # t0 = Adresse des ersten Elements
6  li t1, 6          # t1 = Anzahl der Elemente
7  li t2, 0          # t2 = Laufindex
8  li t3, 0          # ???
9
10 loop:
11  beq t2, t1, end   # Abbruchbedingung
12
13  lw t4, 0(t0)      # Lade Wort von 0(t0) in t4
14  bgez t4, addit    # Wenn t4 >= 0, zu addit
15  j skip
16
17 addit:
18  add t3, t3, t4    # t3 += t4
19
20 skip:
21  addi t0, t0, 4     # t0 += 4
22  addi t2, t2, 1     # t2 += 1
23  j loop
24
25 end:
26  add a0, t3, zero  # Ergebnis in a0
27
28
```

1. Beschreiben Sie eigenen Worten, was obiges Programm berechnet. 2 P.
2. Was steht nach Ausführung des Programms in dezimaler Schreibweise in Register a0? 1 P.
3. Erläutern Sie den Kontrollfluss des Programms. Denken Sie an die Anzahl der Schleifendurchläufe und was in jedem Schleifendurchlauf passiert. 2 P.
4. Geben Sie die Anzahl der Speicherzugriffe an, die das Programm benötigt. Zählen Sie dabei nur die Zugriffe auf den Datenspeicher (nicht auf den Befehlsspeicher). 1 P.

Aufgabe 8 Pipelining (9 Punkte)

Ein idealisierter RISC-Prozessor verfügt über eine 5-stufige Pipeline (IF, ID, EX, MEM, WB), die voll ausgelastet betrieben werden kann. Es werden keine Stalls, keine Branches und keine Hazards betrachtet (idealer Fall).

Ein Programm besteht aus 100 Instruktionen.

1. Erklären Sie den Begriff Durchsatz (Throughput) in Bezug auf eine Pipeline. Wie verhält sich der Durchsatz im Vergleich zu einem nicht-pipelined Prozessor? 1 P.
2. Wie viele Taktzyklen werden benötigt, um die 100 Instruktionen auszuführen? Begründen Sie Ihre Antwort anhand des Pipelining-Prinzips. 1 P.
3. Berechnen Sie die mittlere Zyklusanzahl pro Instruktion (CPI) in diesem Szenario. 1 P.
4. Wie viele Taktzyklen würde ein äquivalenter nicht-pipelined Prozessor mit derselben 5-stufigen Ausführungseinheit benötigen? 1 P.
5. Berechnen Sie den Speedup der Pipeline gegenüber dem nicht-pipelined Prozessor. 1 P.

Betrachten Sie nun den folgenden RISC-V Assemblercode, der in der Ihnen aus der Vorlesung bekannten Pipeline ausgeführt wird:

```
1 S1: add x5, x1, x2
2 S2: sub x6, x5, x3
3 S3: and x7, x6, x4
4 S4: or x8, x7, x5
5 S5: add x5, x8, x9
6 S6: xor x10, x5, x1
7
```

6. Erklären Sie den Begriff echte Datenabhängigkeit. Warum ist diese Art von Abhängigkeit kritisch für Pipelines? 1 P.
7. Bestimmen Sie alle echten Datenabhängigkeiten des obigen Programmstücks. Sie können die Marken S1 bis S6 als Abkürzung des Befehls nutzen. 3 P.

Aufgabe 9 Cache-Speicher

(12 Punkte)

Caches sind für Prozessoren wichtig zur Bereitstellung einer leistungsfähigen Speicherhierarchie. Nachfolgend finden Sie eine Liste von Speicheradressen auf die sequentiell lesend zugegriffen werden soll. Es soll hierbei eine Byte-Adressierung verwendet werden.

0x03, 0xb4, 0x2b, 0x02, 0xbf, 0x58, 0xbe, 0x0e, 0xf5,
0x2c, 0xba, 0xfd

1. Geben Sie für jeden dieser Speicherzugriffe den Tag, den Zeilenindex und den Byteoffset in hexadezimaler Schreibweise an. Gehen Sie dabei von einem direkt-abgebildeten Cache (*direct-mapped cache*) mit **32 Speicherblöcken** aus, bestehend aus jeweils **zwei Byte**. Listen Sie ebenfalls auf, ob es sich dabei um einen Hit (H) oder Miss (M) beim Speicherzugriff handelt. Der Cache sei dabei initial leer. Verwenden Sie die auf dem Lösungsblatt bereit gestellte Tabelle. 6 P.
2. Geben Sie für jeden dieser Speicherzugriffe den Tag, den Zeilenindex und den Byteoffset in hexadezimaler Schreibweise an. Gehen Sie dabei von einem direkt-abgebildeten Cache (*direct-mapped cache*) mit **16 Speicherblöcken** aus, bestehend aus jeweils **vier Bytes**. Listen Sie ebenfalls auf, ob es sich dabei um einen Hit (H) oder Miss (M) beim Speicherzugriff handelt. Der Cache sei dabei initial leer. Verwenden Sie die auf dem Lösungsblatt bereit gestellte Tabelle. 6 P.

Aufgabe 10 Speicherverwaltung (8 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 512 Byte, 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	7
2	3
3	-
4	9
5	6
6	5
7	1

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.

1023, 1024, 1998, 2049

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 KiByte aus. 2 P.

Lösungsblätter zur Klausur

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 15. August 2025, 11:00 – 13:00 Uhr

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

Digitaltechnik und Entwurfsverfahren (TI-1)	
Aufgabe 1	von 6 Punkten
Aufgabe 2	von 12 Punkten
Aufgabe 3	von 7 Punkten
Aufgabe 4	von 6 Punkten
Aufgabe 5	von 14 Punkten

Rechnerorganisation (TI-2)	
Aufgabe 6	von 10 Punkten
Aufgabe 7	von 6 Punkten
Aufgabe 8	von 9 Punkten
Aufgabe 9	von 12 Punkten
Aufgabe 10	von 8 Punkten

Übungsscheine:	
Digitaltechnik und Entwurfsverfahren (TI-1)	von 2 Punkten
Rechnerorganisation (TI-2)	von 2 Punkten

Gesamtpunktzahl:	
-------------------------	--

	Note:
--	--------------

Aufgabe 1 Rechnerarithmetik

1. Darstellung im Zweierkomplement:

- -1_{10} :

- -13_{10} :

- 256_{10} :

- -17_{10} :

2. Dezimale Darstellung:

- 1111_{ZK} :

- 10101_{ZK} :

Name:

Vorname:

Matr.-Nr.:

3

3. Ungleichung:

4. Länge des Codewortes:

5. Erläuterung der Ungleichung:

Aufgabe 2 Minimierungsverfahren

1. + 2. 1. Quinesche Tabelle:

#	Nr.	0. Ordnung	#	Nr.	Ordnung	#	Nr.	Ordnung

#	Nr.	Ordnung	#	Nr.	Ordnung	#	Nr.	Ordnung

Primimplikanten:

Name:

Vorname:

Matr.-Nr.:

5

3. 2. Quinesche Tabelle:

Primimplikant	Minterme												

Die Kernprimimplikanten sind:

4.

Reduzierte Tabelle:

Primimplikant	Minterme												

Überdeckungsfunktion:

5. Disjunktiven Minimalformen (DMF):

Name:

Vorname:

Matr.-Nr.:

7

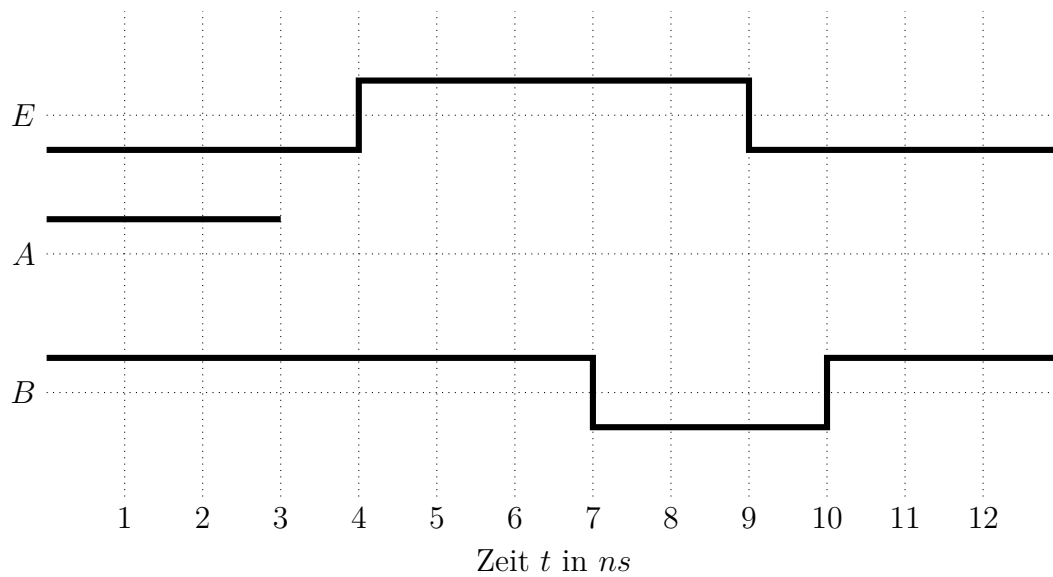
Aufgabe 3 Boolesche Algebra & Schaltnetze

1. Vollständiges Operatorensystem:

2. Schaltfunktion $g(c, b, a)$:

Aufgabe 4 Laufzeiteffekte

1. Verlauf des Signals A :



2. Analyse auf Hazardfehler:

3. Schaltnetz für Signal B :

Name:

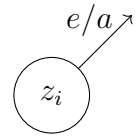
Vorname:

Matr.-Nr.:

10

Aufgabe 5 Schaltwerke

1. Automatengraph:

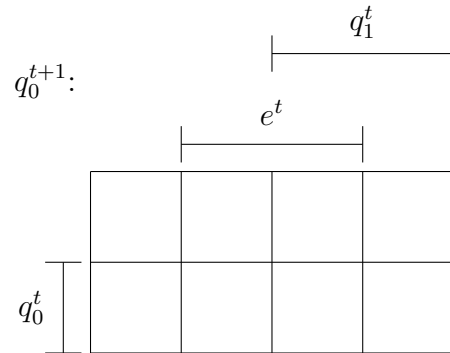
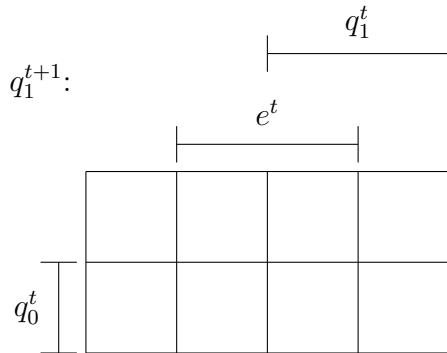


Anzahl der Zustände:

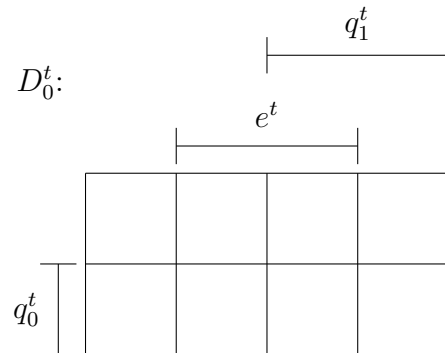
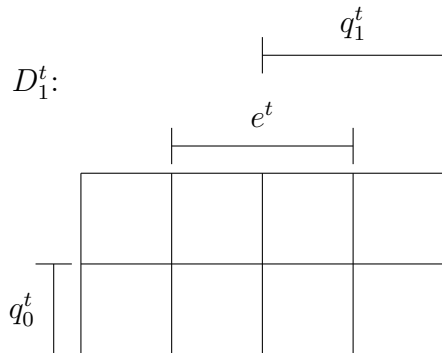
2. Kodierte Ablauftabelle:

Zustand	Eingabe	Folgezustand	FF-Ansteuersignale	Ausgabe

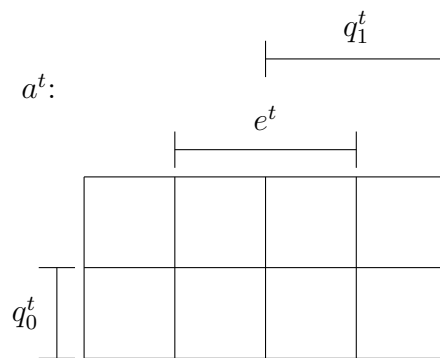
3. (a) DMF der Zustandsübergangsgleichungen:



- (b) DMF der Ansteuerfunktionen der Flipflops:



(c) DMF der Ausgabegleichung:



4. Schaltung des Schaltwerkes:

Name:

Vorname:

Matr.-Nr.:

13

Aufgabe 6 RISC-V-Architektur

1. Vereinfachter Datenpfad:

Name:

Vorname:

Matr.-Nr.:

14

2. Rolle der Steuerleitungen:

- RegWrite:

- PCSrc:

- ALUSrc:

3. Ablauf von *lw*:

Name:

Vorname:

Matr.-Nr.:

15

Aufgabe 7 RISC-V Assembler

1. Verhalten des Programms:

2. Register *a0*:

3. Kontrollfluss:

4. Anzahl Speicherzugriffe:

Name:

Vorname:

Matr.-Nr.:

16

Aufgabe 8 Pipelining

1. Durchsatz:

2. Taktzyklen:

3. CPI:

4. Ohne Pipeline:

Name:

Vorname:

Matr.-Nr.:

17

5. Speedup:

6. Echte Datenabhängigkeit (Erklärung):

7. Echte Datenabhängigkeiten:

Aufgabe 9 Cache-Speicher

1. Direkt-abgebildeter Cache mit 32 Speicherblöcken:

Adresse	Hilfsspalte	Tag	Index	Offset	Hit/Miss
0x03					
0xb4					
0x2b					
0x02					
0xbf					
0x58					
0xbe					
0x0e					
0xf5					
0x2c					
0xba					
0xfd					

2. Direkt abgebildeter Cache mit 16 Speicherblöcken:

Adresse	Hilfsspalte	Tag	Index	Offset	Hit/Miss
0x03					
0xb4					
0x2b					
0x02					
0xbf					
0x58					
0xbe					
0x0e					
0xf5					
0x2c					
0xba					
0xfd					

Aufgabe 10 Speicherverwaltung

1. Unterteilung der virtuellen Adresse:

2. Physikalische Adressen:

Virtuelle		Physikalische	
Adresse	Seitennummer	Seitennummer	Adresse
1023			
1024			
1998			
2049			

3. Bedingungen für Beschleunigung:

4. Breite des Tags: