

Klausur
„Technische Informatik“
Bachelor Cyber Security & Privacy

Name <small>(bitte in Blockschrift)</small>	<i>Musterlösung</i>	
Datum	19.09.2023	SS 2023-2

Multiple Choice	Rechenaufgabe 1	Rechenaufgabe 2	Rechenaufgabe 3	Gesamtpunkte	Vorleistung (Praktikum)

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 (40 Punkte)

Nr.	Frage	Antwortmöglichkeiten:			Antwort A,B oder C ?
		A	B	C	
1	Welche Eigenschaft sollte ein Gerät zur Messung der elektrischen Stromstärke (Amperemeter) haben?	Der Innenwiderstand R_i sollte möglichst gering sein	Der Innenwiderstand sollte möglichst genau $R_i = 1 \text{ M}\Omega$ betragen	Der Innenwiderstand R_i sollte möglichst groß sein	
2	Der Stromfluss in einem metallischen Leiter, wird bewirkt durch ...	Elektronen	Ionen	Elektronen und Defektelektronen	
3	Worauf beruht die 1. Kirchhoffsche Regel (Knotenregel) zur Netzwerkberechnung?	Energieerhaltung	Ladungserhaltung	Impulserhaltung	
4	Warum werden in der Digitaltechnik komplementäre Transistoren 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	Die Angabe der „Propagation Time“ in Datenblättern digitaler Bausteine bezeichnet die mittlere Signallaufzeit	... die minimal zulässige Periodendauer des Taktsignals	... die mittlere Betriebsdauer, nach der statistisch ein Bauteil ausfallen kann	
6	Die logische XOR-Verknüpfung wird auch bezeichnet als ...	JK-Flip-Flop	Äquivalenz-Verknüpfung	Antivalenz-Verknüpfung	
7	Welche Aussage zu einer logischen Funktion $f(a,b,c)$ in vollkonjunktiver Normalform ist falsch ?	Die Funktion enthält maximal 8 unterschiedliche Maxterme	Die Funktion lässt sich stets in eine disjunktive Normalform überführen	Die Funktion kann nicht minimiert werden	
8	Welche Schaltung lässt sich prinzipiell nicht mit einem Schaltnetz realisieren?	Addierer	Zähler	Komperator	
9	Welche Flip-Flops können in einem synchronen Schaltwerk mit Rückkopplung eingesetzt werden?	Taktflankengesteuerte Master-Slave-Flip-Flops	Taktzustandsgesteuerte RS-Flip-Flops	Nur D-Flip-Flops	
10	Eine Multiplikation zweier positiver Ganzzahlen lässt sich dann mit einer Shift-Operation implementieren, wenn einer der Faktoren eine gerade Zahl ist	... wenn beide Faktoren eine Zweierpotenz sind	... wenn einer der Faktoren eine Zweierpotenz ist	
11	Worin besteht die Besonderheit des Gray-Codes?	Die Codierung wird für binär codierte Dezimalzahlen verwendet	Beim Wechsel zwischen benachbarten Zahlen ändert sich genau ein Bit	Die Codierung ist besonders für arithmetische Operationen geeignet	
12	Welche Aussage zur Codierung alphanumerischer Zeichen ist falsch ?	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	
13	Ein "atomarer Befehl" eines Prozessors kann nur von Quantencomputern ausgeführt werden	... ist Bestandteil des Nano-Codes	... wird auch durch einen Interrupt <u>nicht</u> unterbrochen	

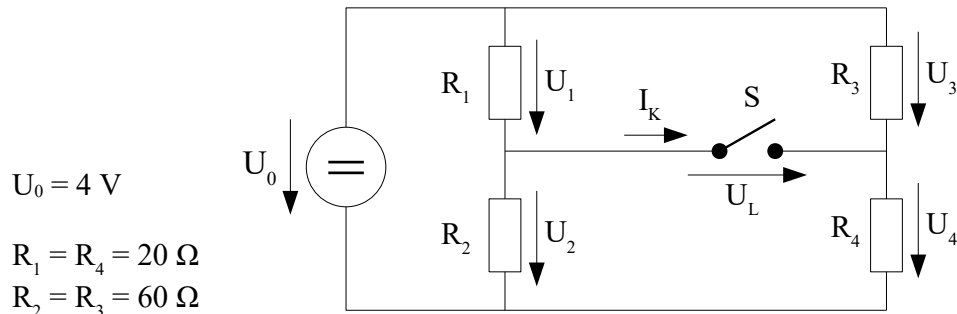
Nr.	Frage	Antwortmöglichkeiten:			Antwort
		A	B	C	A,B oder C ?
14	Für welche Software-Struktur ist ein „Stack“ erforderlich?	Bedingte Verzweigung mit if-Anweisung	Programmschleife mit for-Anweisung	Funktionsaufruf	
15	Welche Aussage zu einer „direkten Adressierung“ ist richtig?	Die Adresse der Speicherstelle ist bereits zur Compilezeit bekannt	Die Adresse der Speicherstelle ergibt sich direkt aus dem Inhalt des Program-Counters	Die Adresse des Operanden ist fest vorgegeben, beispielsweise bei Stack-Operationen	
16	Die „Instruction Set Architecture“ (ISA) eines Prozessors beinhaltet neben Verküpfungs- und Verzweigungsoperationen auch Transferoperationen mit Festplattenzugriffen	... definiert den vollständigen Befehlssatz eines Prozessors	... definiert die in einer Hochsprache verwendbaren Befehle und Funktionen	
17	Welche Aufgabe hat die „Memory Management Unit“ (MMU)	Übersetzung einer virtuellen Adresse in eine physikalische Adresse	Organisation der Speicherstruktur eines Memory-Sticks	Verwaltung des Level-3-Caches	
18	Als Harvard-Architektur wird eine Rechnerarchitektur bezeichnet, die sich durch eine Befehlsphasen-Pipeline und wenige orthogonale Befehle auszeichnet	... bei der die Speicheraufteilung für Programm-Code und Daten je nach Anwendung frei gewählt werden kann	... die für Programm-Code und Daten getrennte Speicherbereiche und Datenpfade aufweist	
19	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	
20	Die Einheit „Whetstone“ wird verwendet für die Effizienz des Level-1-Caches	... für synthetische Benchmarks von Floating-Point-Rechenleistungen	... für die maximale Speicherzugriffsrate des Hauptspeichers	

2.) Rechenaufgaben

Aufgabe 1 (20 Punkte)

Gemäß folgender Abbildung 1 ist eine Schaltung mit der Spannungsquelle U_0 , den Widerständen R_1 , R_2 , R_3 und R_4 sowie einem Schalter S gegeben.

Abbildung 1:



Der Schalter S ist zunächst **geöffnet**.

a) Geben Sie die Spannungen U_1 bis U_4 an!

(5 Punkte)

b) Berechnen Sie die Spannung U_L !

(5 Punkte)

Der Schalter S wird nun **geschlossen**.

c) Geben Sie erneut die Spannung U_1 bis U_4 an!

(5 Punkte)

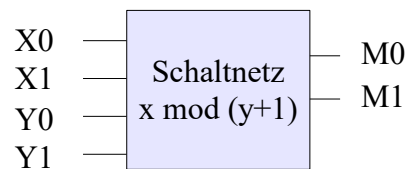
d) Berechnen Sie den Strom I_K !

(5 Punkte)

Aufgabe 2 (20 Punkte)

Es soll ein Schaltnetz entworfen werden, das eine mathematische Modulo-Operation (Rest der ganzzahligen Division) durchführt (Abbildung 2). Die beiden Operanden x und y sowie das Ergebnis m werden als 2 Bit Dualzahl kodiert. Da die Division durch 0 nicht definiert ist, wird der um 1 erhöhte Eingangswert als Divisor verwendet.

Abbildung 2:



a) Vervollständigen Sie die Wahrheitstabelle des Schaltnetzes!

(3 Punkte)

x	$X1$	$X0$	$y+1$	$Y1$	$Y0$	$x \bmod (y+1)$	$M1$	$M0$
0	0	0	1	0	0	0	0	0
0	0	0	2	0	1			
0	0	0	3	1	0			
0	0	0	4	1	1			
1	0	1	1	0	0	0	0	0
1	0	1	2	0	1	1	0	1
1	0	1	3	1	0	1	0	1
1	0	1	4	1	1	1	0	1
2	1	0	1	0	0	0	0	0
2	1	0	2	0	1	0	0	0
2	1	0	3	1	0	2	1	0
2	1	0	4	1	1			
3	1	1	1	0	0	0	0	0
3	1	1	2	0	1			
3	1	1	3	1	0	0	0	0
3	1	1	4	1	1	3	1	1

b) Geben Sie die Schaltfunktionen $M1 = f(X1, X0, Y1, Y0)$ und $M0 = g(X1, X0, Y1, Y0)$ jeweils in volldisjunktiver Normalform an!

(5 Punkte)

c) Minimieren Sie die beiden Funktion ***M1*** und ***M0*** !
(6 Punkte)

d) Implementieren Sie das gesamte Schaltnetz!
(6 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 "not equal"
JNE	addr	Prüft Compare-Flag und springt an die Programm-Adresse addr, falls Flag "not equal" gesetzt ist
JMP	addr	Springt unbedingt an die Programm-Adresse addr

Die in Abbildung 3a gezeigte Codesequenz in der Programmiersprache C wird von einem Compiler in einen Assemblercode übersetzt, dieser ist in der Abbildung 3b, Spalte "Instruktion", dargestellt.

Abbildung 3a

```
1  int b = 3;
2  int x = 7;
3
4  if( b == 1)
5  {
6      x = x - 1;
7  }
8  else
9  {
10     x = x + 2;
11 }
```

Abbildung 3b

Programm-Speicher	
Adresse	Instruktion
8000	MOV R1, #3
8004	<input type="text"/> R2, #7
8008	<input type="text"/> R3, #1
800C	CMP <input type="text"/>
8010	JNE <input type="text"/>
8014	<input type="text"/> R2
8018	JMP <input type="text"/>
801C	MOV R3, #2
8020	ADD <input type="text"/> , R2
8024	NOP

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

- b) Ermitteln Sie die Inhalte der Register **PC**, **R1**, **R2** und **R3**, die sich jeweils nach der Ausführung der Instruktionen ergeben.
(10 Punkte)

Register			
PC	R1	R2	R3
8004	0003	0000	0000

Multiple-Choice

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A	B	A	A	C	C	B	A	C	B	C	C	C	A	B	A	C	B	B

Rechenaufgabe 1

- a) S offen: R_1 und R_2 sowie R_3 und R_4 bilden je eine Reihenschaltung, die parallel an derselben Spannungsquelle U_0 liegen. Mit der Spannungsteilerregel gilt:

$$U_1 = \frac{R_1}{R_1 + R_2} \cdot U_0 = \frac{20}{80} \cdot 4V = 1V$$

$$U_2 = \frac{R_2}{R_1 + R_2} \cdot U_0 = \frac{60}{80} \cdot 4V = 3V$$

$$U_3 = \frac{R_3}{R_3 + R_4} \cdot U_0 = \frac{60}{80} \cdot 4V = 3V$$

$$U_4 = \frac{R_4}{R_3 + R_4} \cdot U_0 = \frac{20}{80} \cdot 4V = 1V$$

Anmerkung: Die Lösung kann vereinfacht werden, da Aufgrund paarweise gleicher Widerstandswerte

$$U_3 = U_2 = 3V \quad \text{und} \quad U_4 = U_1 = 1V$$

gilt.

- b) Maschenumlauf:

$$U_L + U_4 - U_2 = 0 \quad \Rightarrow \quad U_L = U_2 - U_4 = 3V - 1V = 2V$$

- c) S geschlossen: R_1 und R_3 sowie R_2 und R_4 bilden je eine Parallelschaltung, die wiederum in Reihe geschaltet sind. Insbesondere gilt:

$$R_{13} = \frac{R_1 \cdot R_3}{R_1 + R_3} = 15\Omega \quad \text{und} \quad R_{24} = \frac{R_2 \cdot R_4}{R_2 + R_4} = 15\Omega$$

Wieder mit Spannungsteilerregel:

$$U_1 = U_3 = \frac{R_{13}}{R_{13} + R_{24}} \cdot U_0 = 2V \quad \text{und} \quad U_2 = U_4 = \frac{R_{24}}{R_{13} + R_{24}} \cdot U_0 = 2V$$

Anmerkung: Die Lösung kann vereinfacht werden, da Aufgrund der gegebenen Widerstandswerte $R_{13} = R_{24}$ gilt, also:

$$U_1 = U_3 = U_2 = U_4 = \frac{1}{2} \cdot U_0 = 2V$$

- d) Knotengleichung für den Knoten zwischen R_1 und R_2 :

$$I_1 - I_k - I_2 = 0 \quad \text{mit} \quad I_1 = \frac{U_1}{R_1} = \frac{2V}{20\Omega} = \frac{1}{10}A \quad \text{und} \quad I_2 = \frac{U_2}{R_2} = \frac{1}{30}A$$

$$I_k = \left(\frac{1}{10} - \frac{1}{30} \right) A = \frac{1}{15}A$$

Rechenaufgabe 2

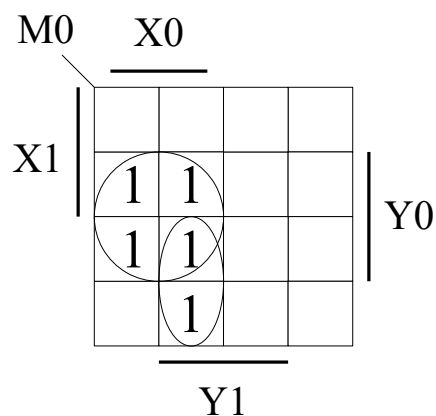
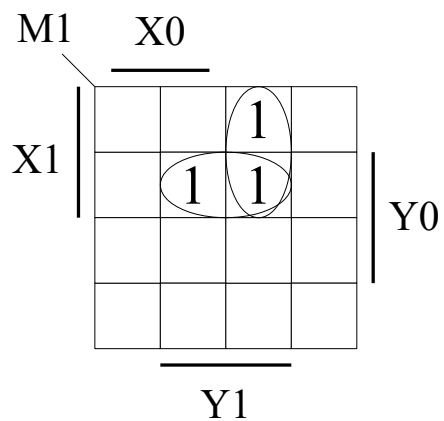
a)

x	$X1$	$X0$	$y+1$	$Y1$	$Y0$	$x \bmod (y+1)$	$M1$	$M0$
0	0	0	1	0	0	0	0	0
0	0	0	2	0	1	0	0	0
0	0	0	3	1	0	0	0	0
0	0	0	4	1	1	0	0	0
1	0	1	1	0	0	0	0	0
1	0	1	2	0	1	1	0	1
1	0	1	3	1	0	1	0	1
1	0	1	4	1	1	1	0	1
2	1	0	1	0	0	0	0	0
2	1	0	2	0	1	0	0	0
2	1	0	3	1	0	2	1	0
2	1	0	4	1	1	2	1	0
3	1	1	1	0	0	0	0	0
3	1	1	2	0	1	1	0	1
3	1	1	3	1	0	0	0	0
3	1	1	4	1	1	3	1	1

b) $M1 = X1 \cdot \bar{X}0 \cdot Y1 \cdot \bar{Y}0 + X1 \cdot \bar{X}0 \cdot Y1 \cdot Y0 + X1 \cdot X0 \cdot Y1 \cdot Y0$

$$M0 = \bar{X}1 \cdot X0 \cdot \bar{Y}1 \cdot Y0 + \bar{X}1 \cdot X0 \cdot Y1 \cdot \bar{Y}0 + \bar{X}1 \cdot X0 \cdot Y1 \cdot Y0 + X1 \cdot X0 \cdot \bar{Y}1 \cdot Y0 + X1 \cdot X0 \cdot Y1 \cdot Y0$$

c)



$$M1 = X1 \cdot \bar{X}0 \cdot Y1 + X1 \cdot Y0 \cdot Y1$$

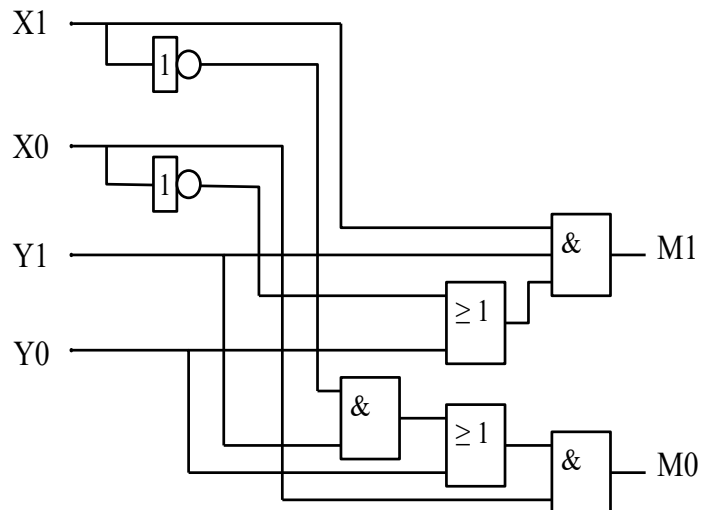
$$M0 = X0 \cdot Y0 + \bar{X}1 \cdot X0 \cdot Y1$$

weitere Vereinfachung möglich:

$$M1 = X1 \cdot Y1 \cdot (\bar{X}0 + Y0)$$

$$M0 = X0 \cdot (Y0 + \bar{X}1 \cdot Y1)$$

d)



Rechenaufgabe 3

a)

Abbildung 3b

Programm-Speicher	
Adresse	Instruktion
8000	MOV R1, #3
8004	MOV R2, #7
8008	MOV R3, #1
800C	CMP R1, R3
8010	JNE 801C
8014	DEC R2
8018	JMP 8024
801C	MOV R3, #2
8020	ADD R2, R3, R2
8024	NOP

b)

Register			
PC	R1	R2	R3
8004	0003	0000	0000
8008	0003	0007	0000
800C	0003	0007	0001
8010	0003	0007	0001
801C	0003	0007	0001
8020	0003	0007	0002
8024	0003	0009	0002
8028	0003	0009	0002