Technische Grundlagen der Informatik			**Test 2** **11.12.2015** 90 Minuten **Gruppe A**
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben. Unleserliche Antworten werden nicht gewertet!

Geben Sie bei Rechenaufgaben den **Lösungsweg** an!

Buch, Mitschriften, Ausdrucke von Folien, Handys, Taschenrechner etc. sind nicht zugelassen!

Zusatzblätter werden nicht akzeptiert!

Bei **Ankreuzfragen** werden Minuspunkte auf Teilaufgaben übernommen. Das Minimum je Gesamtaufgabe beträgt 0 Punkte.

1	[6]	[ ]
2	[6]	[ ]
3	[12]	[ ]
4	[15]	[ ]
5	[8]	[ ]
6	[10]	[ ]
7	[8]	[ ]
8	[8]	[ ]
9	[5]	[ ]
10	[10]	[ ]
11	[12]	[ ]
Summe	[100]	[ ]

---

1. *(6 Punkte)* Lesen Sie die Funktion $f(x_1, x_2, x_3, x_4)$ in minimaler disjunktiver Form aus!

	$\neg x_4$	x_4	$\neg x_4$	
$\neg x_3$	0	1	X	0
x_3	X	X	1	0
	1	1	1	0
$\neg x_3$	1	0	0	X
	x_1		$\neg x_1$	

x_2 (rows 1–2), $\neg x_2$ (rows 3–4)

$f(x_1, x_2, x_3, x_4) =$

Platz für Notizen:

2. *(6 Punkte)* Stellen Sie für die nachfolgend beschriebene Boolesche Funktion $F(e_2, e_1, e_0)$ die Wahrheitstafel auf und lesen Sie anschließend F in kanonisch konjunktiver Normalform aus!

   Die Eingangswerte e_2, e_1 und e_0 werden als Binärzahl $(e_2e_1e_0)_2$ im **Einerkomplement** interpretiert. Je nach Wert der Zahl Z ergeben sich für die Funktion F die folgenden Werte:

$$F(e_2, e_1, e_0) = \begin{cases} 0 & \text{falls} \quad Z < -1 \\ 1 & \text{falls} \quad Z \geq -1 \end{cases}$$

e_2	e_1	e_0	$F(e_2, e_1, e_0)$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

   $F(e_2, e_1, e_0)$ in KKNF:

Platz für Notizen:

3. *(12 Punkte)* Gegeben sei ein drei Bit langes Hamming-Codewort $c_1c_2c_3$. Die ersten beiden Bits entsprechen den Prüfbits $p_1 = c_1$, $p_2 = c_2$ und das dritte Bit entspricht dem Datenbit $d_1 = c_3$. Bei einem fehlerfreien Codewort gilt also:

$$\begin{aligned} p_1 &= c_3 \ mod\ 2 \\ p_2 &= c_3 \ mod\ 2 \end{aligned}$$

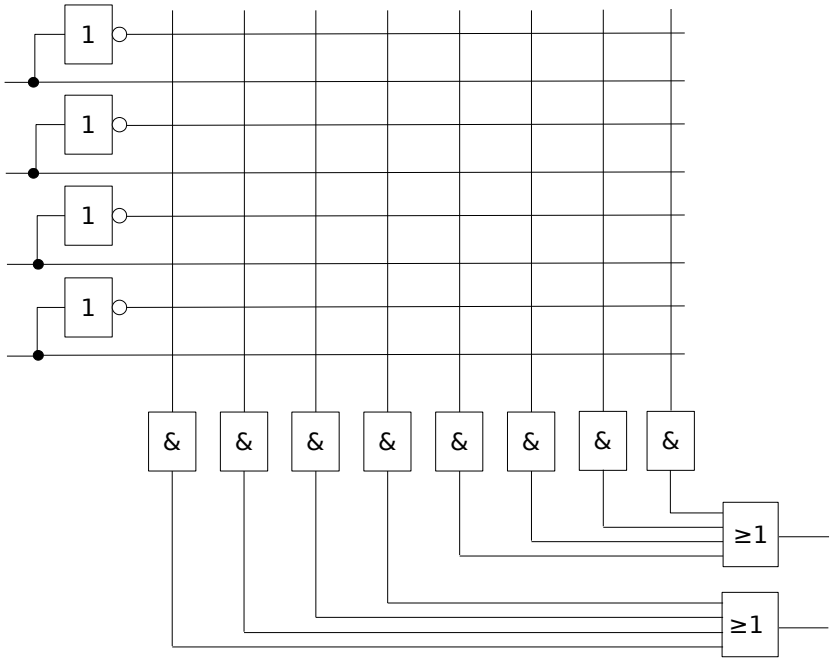
Entwerfen Sie ein Schaltnetz, welches ein derartiges drei Bit langes Hamming-Codewort einliest und überprüft, ob darin ein gestörtes Bit enthalten ist oder nicht (Mehrfachfehler werden nicht behandelt). Je nachdem, ob ein Bit fehlerhaft ist, soll für die Ausgänge a_1 und a_2 gelten:

Ausgabe a_1a_2	fehlerhaftes Bit
00	keines
01	c_1
10	c_2
11	c_3

(a) Befüllen Sie die zugehörige Wahrheitstafel!

(b) Realisieren Sie das Schaltnetz im nachfolgenden PAL.
Achten Sie auf die korrekte Beschriftung der Eingänge und Ausgänge!

c_1	c_2	c_3	a_1	a_2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

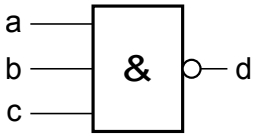


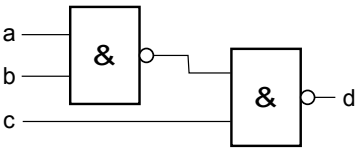
Platz für Notizen:

Platz für Notizen:

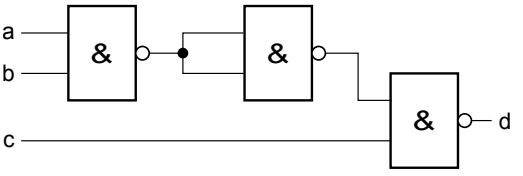
4. *(15 Punkte)* Gegeben sind die Schaltungen (1) und (2) sowie jeweils drei Umformungen. Geben Sie zu jeder Umformung an, ob es sich dabei um eine *gültige* oder *nicht gültige* Realisierung der gegebenen Schaltung handelt. Eine Realisierung ist gültig, wenn die jeweilige Umformung dieselbe Funktion ausführt wie die ursprünglich gegebene Schaltung.

Schaltung (1): Eingänge a, b und c sowie Ausgang d.
(korrekte Antwort: +2 Punkte; falsche Antwort: -2 Punkte; keine Antwort: 0 Punkte)

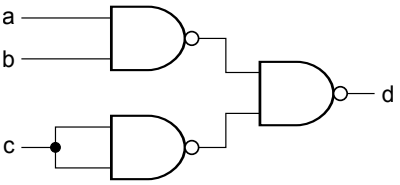




❍ gültig
❍ nicht gültig

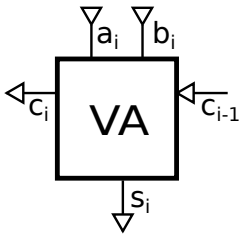


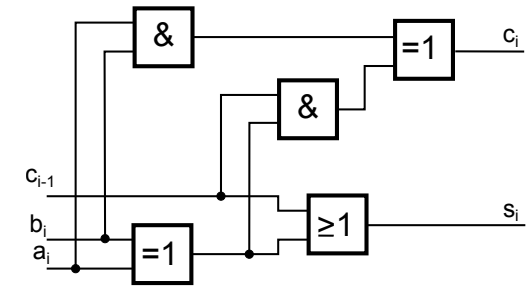
❍ gültig
❍ nicht gültig



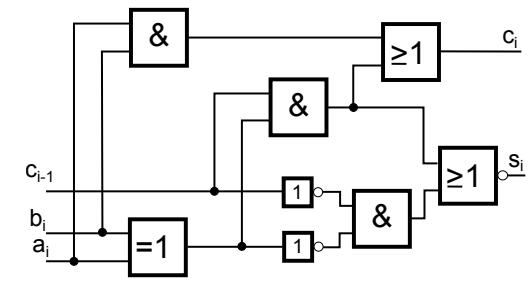
❍ gültig
❍ nicht gültig

Schaltung (2): Volladdierer mit Eingängen a_i, b_i und c_{i-1} sowie Ausgängen s_i und c_i.
(korrekte Antwort: +3 Punkte; falsche Antwort: -3 Punkte; keine Antwort: 0 Punkte)

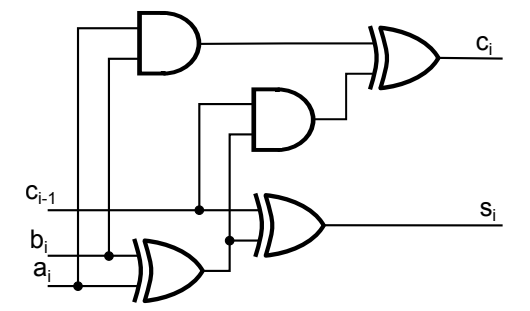




❍ gültig
❍ nicht gültig



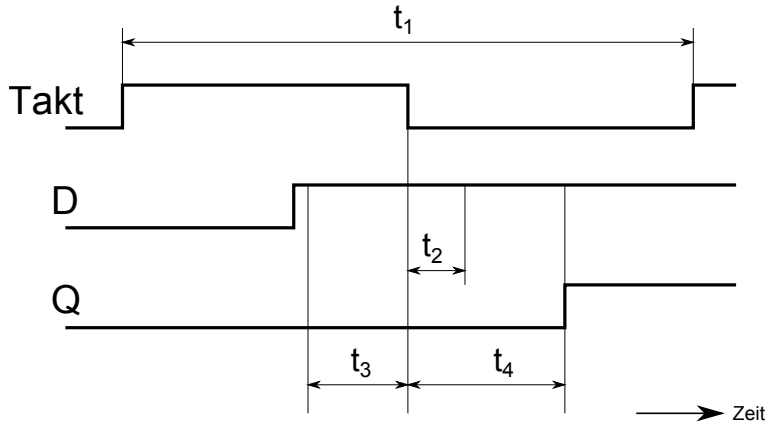
❍ gültig
❍ nicht gültig

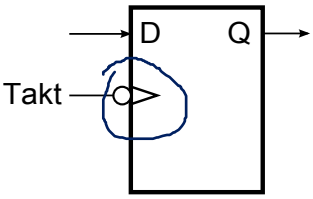


❍ gültig
❍ nicht gültig

t1 erstreckt sich von der ersten positiven Taktflanke bis zur nächsten positiven Taktflanke

5. *(8 Punkte)* Gegeben ist der Signalverlauf an den Ein- und Ausgängen eines D-Flip-Flops.





Benennen Sie die in der Grafik eingezeichneten Zeiten t_1 bis t_4 in nachfolgender Tabelle!

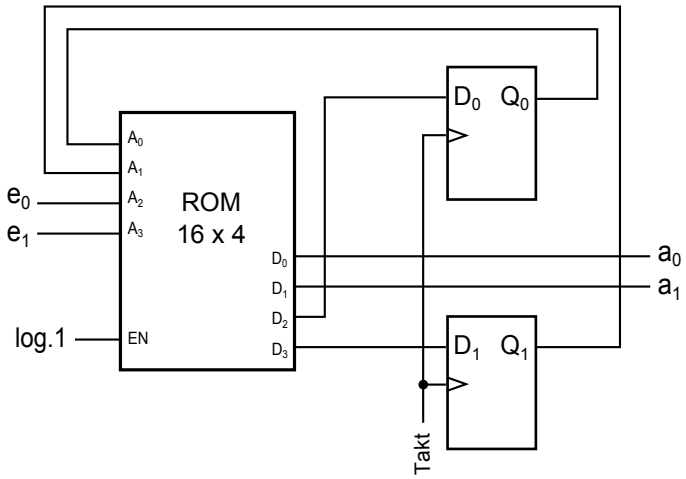
Bezeichnung in der Grafik	Benennung
t_1	Taktperiode
t_2	thold ... Haltezeit
t_3	t_setup Setup-zeit
t_4	tpd ... propagation delay

Platz für Notizen:

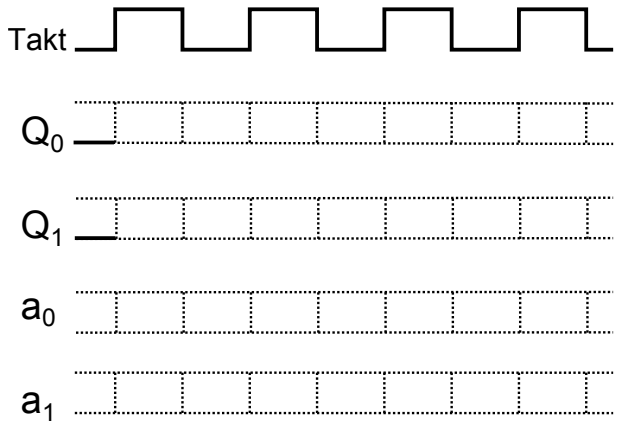
Zeit, die das Schaltwerk braucht um das Eingabesignal zu verarbeiten, und um das gültige Resultat darzustellen.

6. *(10 Punkte)* Gegeben ist das rechts dargestellte Schaltwerk mit zwei Eingabewerten e_0 und e_1 und zwei Ausgabewerten a_0 und a_1. Im 16×4 ROM-Baustein, der Teil der Schaltung ist, sind die in der Tabelle links angegebenen Werte gespeichert:

A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1	0	0
0	0	0	1	1	0	1	1
0	0	1	0	1	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	0
0	1	1	1	0	0	0	1
1	0	0	0	0	1	0	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	1	1	1
1	1	0	1	1	0	1	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1



(a) Zeichnen Sie den Verlauf der Signale Q_0, Q_1, a_0 und a_1 für die Eingangswerte $e_0e_1 = 00$!

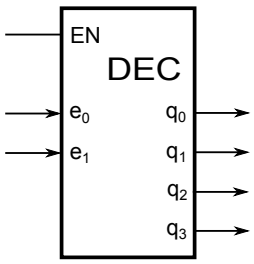


(b) Welche Art der Zustandscodierung wurde verwendet?
(korrekte Antwort: +2 Punkte, falsche Antwort: -2 Punkt, keine Antwort: 0 Punkte)

- ❍ (*n aus 1*) - Zustandscodierung
- ❍ (*1 aus n*) - Zustandscodierung
- ❍ (*1 aus 1*) - Zustandscodierung
- ❍ dichte Zustandscodierung

Platz für Notizen:

7. *(8 Punkte)* Der rechts abgebildete Decodierer soll als ROM realisiert werden.
Hinweis: Der ROM verfügt über eine separate Enable-Leitung.



(a) Wie viele Adressleitungen benötigt der ROM?

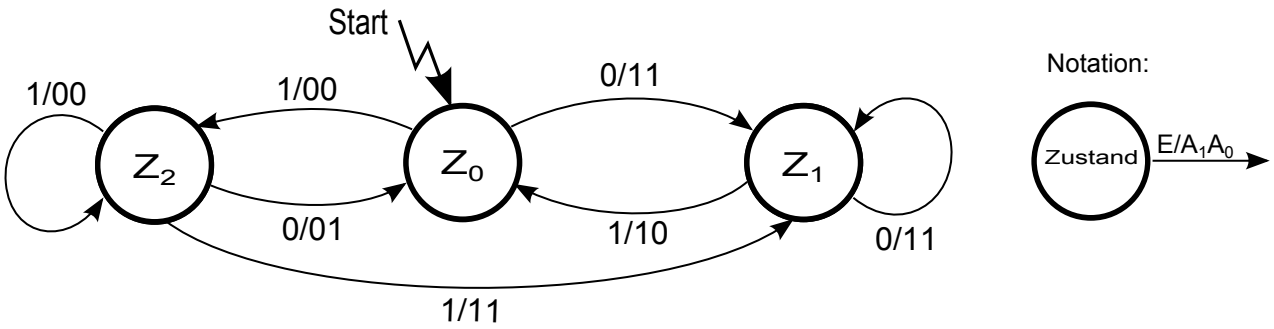
(b) Wie viele Datenleitungen benötigt der ROM?

(c) Wie viele Daten werden im ROM gespeichert?

(d) Wofür stehen die Abkürzungen ROM und RAM im Bereich der digitalen Speichertechnik?

Platz für Notizen:

8. *(8 Punkte)* Gegeben ist der folgende Automat:



Überprüfen Sie die folgenden Aussagen auf ihre Richtigkeit und kreuzen Sie entsprechend an!
(korrekte Antwort: +2 Punkte, falsche Antwort: -2 Punkte, keine Antwort: 0 Punkte)

Unabhängig vom Ausgangszustand befindet sich der Automat, nachdem zwei Takte lang '0' eingelesen wurde, immer in Z_1.	richtig	❍	❍	falsch
Der dargestellte Automat ist deterministisch.	richtig	❍	❍	falsch
Bei dem dargestellten Automaten handelt es sich um einen unendlichen Automaten.	richtig	❍	❍	falsch
Wenn der Automat eine k Bit lange Eingabe liest, liefert er eine $2k$ Bit lange Ausgabe.	richtig	❍	❍	falsch

---

9. *(5 Punkte)* Analysieren Sie folgende Micro16-Instruktionen und kreuzen Sie entsprechend an!
(korrekte Antwort: +1 Punkt, falsche Antwort: -1 Punkt, keine Antwort: 0 Punkte)

gültig ❍	❍ ungültig	`R5 ← 1011`
gültig ❍	❍ ungültig	`MAR ← MBR+1`
gültig ❍	❍ ungültig	`R3 ← R1+R2; if Z goto 5`
gültig ❍	❍ ungültig	`R3 ← ¬rsh(R6)`
gültig ❍	❍ ungültig	`MAR ← 1; MBR ← R1+1; R1 ← R1+1; wr`

Platz für Notizen:

10. *(10 Punkte)* Gegeben sind die folgenden Micro16-Instruktionen in binärer Notation:

	AMUX	COND	ALU	SH	MBR	MAR	RDWR	MS	ENS	S-BUS	B-BUS	A-BUS	ADR
A:	0	00	01	01	0	0	0	0	1	0101	0001	0001	0000 0000
B:	1	01	00	00	0	0	0	0	0	0000	0000	0000	0000 0110
C:	0	00	01	00	1	1	0	1	0	0000	0110	0010	0000 0000

Codierung	Beschreibung
ALU = 01	Addition (linker Operand auf A-Bus)
ALU = 10	bitweises UND (linker Operand auf A-Bus)
ALU = 11	bitweise Negation (A-Bus)
SH = 01	shift left
SH = 10	shift right
COND = 01	Sprung auf N=1
COND = 10	Sprung auf Z=1
COND = 11	unbedingter Sprung

Register	Adresse
0	0000
+1	0001
-1	0010
R0	0100
R1	0101
...	...
R10	1110

(a) Geben Sie die Instruktion A in hexadezimaler Form an!

(b) Geben Sie die Instruktionen A, B und C in symbolischer Notation an!
Hinweis: Es handelt sich um gültige Instruktionen.

Platz für Notizen:

11. *(12 Punkte)* Es soll ein Micro-Programm für den Micro16 Prozessor entworfen werden, das die bitweise Äquivalenz zweier 16-Bit Werte A und B berechnet: $C = A \equiv B$.
Die beiden 16-Bit Werte liegen initial in den Registern R1 und R2. Das Ergebnis C soll in Register R0 abgelegt werden.

Beispiel:
R1: 0000000011111111
R2: 0000111100001111
R0: 1111000000001111

Platz für Notizen: