

Technische Grundlagen der Informatik			Test 2 13.12.2019 90 Minuten Gruppe A
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben.
Verwenden Sie keine Korrekturhilfsmittel. Streichen Sie
Passagen, die nicht gewertet werden sollen, deutlich durch.

Unleserliche Antworten werden nicht gewertet!

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

Es sind keine Hilfsmittel zugelassen. Dies inkludiert Bücher,
Mitschriften, Ausdrücke von Folien, Smartphones, Smartwat-
ches, Taschenrechner etc.

Zusatzblätter werden nicht akzeptiert!

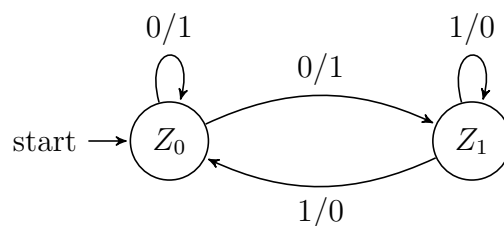
1	[5]	[]
2	[10]	[]
3	[7]	[]
4	[12]	[]
5	[13]	[]
6	[8]	[]
7	[15]	[]
8	[9]	[]
9	[11]	[]
10	[10]	[]
Summe	[100]	[]

Selbsteinschätzung:

[]

1. (____ / 5 Punkte) Gegeben ist der folgende Automat. Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt.

(richtig: +1 Punkte, falsch: -1 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)



	wahr	falsch
Der dargestellte Automat befindet sich immer dann in Zustand Z_1 , wenn zwei Mal hintereinander '0' eingelesen wurde.	<input type="radio"/>	<input checked="" type="radio"/>
Der dargestellte Automat ist deterministisch.	<input type="radio"/>	<input checked="" type="radio"/>
Der dargestellte Automat ist endlich.	<input checked="" type="radio"/>	<input type="radio"/>
Der dargestellte Automat ist vollständig.	<input type="radio"/>	<input checked="" type="radio"/>
Beim dargestellten Automaten werden immer genauso viele '0' ausgegeben wie eingelesen wurden.	<input type="radio"/>	<input checked="" type="radio"/>

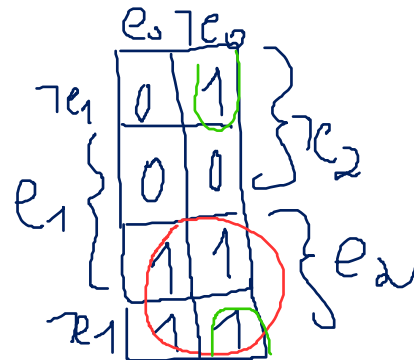
2. (____ / 10 Punkte) Gegeben ist folgende Boole'sche Funktion f :

$$f(e_2, e_1, e_0) = \begin{cases} 1 & \text{wenn } (e_2 e_1 e_0)_{2K} < (1)_{10} \\ 0 & \text{sonst} \end{cases}$$

Hierbei entspricht $(e_2 e_1 e_0)_{2K}$ der Interpretation der Bitfolge $(e_2 e_1 e_0)$ als Zahl im Zweierkomplement.

- (a) Befüllen Sie die zugehörige Wahrheitstabelle und ermitteln Sie mithilfe eines KV-Diagramms die minimale DNF.

	e_2	e_1	e_0	$f(e_2, e_1, e_0)$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
-4	1	0	0	1
-3	1	0	1	1
-2	1	1	0	1
-1	1	1	1	1



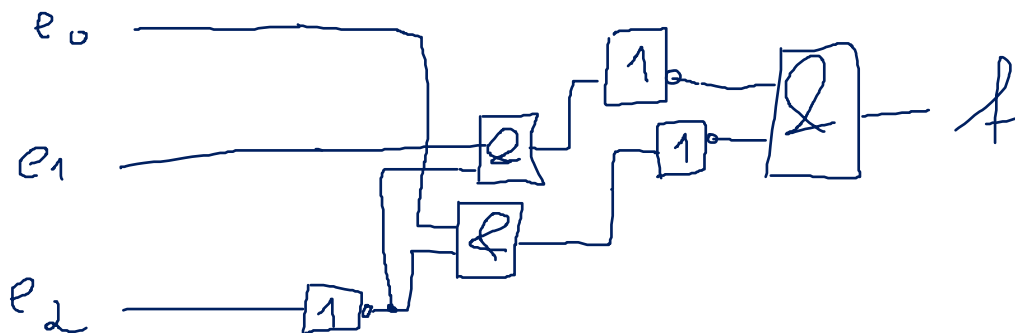
DNF: $e_2 \vee (\neg e_0 \wedge \neg e_1)$

- (b) Entwerfen Sie eine Schaltung mit drei Eingängen und einem Ausgang welche die Funktion

$$f(e_2, e_1, e_0) = (\neg e_0 \vee e_2) \wedge (\neg e_1 \vee e_2)$$

berechnet. Beschriften Sie entsprechend die Eingänge mit e_0 , e_1 und e_2 und den Ausgang mit f . Ihnen stehen folgende Bausteine zur Verfügung: **drei AND-Gatter** mit je zwei Eingängen und **3 NOT-Gatter**.

$$\neg(e_0 \wedge \neg e_2) \wedge \neg(e_1 \wedge \neg e_2)$$



3. (____ / 7 Punkte) Aus einer Wahrheitstabelle mit der Variablenreihenfolge $\pi = (x_1, x_2, x_3, x_4, x_5)$ wurde folgender Bead mit Don't-Cares ausgelesen:

00XX 1XX1 0XXX XXX1 X00X 111X 0XXX 11XX

Dieser Bead wurde mit dem in der Vorlesung vorgestellten Verfahren aus der Wahrheitstabelle von unten nach oben ausgelesen.

- (a) Erstellen Sie das minimale BDD mittels der **gierigen** Variante von Beads.

Kann man einen Square bilden?

00XX 1XX1 0XXX XXX1
X00X 111X 0XXX 11XX

0000 1111 0000 1111 =>
ein Square kann gebildet werden =>
somit kann die erste Ebene übersprungen werden

Die zweite Ebene ist ebenfalls ein Square

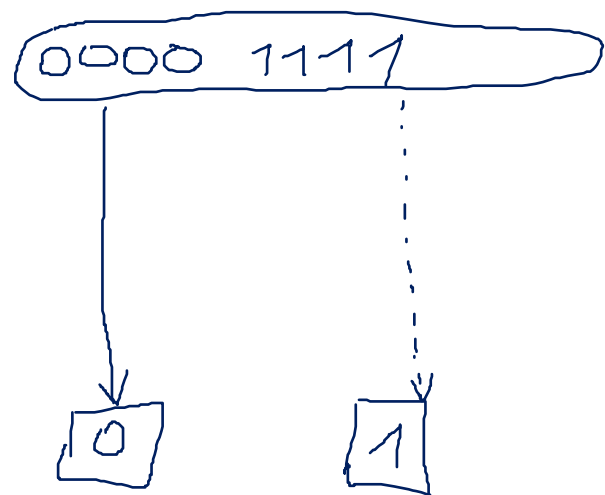
Aufpassen, da die Beads verkehrt herum
aus der Wahrheitstabelle ausgelesen werden.

X₁

X₂

X₃

X₄



- (b) Erstellen Sie die ITE-Darstellung Ihres BDDs aus Unteraufgabe (a).

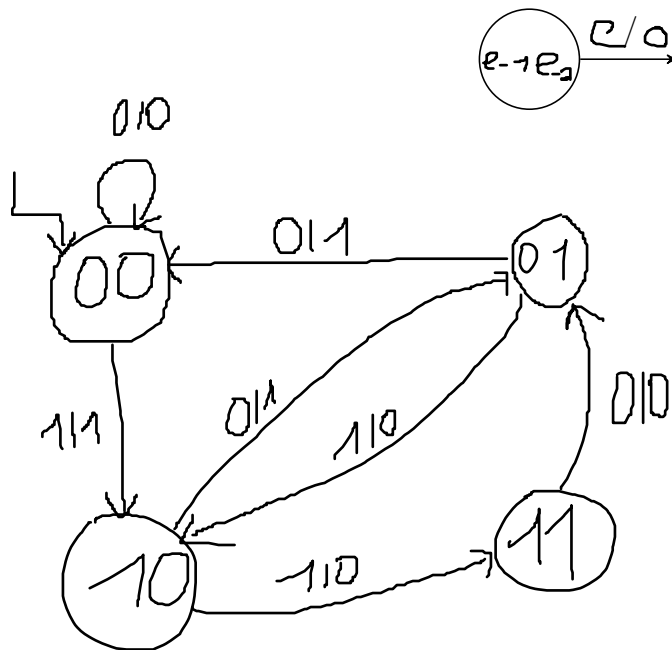
if x₃ then 0
else 1

4. (____ / 12 Punkte) Entwerfen Sie einen **Mealy-Automaten** mit folgender Funktionalität: der Mealy-Automat liest ein Symbol (Bit) **e** ein und gibt ein Symbol (Bit) **a** aus. Die Ausgabe a wird entsprechend dem Boole'schen Ausdruck

$$e \oplus e_{-1} \oplus e_{-2}$$

berechnet. Hier bezeichnet e_{-1} jenes Eingabesymbol, welches vor e aufgetreten ist, und e_{-2} jenes Eingabesymbol, welches vor e_{-1} aufgetreten ist. Nehmen Sie dabei an, dass zu Beginn $e_{-1} = e_{-2} = 0$ gilt. \oplus bezeichnet die XOR-Operation.

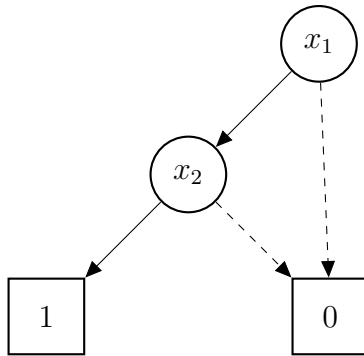
Verwenden Sie **maximal vier** Zustände und erklären Sie die verwendete Notation:



Die Ausgabe des Automaten ist das Ergebnis der XOR Operation zwischen e , e_{-1} , e_{-2}

Wenn nun 1 eingelesen wird, dann wechselt man in den Zustand 10. Es wird 1 ausgegeben, da die XOR Operation eine 1 liefert. Nach diesem Konstruktionsschema kann man den weiteren Automaten konstruieren.

5. (____ / 13 Punkte) Der nachstehende BDD beschreibt eine Boole'sche Funktion $f(x_1, x_2)$.



Wenn man die Wahrheitstabelle aufstellen will, muss einfach die Pfade zur 0 und 1 auslesen. Dabei muss man auf die negierten und nicht negierten Kanten achten. Wie man sieht, führt nur eine Kombination zu einer 1 und zwar wenn beide Variablen nicht negiert sind.

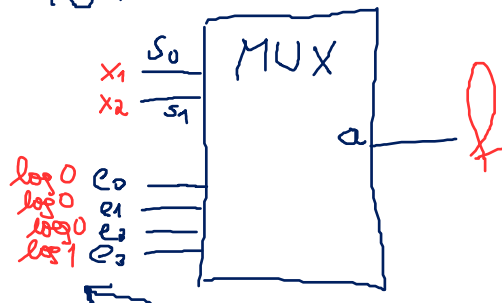
Q1)

	x_1	x_2	f
	0	0	0
	0	1	0
	1	0	0
	1	1	1

- (a) Geben Sie eine Wahrheitstabelle für die im BDD codierte Funktion $f(x_1, x_2)$ an.
- (b) Realisieren Sie die Funktion $f(x_1, x_2)$ mithilfe eines (4 to 1)-MUX Bausteins als Schaltung. Sie können dabei auch Leitungen direkt mit $\log 0$ bzw. $\log 1$ verbinden. Vergessen Sie nicht alle Ein- und Ausgangsleitungen des MUX Bausteins zu beschriften.

MUX

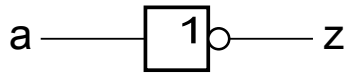
Mit den Select-Eingängen kann man wählen, welcher Eingang durchgeschaltet wird. Liegt also 0, 0 an wird e_0 durchgeschaltet. Und so weiter. Mit x_1 und x_2 kann die Zeile der Wahrheitstabelle wählen.



- (c) Gegeben sind die Schaltnetze (1) und (2) sowie jeweils 2 Umformungen. Geben Sie zu jeder Umformung an, ob sie *gültig* oder *nicht gültig* ist. Eine Umformung ist gültig, wenn das umgeformte Schaltnetz dieselbe Funktion wie das ursprünglich gegebene Schaltnetz ausführt.

(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

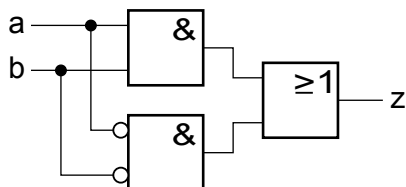
Schaltnetz (1) :



Kreuzen Sie rechts an, ob es sich um eine *gültige* oder um eine *nicht gültige* Umformung handelt.

	<input type="radio"/> gültig <input checked="" type="radio"/> nicht gültig
	<input checked="" type="radio"/> gültig <input type="radio"/> nicht gültig

Schaltnetz (2) :

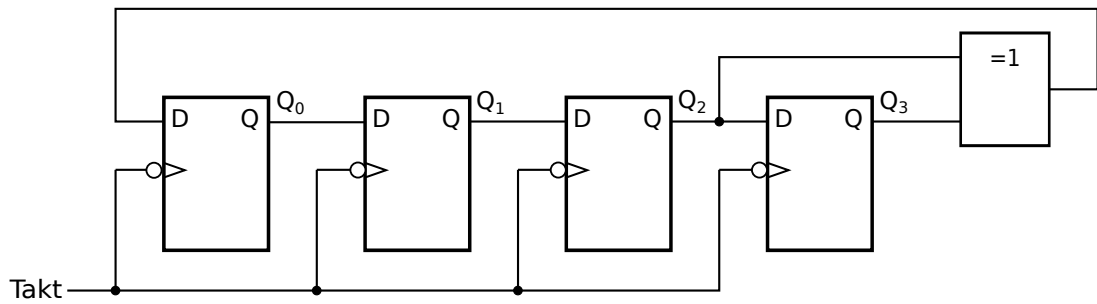


Kreuzen Sie rechts an, ob es sich um eine *gültige* oder um eine *nicht gültige* Umformung handelt.

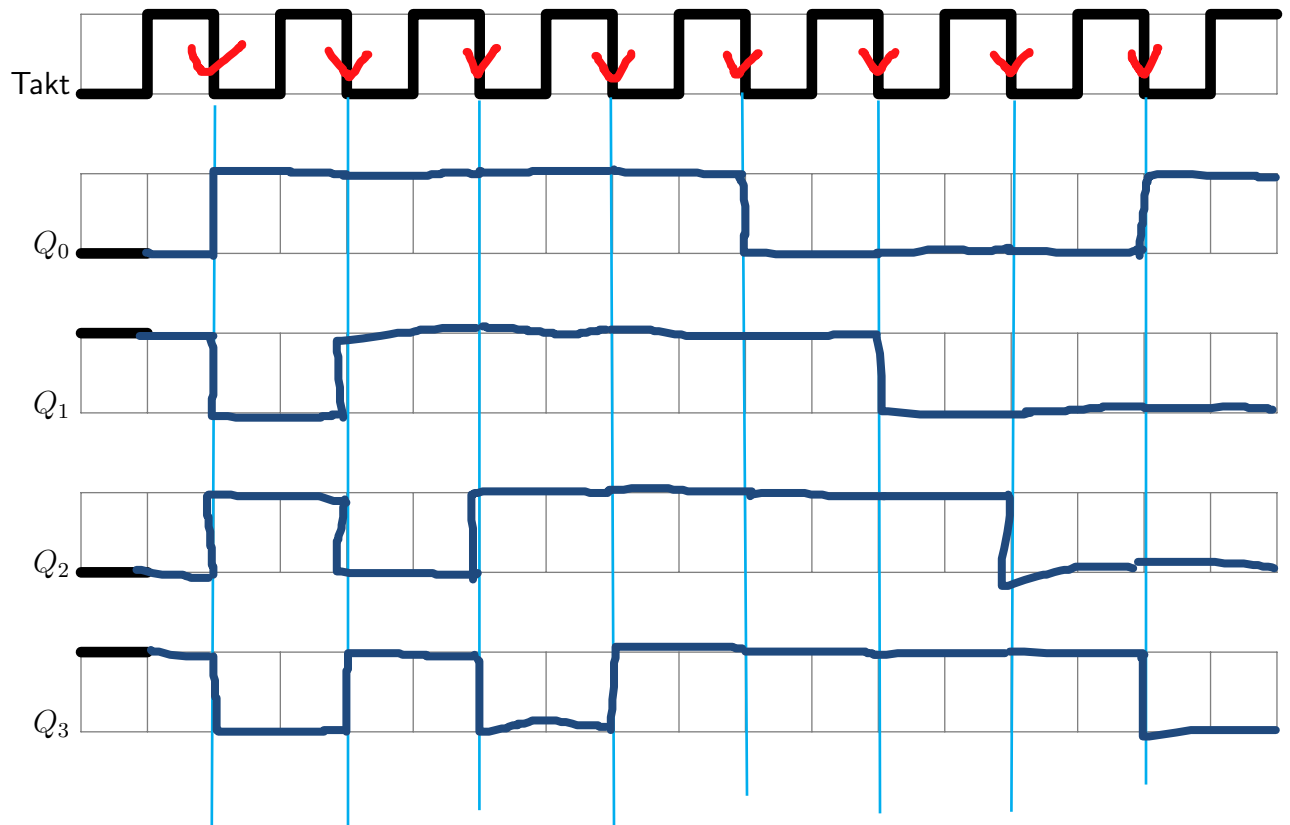
	<input checked="" type="radio"/> gültig <input type="radio"/> nicht gültig
	<input checked="" type="radio"/> gültig <input type="radio"/> nicht gültig

Hier haben wir es mit Taktflankengesteuerten D-Flipflops zu tun, die auf eine negative Taktflanke reagieren, wie man an dem Krinkel erkennen kann. Deshalb kann man sich die negativen Taktflanken einzeichnen, da sich nur zu diesem Zeitpunkt etwas verändert. Alle Elemente sind von der negativen Taktflanke abhängig.

6. (____ / 8 Punkte) Gegeben ist das dargestellte rückgekoppelte Schieberegister.



Initial ist der Inhalt des Schieberegisters $(Q_0, Q_1, Q_2, Q_3) = (L, H, L, H)$ ($L \dots$ Low, $H \dots$ High). Zeichnen Sie den Verlauf der Signale Q_0 , Q_1 , Q_2 und Q_3 .



7. (____ / 15 Punkte) Gegeben ist der O'Brien-Code zur binären Kodierung von Dezimalziffern.

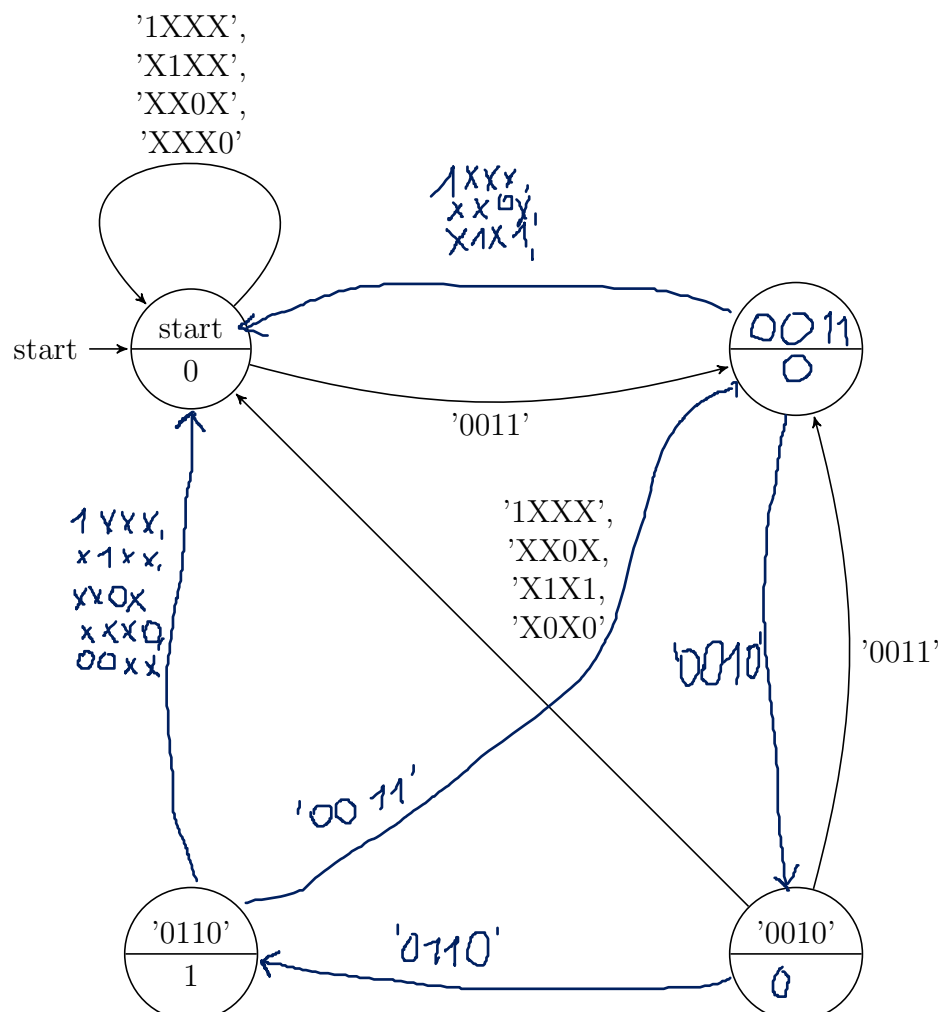
Ziffer	D_3	D_2	D_1	D_0
0	0	0	0	1
1	0	0	1	1
2	0	0	1	0
3	0	1	1	0
4	0	1	0	0
5	1	1	0	0
6	1	1	1	0
7	1	0	1	0
8	1	0	1	1
9	1	0	0	1

Eine Besonderheit dieses Codes ist, dass die Codewörter zwei aufeinanderfolgender Ziffern sich immer nur in einem Bit unterscheiden (nicht relevant für die Lösung dieses Beispiels).

(a) Vervollständigen Sie folgenden Moore Automaten:

Der Automat soll die im O'Brien-Code codierte Zahlenfolgen 1,2,3, also '0011', '0010', '0110' erkennen. Er besitzt also 4 Eingänge D_3, D_2, D_1, D_0 und einen Ausgang. Werden in 3 aufeinanderfolgenden Takten die O'Brian Codewörter '0011', '0010', '0110' in dieser Reihenfolge am Eingang erkannt, so soll der Ausgang A für einen Taktzyklus auf '1' gesetzt werden. In allen anderen Zuständen soll '0' ausgegeben werden.

Hinweis: Verwenden Sie *Don't Care* 'X' zum Beschriften der Kanten.



- (b) Im Folgenden soll ein Schaltwerk entworfen werden, welches einen zyklischen O'Brien-Code Zähler realisiert. Der Zählwert ($D_3D_2D_1D_0$) soll mit jedem Takt um eins erhöht werden und bei Erreichen des Zählwertes ($D_3D_2D_1D_0$) = '1001' soll der Ausgang A des Schaltwerks auf '1' gesetzt werden. Im darauffolgenden Takt soll wieder mit '0001' zu zählen begonnen werden und die Ausgabe A auf '0' zurück gesetzt werden.

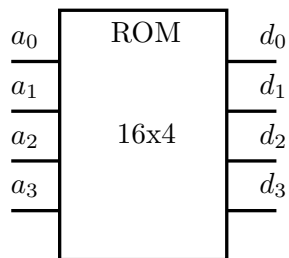
Die Zustandsübergangsfunktion des Schaltwerks soll mittels 16x4 ROM umgesetzt werden. Überlegen Sie sich wie der ROM-Inhalt aussehen muss, wenn für die Zustandskodierung der O'Brien-Code verwendet wird. Sollte aus irgendwelchen Gründen (z.B. Logical Hazards/Glitches) am Adresseingang des ROMs ein ungültiger Vorgängerzustand auftreten, so soll der Automat wieder mit '0001' zu zählen beginnen.

ROM:

Adressen				Datenbits			
a_3	a_2	a_1	a_0	d_3	d_2	d_1	d_0
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

- (c) Ermitteln Sie die Boole'sche Funktion $f(D_3, D_2, D_1, D_0)$ zur Berechnung des Ausgabebits $A = f(D_3, D_2, D_1, D_0)$ des in (b) beschriebenen Schaltwerks in Abhängigkeit des aktuellen Zählstandes ($D_3D_2D_1D_0$).

- (d) Skizzieren Sie das Schaltwerk des in (b) beschriebenen zyklischen O'Brian-Code Zählers. Verwenden Sie dafür das 16x4 ROM aus (b), D-Flip-Flops sowie die notwendigen Logik-Gatter zur Berechnung der Ausgabe A . Beschriften Sie auch die Signalleitungen an denen der aktuelle Zählstand ($D_3D_2D_1D_0$) abgelesen werden kann.



A

Takt

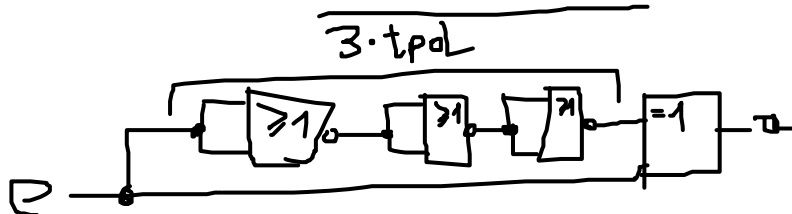
8. (____ / 9 Punkte)

(a) Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt.

(richtig: +1 Punkt, falsch: -1 Punkt, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

	wahr	falsch
Die Haltezeit (hold time) eines Gatters ist in dessen Vorbereitungszeit (setup time) enthalten.	<input type="radio"/>	<input checked="" type="radio"/>
Ein Tri-State Ausgang kann drei physikalische Zustände einnehmen: HIGH (H), LOW (L) und OPEN (Z).	<input checked="" type="radio"/>	<input type="radio"/>
Funktionsspeicher können auch mit Tabellenspeicher realisiert werden.	<input checked="" type="radio"/>	<input type="radio"/>
Bei einem KV-Diagramm werden zur Minimierung immer Blöcke beliebiger Größe gebildet.	<input type="radio"/>	<input checked="" type="radio"/>
Ein Encoder (Codierer) verdichtet Information, somit ist die Eingangscodelänge größer als die Ausgangscodelänge.	<input checked="" type="radio"/>	<input type="radio"/>
Aus einem BDD kann immer die minimale konjunktive Normalform abgelesen werden, indem alle Pfade von der Wurzel bis zur 0 betrachtet und entsprechend ausgelesen werden.	<input type="radio"/>	<input checked="" type="radio"/>

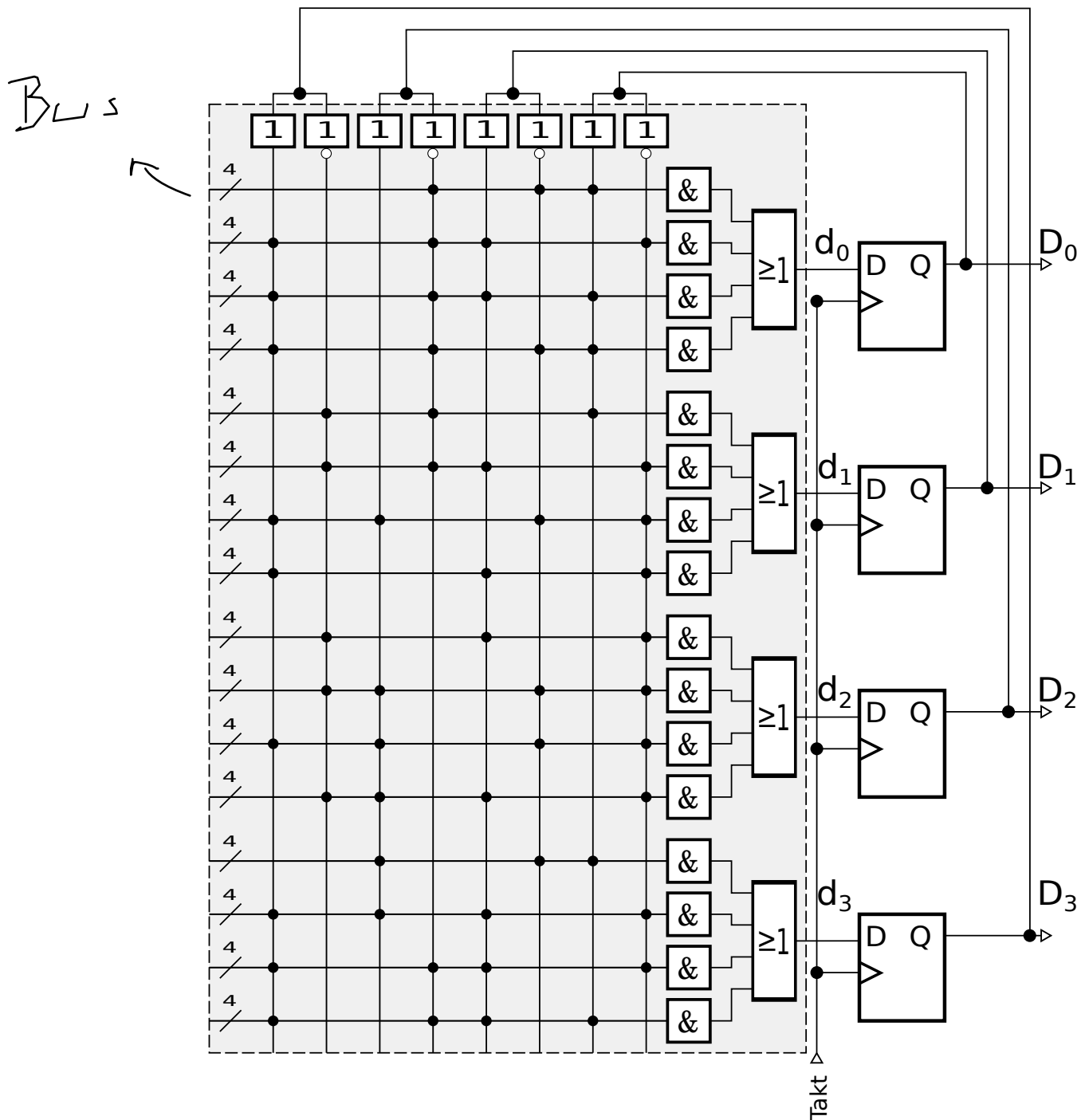
(b) Konstruieren Sie eine monostabile Kippstufe mit Eingang e , welche im stabilen Zustand am Ausgang a eine 1 ausgibt und die Realisierung folgendes Verhalten zeigt: sowohl bei einer positiven als auch bei einer negativen Flanke am Eingang e wird am Ausgang a für die Dauer von drei Gatterdurchlaufzeiten ($3 \cdot t_{pd}$) eine 0 ausgegeben. Verwenden Sie ausschliesslich XOR- und NOR-Gatter mit zwei Eingängen.



	NOR	XOR
00	1	0
01	0	1
10	0	1
11	1	0



9. (____ / 11 Punkte) Folgende Schaltung bestehend aus vier D-Flip-Flops und einem PLD realisiert einen zyklischen O'Brian Zähler:



- (a) Lesen Sie aus dem PLD die darin programmierte Boole'sche Funktion $\underline{d_0} = f_0(D_0, D_1, D_2, D_3)$ in disjunktiver Normalform aus.

$$d_0 = (\neg D_2 \wedge \neg D_1 \wedge D_0) \vee (D_3 \wedge \neg D_2 \wedge D_1 \wedge \neg D_0) \vee (D_3 \wedge \neg D_2 \wedge D_1 \wedge D_0) \vee (D_3 \wedge D_2 \wedge \neg D_1 \wedge D_0)$$

- (b) Ermitteln Sie mit Hilfe eines KV-Diagramms eine minimale konjunktive Normalform der Funktion:

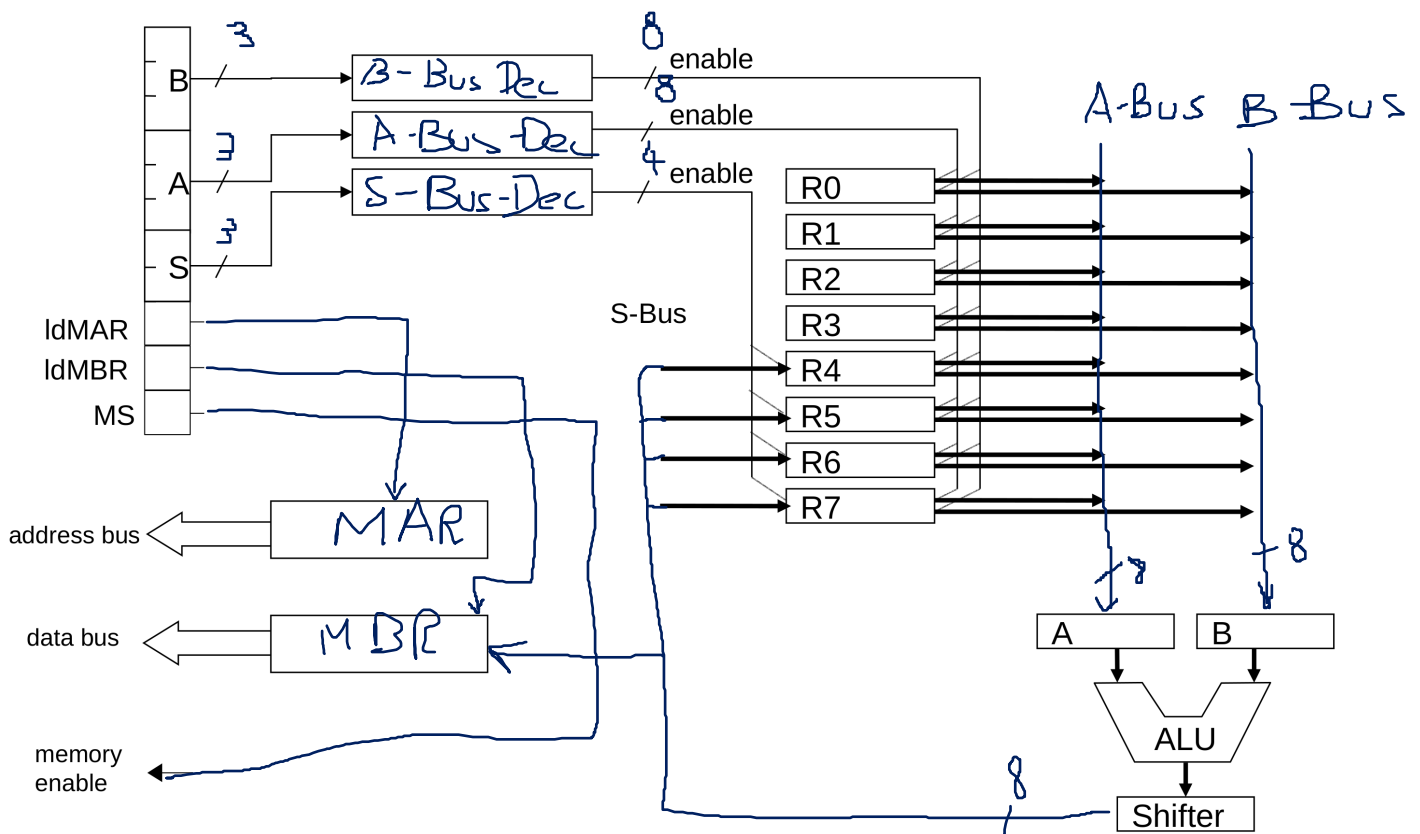
$$d_3 = (\neg D_0 \wedge \neg D_1 \wedge D_2) \vee (\neg D_0 \wedge D_1 \wedge D_2 \wedge D_3) \vee (\neg D_0 \wedge D_1 \wedge \neg D_2 \wedge D_3) \vee (D_0 \wedge D_1 \wedge \neg D_2 \wedge D_3)$$

	$\neg D_3$	D_3	D_3	$\neg D_3$	
$\neg D_2$	0	0	0	0	$\neg D_1$
D_2	0	0	1	1	$\neg D_1$
D_2	0	0	1	0	D_1
$\neg D_2$	0	1	1	0	D_1
	D_0	D_0	$\neg D_0$	$\neg D_0$	

$$d_3 = (D_2 \vee D_1) \wedge (\neg D_2 \vee \neg D_0) \wedge (\neg D_1 \vee D_3)$$

10. (____ / 10 Punkte) Mikroprozessorarchitektur

Die nachfolgende Abbildung gibt den groben Aufbau einer 8-Bit Mikroprozessorarchitektur wieder, die aus einer ALU und einem Shifter besteht. Das Register-File besteht aus 8 Registern R0-R7, die eine Datenwortgröße von jeweils 8 Bit haben. Das Ergebnis soll in den 4 Registern R4-R7 gespeichert werden können. Weiters soll die 8-Bit Mikroprozessorarchitektur über ein Memory Address Register (MAR) und ein Memory Buffer Register (MBR) mit einem externen Speicher verbunden werden können um Ergebnisse sichern zu können (kein Lesezugriff). Die Adresse wird über den A-Bus in das MAR gespeichert. Das MBR bekommt seine Daten über den S-BUS geliefert.



- (a) Vervollständigen Sie die oben angegebene Abbildung durch Einzeichnen der **fehlenden Signalleitungen und Busse** sowie durch Beschriftung der Schaltungselemente (**Register, Bus-Decoder**). Vergessen Sie nicht die **Busbreiten** und **Namen der Busse** anzugeben.
- (b) Wie groß ist der Adressbereich, den Sie über das MAR ansprechen können?

$$2^8 = 256 \text{ Adr.}$$