Technische Grundlagen der Informatik			**Test 2** **16.12.2016** 100 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	[10]	[ ]
3	[12]	[ ]
4	[12]	[ ]
5	[10]	[ ]
6	[10]	[ ]
7	[12]	[ ]
8	[10]	[ ]
9	[8]	[ ]
10	[10]	[ ]
Summe	[100]	[ ]

---

1. *(6 Punkte)* Die minimale disjunktive Form einer Booleschen Funktion $f(a, b, c, d)$ sei $\neg c \vee (\neg a \wedge \neg b)$.

   (a) Aus *wie vielen* Blöcken in einem KV-Diagramm wurde die obige minimale disjunktive Form gebildet?

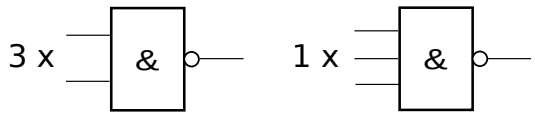
   (b) Wie lautet die minimale konjunktive Form von f?

2. *(10 Punkte)* Es soll eine *2-aus-3-Schaltung* entworfen werden. Dabei wird der Ausgang f genau und nur dann logisch 1, wenn mindestens zwei der drei Eingänge x, y und z logisch 1 sind.

   (a) Befüllen Sie die zugehörige Wahrheitstafel:

z	y	x	$f(x,y,z)$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

   (b) Entwerfen Sie grafisch die *2-aus-3-Schaltung* und beschriften Sie entsprechend alle Ein- und Ausgänge, wobei Ihnen nur folgende Bausteine zur Verfügung stehen: drei NAND-Gatter mit je zwei Eingängen und ein NAND-Gatter mit drei Eingängen:

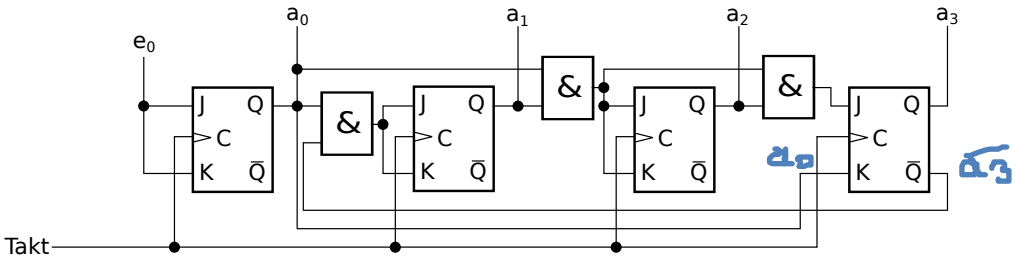


   Ihre Schaltung:

   (c) Kreuzen Sie an, ob die folgenden Booleschen Ausdrücke der Funktionalität der beschriebenen *2-aus-3-Schaltung* korrekt entsprechen oder nicht.
   (richtige Antwort: +1 Punkt, falsche Antwort: -1 Punkt, keine Antwort: 0 Punkte)

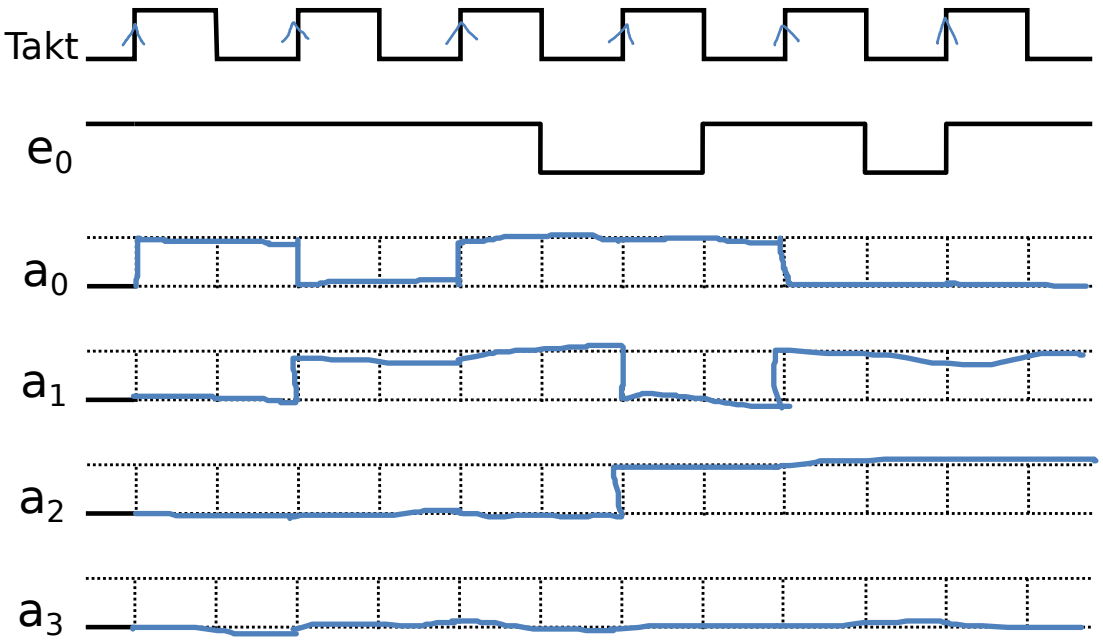
korrekt ❍	❍ nicht korrekt	`x ∧ (y ∨ z) ∨ (y ∧ z)`
korrekt ❍	❍ nicht korrekt	`x ∨ (x ∨ z) ∧ (x ∧ z)`
korrekt ❍	❍ nicht korrekt	`(y ∧ z) ∨ (x ∧ y ∧ z)`
korrekt ❍	❍ nicht korrekt	`(x ∧ z) ∨ y ∧ (x ∨ z)`

3. *(12 Punkte)* Gegeben ist die folgende Schaltung:



Man muss immer schauen, was kurz vor dem Takt war.

Vervollständigen Sie das zugehörige Timing-Diagramm!



Platz für Notizen:

Zuerst muss man schauen, auf welche Taktflanke die Flipflops reagieren. In diesem Beispiel reagieren die Flipflops auf die positiven Taktflanken. Daher muss man sich nur die positiven Taktflanken anschauen und nach unten zeichnen, da nur dort der Ausgang verändert werden kann.

Das erste JK-Flipflop hat beide Eingänge zusammengeschalten. a0 ist der Ausgang des ersten JK-Flipflops, weshalb man a0 schon ganz zeichnen kann. Die Besonderheit des JK-Flipflops ist, dass bei einer Eingabe von 1 1 der Flipflop toggelt.

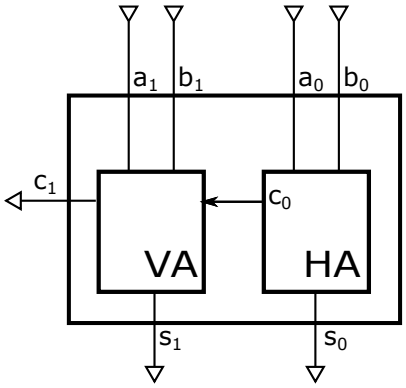
Das zweite JK-Flipflop hat ebenso beide Eingänge zusammengeschalten. Es berechnet sich aus der UND-Verknüpfung von a0 und der negation von a3 also Q(quer) wie man oben erkennen kann.

Das dritte JK-Flipflop hat wieder beide Eingänge zusammengeschalten. Es berechnet sich aus der UND-Verknüpfung von a0 und a1.

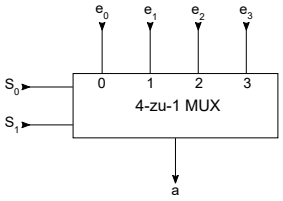
Das vierte JK-Flipflop hat beide Eingänge nicht zusammengeschalten. J berechnet sich aus der UND-Verknüpfung von a2, a1 und a0. k berechnet sich aus a0

Platz für Notizen:

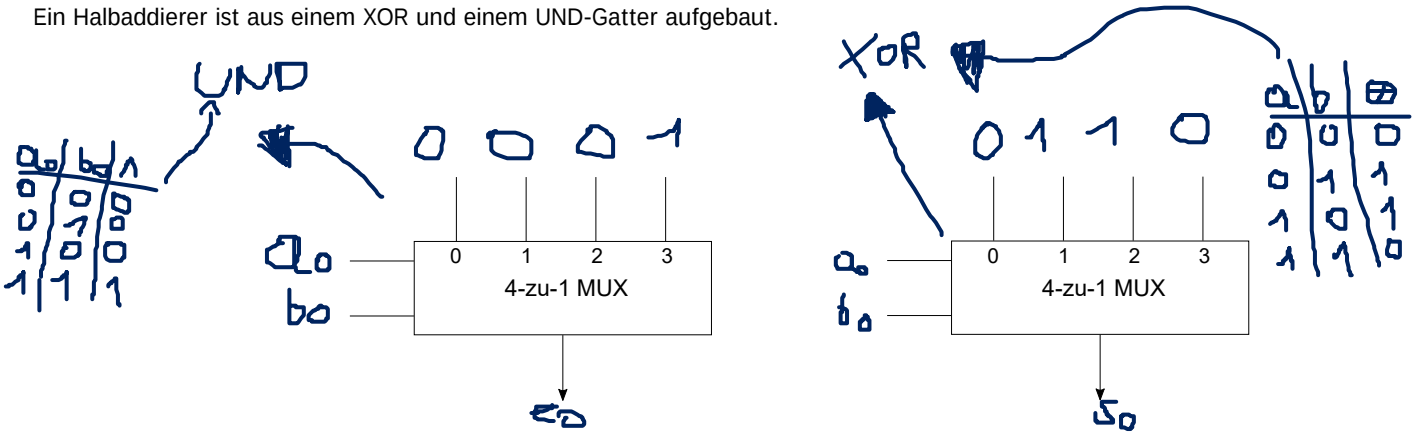
4. *(12 Punkte)* Sie haben folgendes Schaltnetz für einen 2-Bit-Addierer gegeben.

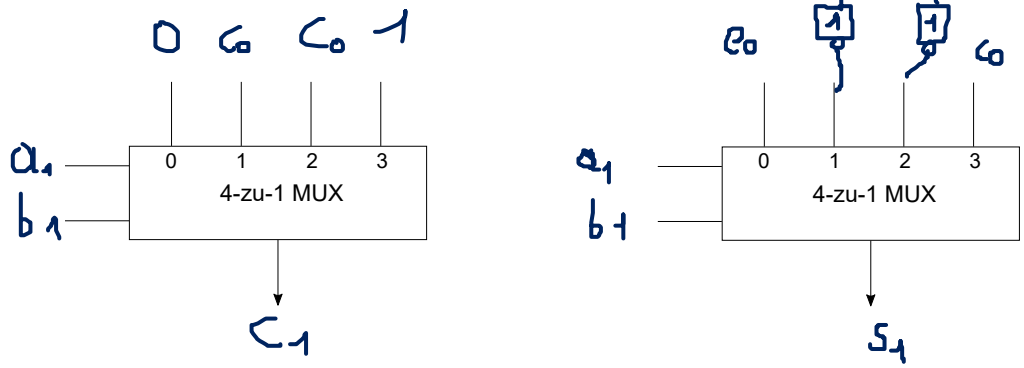


Realisieren Sie nun nur mit Multiplexern ein Schaltnetz, das die Funktionalität dieses 2-Bit-Addierers nachbildet. Ihnen stehen zur Lösung dieser Aufgabe **vier** 4-zu-1-Multiplexer (siehe Abbildung rechts) zur Verfügung. Weiters dürfen Sie Not-Gatter verwenden und bei Bedarf logisch 0 und 1 an Eingängen anlegen.



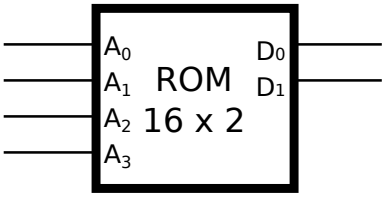
Ein Halbaddierer ist aus einem XOR und einem UND-Gatter aufgebaut.





a_1	b_1	c_0	s_1	c_1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

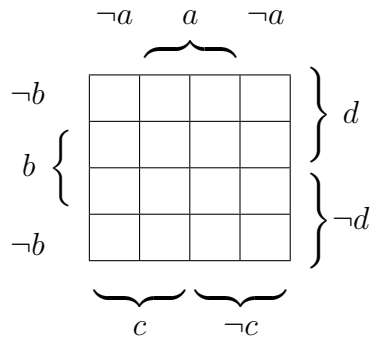
5. *(10 Punkte)* Gegeben ist folgender ROM Speicher:



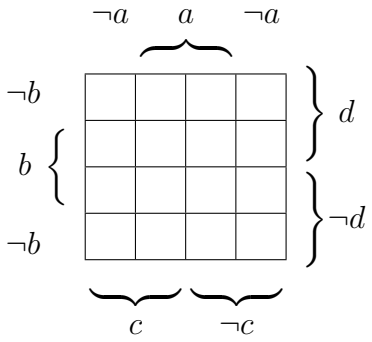
A_3 a	A_2 b	A_1 c	A_0 d	D_0 x	D_1 y
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

(a) Ermitteln Sie die im ROM realisierten Booleschen Ausdrücke x und y mit Hilfe von KV-Diagrammen. Achten Sie auf einen minimalen Ausdruck!

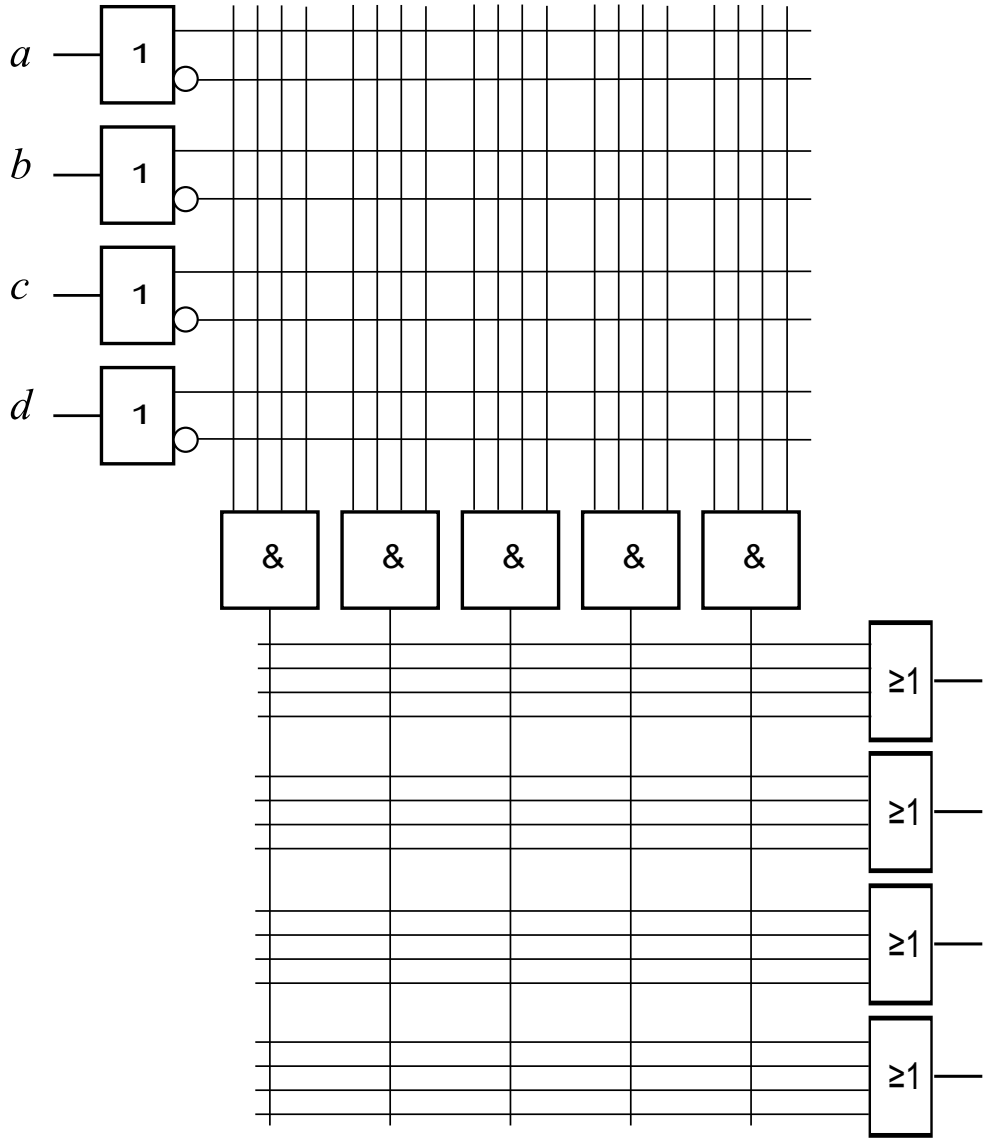
x:



y:



(b) Realisieren Sie diese Funktionen mit dem nachfolgenden PLA und beschriften Sie die verwendeten Ausgänge des PLAs entsprechend. Der PLA darf nicht durch zusätzliche Leitungen erweitert werden und es stehen Ihnen auch keine weiteren Bauteile zur Verfügung!



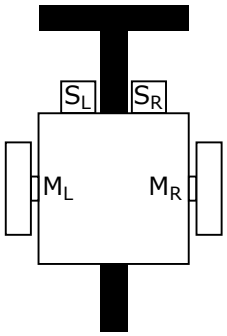
6. *(10 Punkte)* Kreuzen Sie an, ob es sich um gültige oder ungültige Aussagen handelt!
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

gültig	ungültig	
❍	❍	Funktionsspeicher können auch mit Tabellenspeicher realisiert werden.
❍	❍	Ein Tri-State Ausgang kann drei physikalische Zustände einnehmen: HIGH (H), LOW (L) und OPEN (Z).
❍	❍	Bei einem KV-Diagramm werden zur Minimierung immer Blöcke beliebiger Größe gebildet.
❍	❍	Ein Encoder (Codierer) verdichtet Information, somit ist die Eingangscodelänge größer als die Ausgangscodelänge.
❍	❍	Die Haltezeit (hold time) eines Gatters ist in dessen Vorbereitungszeit (setup time) enthalten.

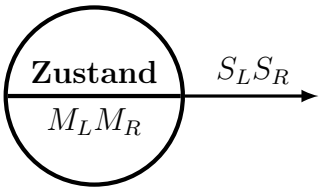
Platz für Notizen:

7. *(12 Punkte)* Realisieren Sie einen Line-Follower Roboter mittels Moore-Automaten.

Um einer schwarzen Linie auf einem weißen Untergrund folgen zu können, besitzt der Roboter zwei Sensoren, welche sich in Fahrtrichtung vorne befinden. Ein Sensor (S_L oder S_R) gibt auf dem weißen Untergrund den logischen Wert '0' bzw. auf der schwarzen Linie '1' aus. Befindet sich der Roboter genau auf der schwarzen Linie, detektieren beide Sensoren den weißen Untergrund und der Roboter fährt gerade aus. Sollten beide Sensoren '1' ausgeben, soll der Roboter stehen bleiben. Sobald der Roboter, während der Geradeausfahrt, links oder rechts von der schwarzen Linie abweicht, muss entsprechend gegengesteuert werden. Die Fortbewegung des Roboters erfolgt mittels zweier Motoren mit Rädern M_L und M_R. Die Kombination $M_L M_R$ = '11' lässt den Roboter **vorwärts** fahren, '10' führt zu einer **Drehung im Uhrzeigersinn** und '01' zu einer **Drehung gegen den Uhrzeigersinn**. Bei $M_L M_R$ = '00' **stoppt** der Roboter.
Hinweis: Dieser Automat lässt sich mit vier Zuständen realisieren! Zu Beginn steht der Roboter.



Notation:



8. *(10 Punkte)* Schreiben Sie ein Micro16 Programm, mit dem das k-te Bit in einem Datenwort auf logisch 0 gesetzt werden kann. Im Register AC befindet sich die Bitnummer k ($0 \leq k \leq 15$). Das Datenwort ist im Register R0 abgelegt. Das modifizierte Datenwort soll am Ende wieder im Register R0 gespeichert werden.

   Hinweis: Die Nummerierung der Bits erfolgt in der üblichen Weise von rechts nach links mit 0 beginnend. Das Bit mit der Nummer 0 ist daher das Bit mit der geringsten Wertigkeit (lsb), das Bit mit der Nummer 15 jenes mit der höchsten Wertigkeit (msb).

   Beispiele:
   AC = 00000000 00000100, R0 = 10101010 010$\underline{1}$0101 → R0 = 10101010 010$\underline{0}$0101
   AC = 00000000 00000000, R0 = 11111111 11111111 → R0 = 11111111 11111110

Platz für Notizen:

9. *(8 Punkte)* Übersetzen Sie die nachfolgenden Instruktionen in Micro16-Code! Bei den Instruktionen handelt es sich um korrekte Micro16-Instruktionen.

Instruktion A: `MAR ← (-1); rd`

Instruktion B: `R0 ← ¬(1); rd`

Instruktion C: `R0 ← MBR & R0`

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) Tragen Sie die Instruktionen in binärer Form in die nachfolgende Tabelle ein!

	AMUX	COND	ALU	SH	MBR	MAR	RDWR	MS	ENS	S-BUS	B-BUS	A-BUS	ADR
A:													
B:													
C:													

(b) Steht nach der Ausführung eine gerade oder ungerade Zahl im Register R0? Begründen Sie ihre Antwort!

Platz für Notizen:

10. *(10 Punkte)* Kreuzen Sie an, ob es sich um gültige oder ungültige Aussagen handelt!
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

gültig	ungültig	
❍	❍	Die Größe des Micro16-Programmspeichers beträgt 1024 Byte.
❍	❍	RAM steht für Random-Address Memory.
❍	❍	Bei einem Mealy-Schaltwerk hängt die Ausgangsfunktion vom aktuellen Zustand und den Eingängen ab.
❍	❍	Wenn beide Eingänge eines JK-Flip-Flops gleich 0 sind, erfolgt ein Toggeln der Ausgänge Q und $\neg Q$ synchron zum Takt.
❍	❍	Das MAR (Memory Address Register) ist 16 Bit groß und besitzt 0x7FFF als höchste Adresse.

Platz für Notizen: