

Universität Paderborn
Institut für *Elektrotechnik und Informationstechnik*
Fachgebiet *Datentechnik*
Prof. Sybille Hellebrand

Klausur
Grundlagen der technischen Informatik

15. März 2018

Punkteverteilung								
Aufgabe	1	2	3	4	5	Σ	BP	Σ
maximale Punkte	10	20	20	20	20	90	9	99
erreichte Punkte								

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

Aufkleber

Name:	
Matrikelnummer:	
Studienrichtung:	

Studium Generale: ☐ Ja
☐ Nein

Hinweise:

Für die Lösung der Klausuraufgaben sind ausschließlich die Aufgabenblätter zu verwenden. Lösungsangaben außerhalb der Aufgabenblätter („Schmierzettel“, etc.) werden bei der Bewertung nicht berücksichtigt!

Beschriften Sie jede Doppelseite mit Ihrer Matrikelnummer!

Mit Bleistift oder der Korrekturfarbe rot angefertigte Lösungen werden nicht bewertet!

Die Verwendung von „Tipp-Ex“ oder „Tintenkiller“ ist untersagt.

Es ist ein handgeschriebener DIN-A4 Zettel als Hilfsmittel zugelassen!

Es sind keine weiteren Hilfsmittel zugelassen!

Die Aufgabe 5 muss von Studierenden, welche die Veranstaltung als Studium Generale gewählt haben nicht bearbeitet werden!

Aufgabe 1: (Darstellung von Information)

10 Punkte

- a) Stellen Sie die Dezimalzahl 38_{10} als Binärzahl ohne Vorzeichen dar.

Kreuzen Sie die richtige Lösung an.

(2 Punkte)

- ☐ 100110_2
- ☐ 010111_2
- ☐ 1000110_2
- ☐ Keine Lösung ist richtig.

- b) Gegeben sei die Binärzahl $1010\ 1100\ 0110\ 1000_2$. Geben Sie die Zahl in Hexadezimaldarstellung an.

Kreuzen Sie die richtige Lösung an.

(2 Punkte)

- ☐ $AC68_{16}$
- ☐ $BC68_{16}$
- ☐ $AD68_{16}$
- ☐ Keine Lösung ist richtig.

- c) Die C Standardbibliothek `stdint.h` stellt den Datentypen `uint32_t` zur Verfügung. Zur Darstellung einer Zahl in diesem Datentypen wird eine 32-Bit Binärzahl ohne Vorzeichen verwendet. Welcher Wertebereich kann mit diesem Datentypen abgebildet werden?

Kreuzen Sie die richtige Lösung an.

(2 Punkte)

- ☐ $[-(2^{31} - 1), 2^{31} - 1]$
- ☐ $[-2^{31}, 2^{31} - 1]$
- ☐ $[0, 2^{32} - 1]$
- ☐ $[0, 2^{32}]$
- ☐ Keine Lösung ist richtig.

- d) Gegeben sei die Dezimalzahl $-8,5_{10}$. Geben Sie die Zahl in IEEE 754 Gleitkommaformat an.

Hinweis: Der Exponent wird in der Exzess 127 Darstellung dargestellt.

Kreuzen Sie die richtige Lösung an.

(2 Punkte)

☐ 1 10000010 100010000000000000000000₂

☐ 1 10000011 100010000000000000000000₂

☐ 1 00000011 000100000000000000000000₂

☐ 1 10000010 000100000000000000000000₂

☐ Keine Lösung ist richtig.

- e) Stellen Sie die Dezimalzahl -47_{10} als 8-Bit Binärzahl im 2er-Komplement dar.

Kreuzen Sie die richtige Lösung an.

(2 Punkte)

☐ 00101111_{2C}

☐ 11010001_{2C}

☐ 10101111_{2C}

☐ Keine Lösung ist richtig.

Aufgabe 2: (Logikoptimierung)

20 Punkte

- a) Vervollständigen Sie folgende Aussagen über den Algorithmus für die zweite Phase des Quine-McCluskey Verfahrens: (7,5 Punkte)

Phase II: Lösung des _____ problems

Gesucht wird eine kosten_____

_____ der Einsstellen mit _____

Vorgehen:

i) Reduktion _____

ii) Arbeiten mit Heuristiken

Für i) werden folgende drei Regeln so oft wie möglich angewendet:

Regel 1: Streiche aus der _____ alle

_____ und alle

_____, die davon überdeckt werden.

Regel 2: Streiche alle _____, die von

anderen, nicht teureren dominiert werden.

Regel 3: Entferne alle _____, die einen

anderen _____ dominieren.

- b) Gegeben sei die boolesche Funktion f mit den vier Parametern a , b , c und d . f ist die Summe der Minterme $m_2, m_5, m_6, m_7, m_9, m_{10}, m_{11}, m_{12}, m_{13}, m_{14}$ und m_{15} . Füllen Sie die Wahrheitstabelle für f aus.

$$f = m_2 + m_5 + m_6 + m_7 + m_9 + m_{10} + m_{11} + m_{12} + m_{13} + m_{14} + m_{15}$$

Hinweis: Die Einträge in der Tabelle können als Binärzahlen interpretiert werden. Wenn der Eintrag den Wert i hat, wird der entsprechende Minterm abkürzend mit m_i bezeichnet. Die Zählung startet demnach bei m_0 . (2,5 Punkte)

a	b	c	d	f
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	

Tabelle 1: Wahrheitstafel für f

- c) Führt man die erste Phase des Algorithmus von Quine-McCluskey für die Tabelle in b) aus, so liefert dieser die folgenden Implikanten:

(7 Punkte)

$$--10, -1-1, -11-, 1--1, 1-1-, 11--$$

Führen Sie hiermit nun die zweite Phase des Algorithmus aus.

Überdeckungstabelle 1:

P	m_{--}	m_{-1}	m_{-11}	m_{1-}	m_{1-1}	m_{11-}	m_{--1}	m_{-1-}	m_{-1-1}	m_{1--}	m_{1-1-}	m_{11--}

Geben Sie die im ersten Schritt gefundenen wesentlichen Primimplikanten an:

{_____}

Überdeckungstabelle 2:

P	m_{--}	m_{-1}	m_{-11}	m_{1-}	m_{1-1}	m_{11-}	m_{--1}	m_{-1-}	m_{-1-1}	m_{1--}	m_{1-1-}	m_{11--}

Geben Sie die minimale SOP-Form als Menge von Primimplikanten an:

{_____}.

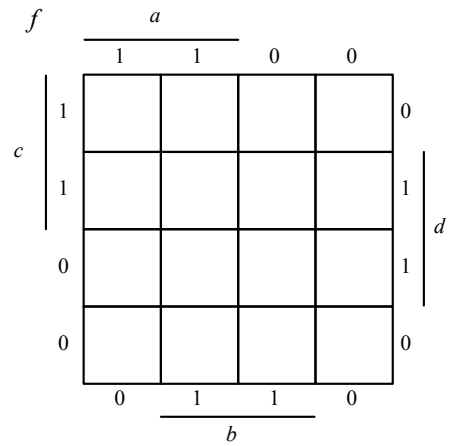
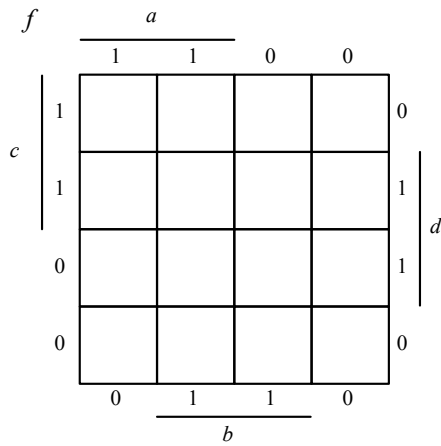
P	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$

Tabelle 2: Ersatz Überdeckungstabelle 1, **ungültige Lösung streichen!**

P	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$	$m_{\text{---}}$

Tabelle 3: Ersatz Überdeckungstabelle 2, **ungültige Lösung streichen!**

- d) Finden Sie für die Funktion f aus b) die minimale SOP-Form nun mit Hilfe des untenstehenden Karnaugh-Diagramms: (3 Punkte)



Ersatzdiagramm, ungültige Lösung streichen!

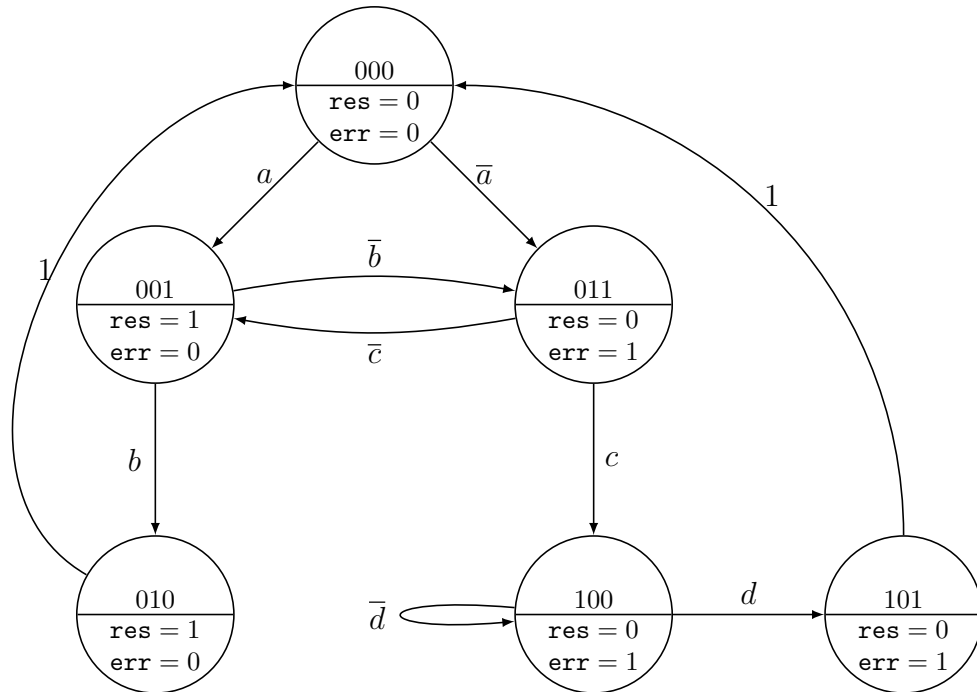
Minimale SOP-Form für f :

{_____}.

Aufgabe 3: (Steuerwerke)

20 Punkte

Der Automat in Abbildung 1 soll als Mikroprogramm mit Zähler realisiert werden.

Abbildung 1: Automat des μ -Programms

- a) Definieren Sie das μ -Befehlsformat und geben Sie an, wie viele Bits pro Speicherzeile benötigt werden. (5 Punkte)

- b) Wie viele Flip-Flops werden für den Zähler benötigt? (3 Punkte)

Wie groß wird der Speicher insgesamt?

Welche weiteren Optimierungsmöglichkeiten gibt es?

- c) Ergänzen Sie die in Abbildung 2 gegebene Struktur des μ -programmierbaren Steuerwerks und geben Sie die Speicherbelegung des μ -Programms an. (12 Punkte)

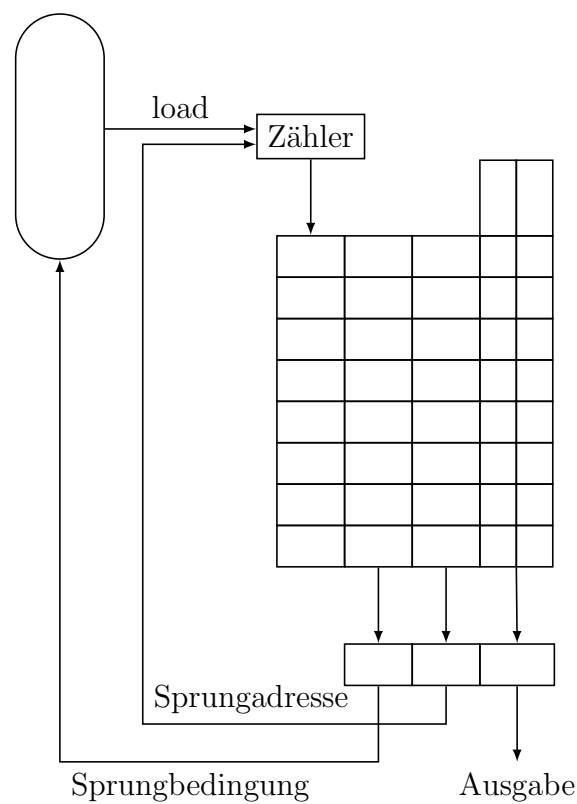


Abbildung 2: Vorlage für Lösung

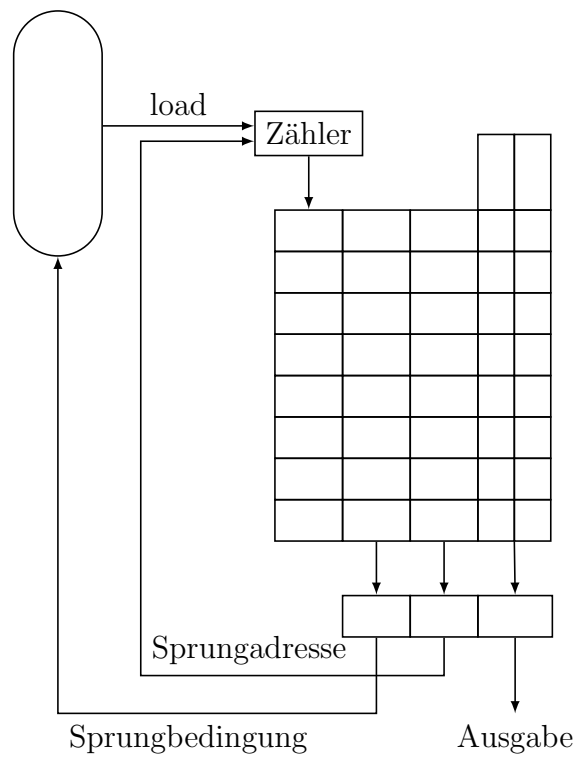


Abbildung 3: Ersatzvorlage für Lösung, **ungültige Lösung streichen!**

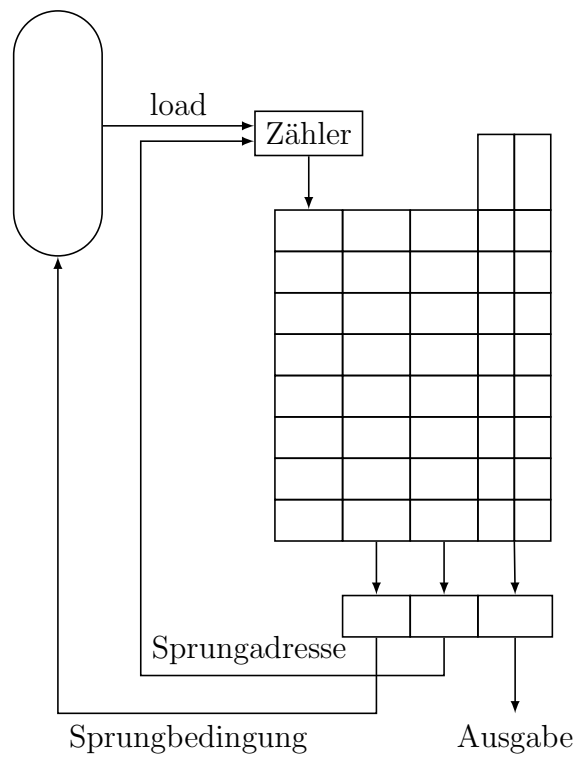


Abbildung 4: Ersatzvorlage für Lösung, **ungültige Lösung streichen!**

Abbildung 5: Schema zur Division **ohne** Rückspeichern

[illegible]

Abbildung 6: Ersatzschema, **ungültige Lösung streichen!**

- b) Abbildung 8 zeigt den Datenpfad und die Steuerung eines sequentiellen Dividierers (ohne Rückspeichern) für n -bit Zahlen. Über den INBUS sollen nacheinander die beiden Zahlen a und b geschickt und in Register A und B geschrieben werden. Das ganzzahlige Ergebnis der Division a/b soll anschließend in Register A stehen, während Register P den Rest der Division enthält. Beide Registerinhalte (A und P) sollen nach der Berechnung nacheinander auf den OUTBUS gelegt werden. Die notwendigen Kontrollpunkte sind mit c1-c9 beschriftet. Beschreiben Sie die Funktion jedes Kontrollpunktes kurz in der nachfolgenden Tabelle.

Hinweis: Das Signal $count_{n-1}$ ist 1, wenn der Zähler $n - 1$ erreicht, sonst 0

(3,5 Punkte)

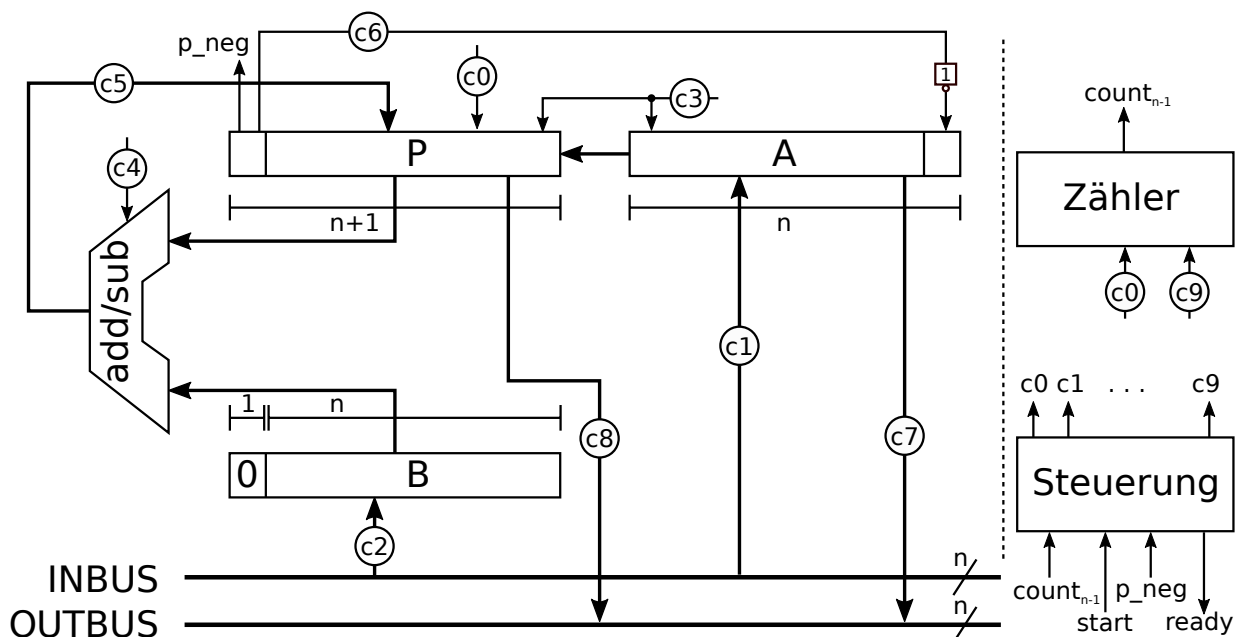


Abbildung 7: Datenpfad eines Dividierers mit Steuerung

Kontrollpunkte:

c0: Register P und Zähler wird mit 0 initialisiert

c1: _____

c2: _____

c3: _____

c4: _____

c5: _____

c6: _____

c7: _____

c8: _____

c9: _____

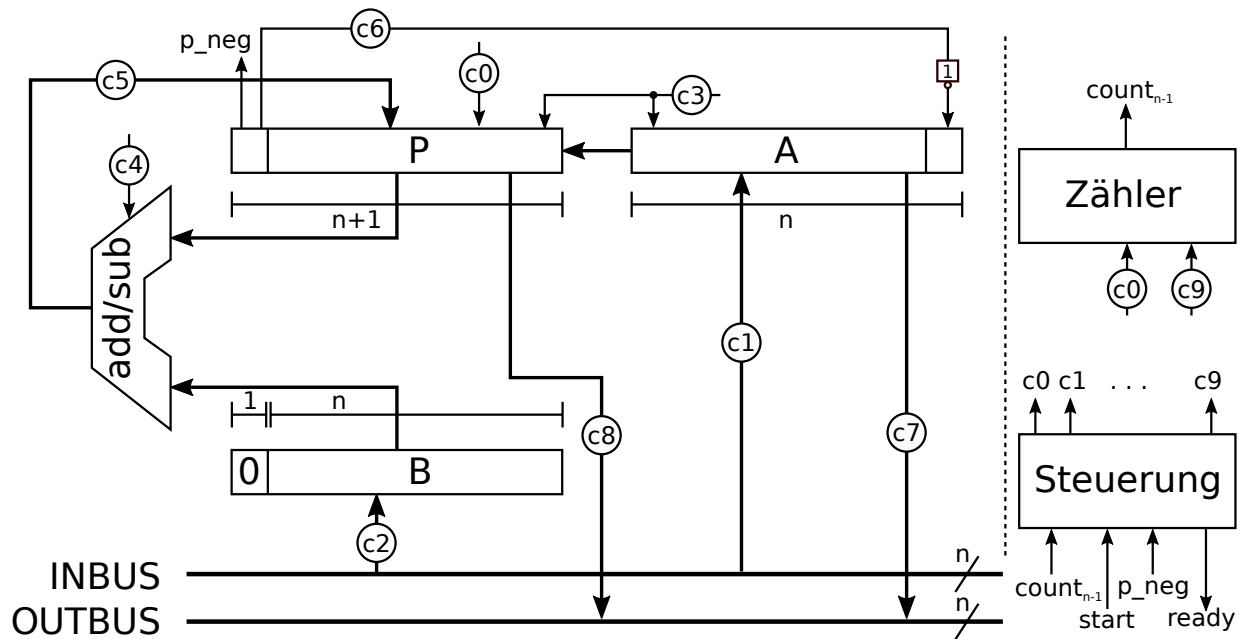


Abbildung 8: Datenpfad eines Dividierers mit Steuerung

Ersatztable für Kontrollpunkte, **ungültige Lösung streichen!**

c0: Register P und Zähler wird mit 0 initialisiert

c1: _____

c2: _____

c3: _____

c4: _____

c5: _____

c6: _____

c7: _____

c8: _____

c9: _____

- c) Abbildung 9 zeigt nun die Steuerung für den sequentiellen Dividierer aus Aufgabenteil b) in Form eines **Moore**-Automaten. Ergänzen Sie den Automaten um die fehlenden Zustandsübergangsbedingungen (p_neg , $count_{n-1}$) und die zugehörigen Steuersignale (c1-c9) aus Aufgabenteil b).

Hinweis: Ausgangssignale, die auf 0 gesetzt sind ($c_i = 0$) können weggelassen werden. (6,5 Punkte)

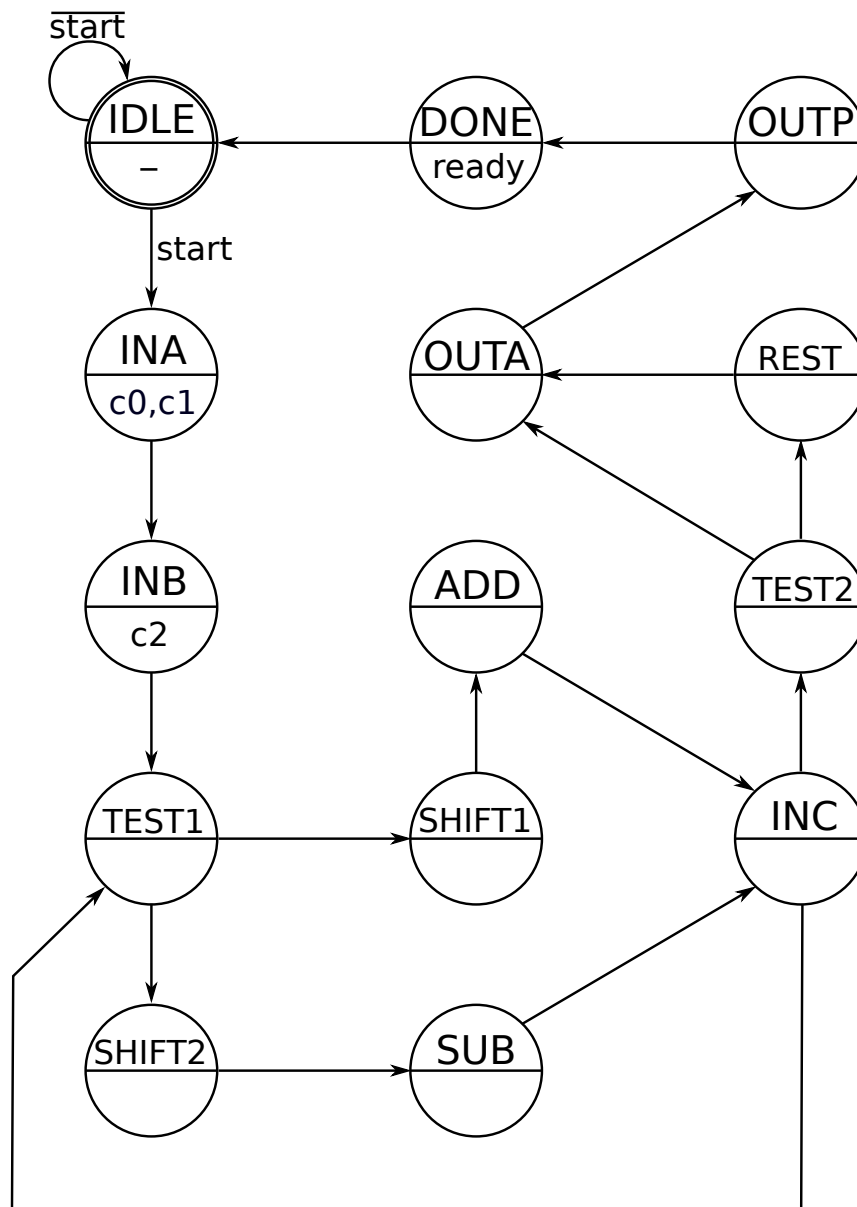
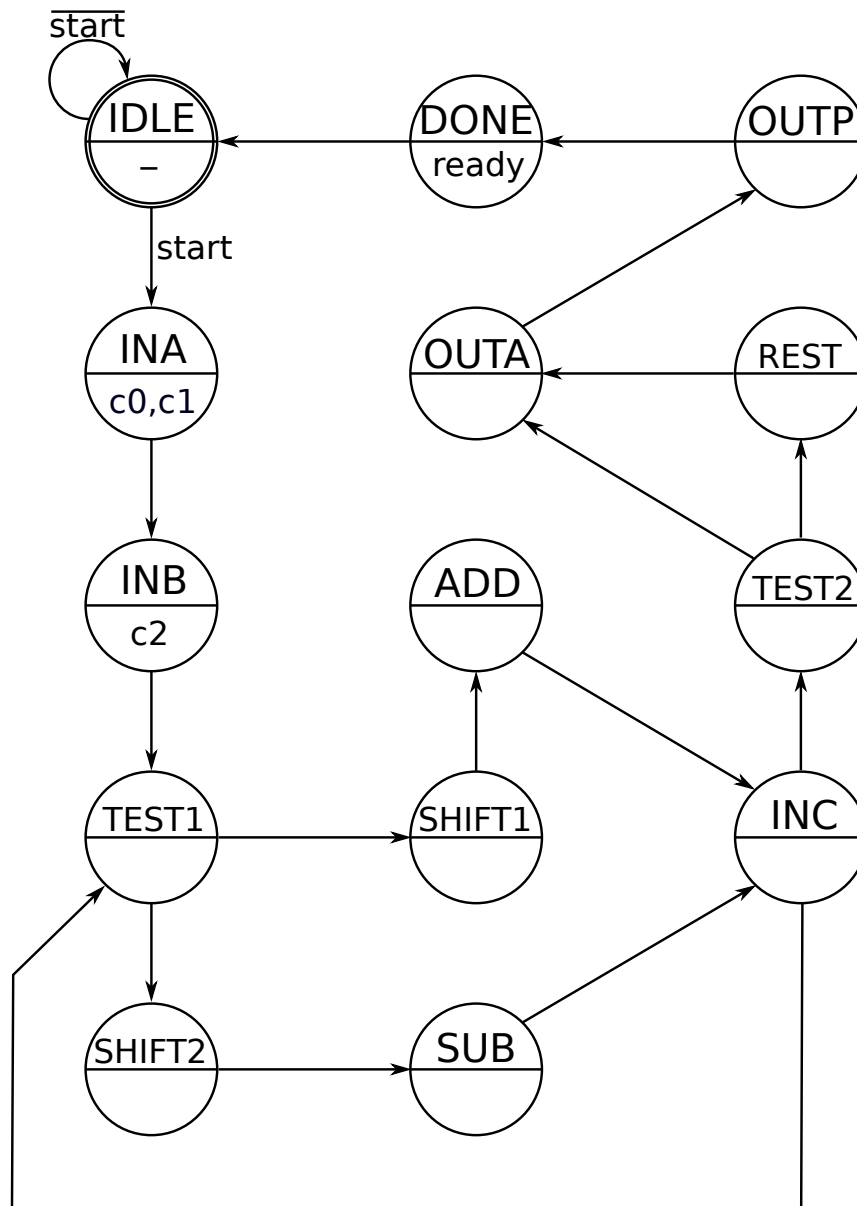


Abbildung 9: **Moore**-Automat einer Dividierer-Steuerung

Abbildung 10: Ersatzautomat, **ungültige Lösung streichen!**

Aufgabe 5: (VHDL)

20 Punkte

Gegeben sei nachfolgende Schaltung:

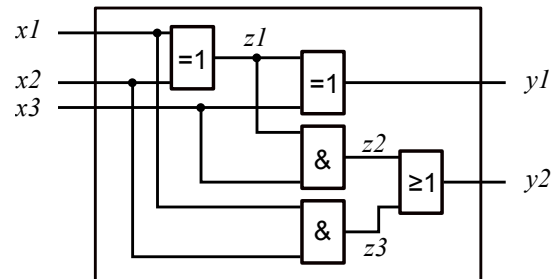


Abbildung 11: Zu realisierende Schaltung

- a) Realisieren Sie die in Abbildung 11 dargestellte Schaltung in VHDL. (5 Punkte)

```

library IEEE;
use IEEE.STD.LOGIC.1164.ALL;

```

```

entity schaltung is

```

```

end schaltung;
architecture Structural of schaltung is

```

```

begin

```

```

end Structural;

```

Die Eingangssignale der Schaltung in Abbildung 11 haben nachfolgende Verläufe:

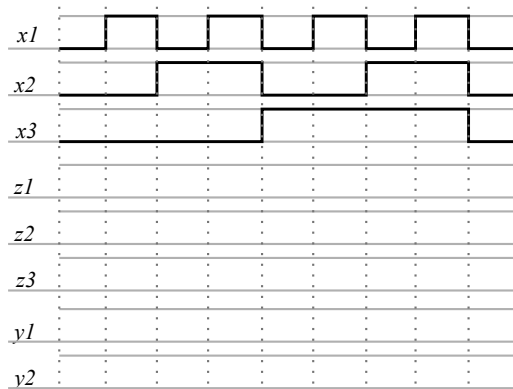


Abbildung 12: Signalverläufe

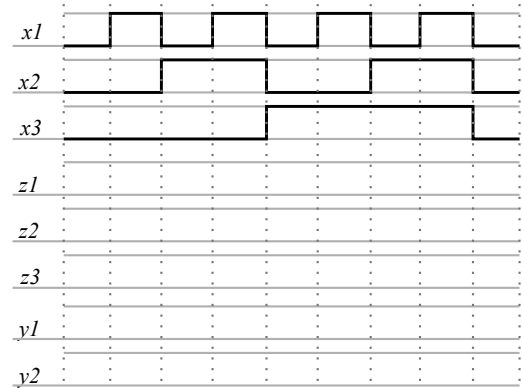


Abbildung 13: Ersatz,
ungültige Lösung streichen!

- b) Ergänzen Sie die Verläufe der Signale $z1$, $z2$, $z3$, $y1$ und $y2$ in der Abbildung 12.

Hinweis: Gehen Sie von idealen Bauteilen aus.

(5 Punkte)

- c) Was macht diese Schaltung?

(3 Punkte)

- d) Gegeben sei ein Volladdierer mit folgender Entity:

```
entity fulladder is
port ( a, b, cin : IN      STD_LOGIC;
       s, cout   : OUT     STD_LOGIC );
end fulladder;
```

Nutzen Sie den oben genannten Volladdierer als Komponente, um einen 3-Bit Ripple-Carry-Addierer zu entwickeln.

(7 Punkte)

Codefragment zu Aufgabenteil d)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity rcadder is
    PORT (
        a, b      : IN      STD_LOGIC_VECTOR(2 downto 0);
        cin       : IN      STD_LOGIC;
        sum       : OUT     STD_LOGIC_VECTOR(2 downto 0);
        cout      : OUT     STD_LOGIC);
end rcadder;

```

Ersatz, ungültige Lösung streichen!

