

Klausur zur Vorlesung

Grundlagen der Technischen Informatik / Digitaltechnik (GTI/DT)

Prof. Marco Platzner
Fachgebiet Technische Informatik
Universität Paderborn

07.08.2010

- Die Bearbeitungsdauer beträgt für alle Studenten **90 Minuten**. Es sind **alle 5 Aufgaben** zu bearbeiten.
- Es sind **keine Hilfsmittel** zugelassen.
- Schreiben Sie **nicht mit Bleistift oder Rotstift**.
- Verwenden Sie **kein eigenes Papier**. Bei Bedarf bekommen Sie Papier bei der Klausuraufsicht.
- Schreiben Sie auf jedes Blatt (auch auf das Konzeptpapier) in Blockschrift Ihren Namen und Ihre Matrikelnummer.
- **Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet!** Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch! Verwenden Sie kein Tipp-Ex.
- Abschreiben und abschreiben lassen oder Hilfe Dritter führt zum Nichtbestehen der Klausur.

Nachname: _____

Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Aufkleber

Aufgabe	1	2	3	4	5	Σ
Punkte	15	20	15	20	20	90
Erreicht						

Aufgabe 1 (Multiple Choice)

[15 Punkte]

Bei den folgenden Fragen können keine, eine oder mehrere Antworten richtig sein. Kreuzen Sie die richtigen Antworten deutlich an.

(a) Wenn E ein minimaler SOP-Ausdruck für eine Funktion z ist, dann:

- ☐ ist jeder Produktterm von E ein Primimplikant von z
- ☐ ist E eine irredundante Summe
- ☐ besteht E nur aus essentiellen Primimplikanten
- ☐ besteht E nur aus Mintermen

(b) Jeder Moore-Automat ...

- ☐ hat eine endliche Zustandsmenge
- ☐ hat eine minimale Anzahl von Zuständen
- ☐ ist vollreduziert
- ☐ lässt sich mit einer One-hot Zustandskodierung implementieren

(c) Welche Zahlendarstellung hat einen symmetrischen Wertebereich?

- ☐ Vorzeichen/Betrag
- ☐ Exzess- m Code
- ☐ 1-er Komplement
- ☐ normalisierte Darstellung im IEEE 754 Format

NAME:

Matrikelnummer:

(d) Wie viele Bitfehler lassen sich mit einem Code mit Hammingdistanz 5 korrigieren?

☐ 2

☐ 3

☐ 4

☐ 5

(e) Wie bezeichnet man die Mikroprogramm-Technik, bei der für jedes benötigte Steuersignal ein Bit in der Mikroinstruktion vorgesehen wird?

☐ horizontale Mikroprogrammierung

☐ diagonale Mikroprogrammierung

☐ vertikale Mikroprogrammierung

Aufgabe 2 (Spezifikation von Automaten mit VHDL)

[20 Punkte]

- (a) Auf der nächsten Seite finden Sie die VHDL Spezifikation des endlichen Automaten FSM. Die Architektur IMP der Entity FSM beinhaltet u.a. die Ports X und Y sowie die lokalen Signale N und S . Tragen Sie in die folgenden Tabelle ein, welche Aussagen für die angegebenen Signale wahr sind. Nehmen Sie an, dass sich Eingangssignale (bis auf das Clock-Signal Clk) jederzeit ändern können.

Das Signal kann sich...	N	S	X	Y
nur bei steigender Taktflanke ändern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sofort bei einem Reset ändern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
jederzeit ändern	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- (b) Vervollständigen Sie den Automatengraph für die Entity FSM in Abbildung 1. Der Automat hat genau drei Zustände und ist nach einem Reset im Zustand "00". Hinweis: Der *when others* Fall muss nicht mit abgebildet werden.

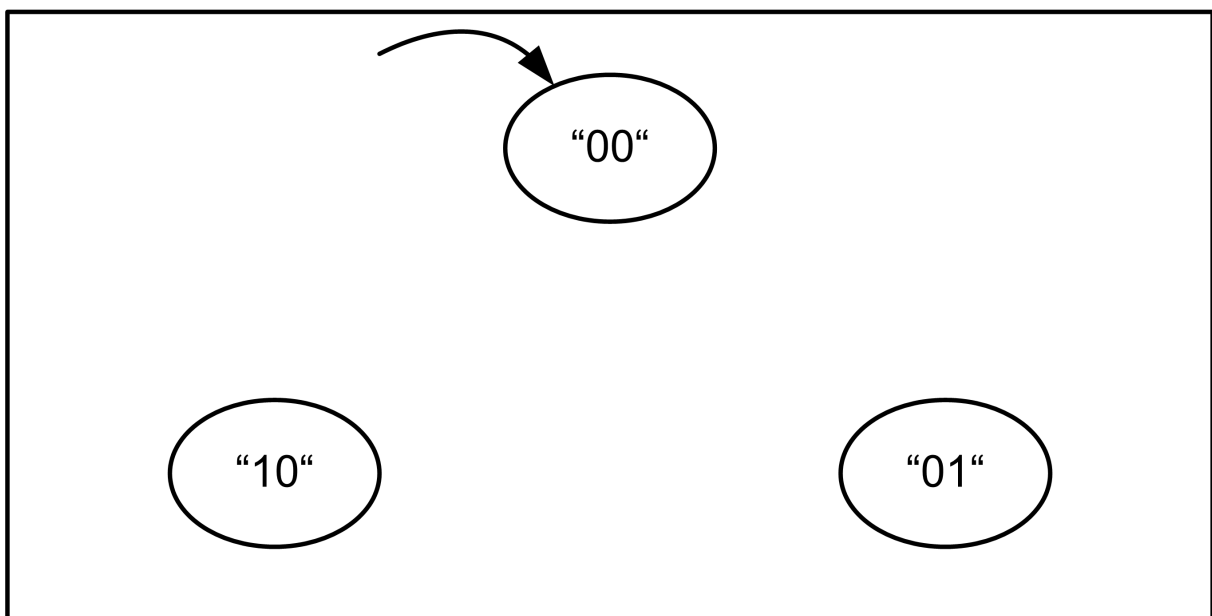


Abbildung 1: Automatengraph

Entity fsm

```
entity FSM is
  port (
    Clk, Rst, X    : in  std_logic;
    Y              : out std_logic
  );
end entity FSM;

architecture IMP of FSM is

  signal S, N : std_logic_vector(1 downto 0);

begin
  P0_PROC: process (Clk, Rst)
  begin
    if (Clk'event and Clk = '1') then
      if (Rst = '1') then
        S <= "00";
      else
        S <= N;
      end if
    end if;
  end process;

  P1_PROC: process (S, X)
  begin
    case S is
      when "00" =>
        if (X = '1') then
          Y <= '0';
          N <= "10";
        else
          Y <= '1';
          N <= "01";
        end if;
      when "01" =>
        Y <= '1';
        if (X = '1') then
          N <= "10";
        else
          N <= "01";
        end if;
      when "10" =>
        N <= "00";
        if (X = '1') then
          Y <= '0';
        else
          Y <= '1';
        end if;
      when others =>
        N <= "00";
      end case;
    end process;
  end architecture IMP;
```

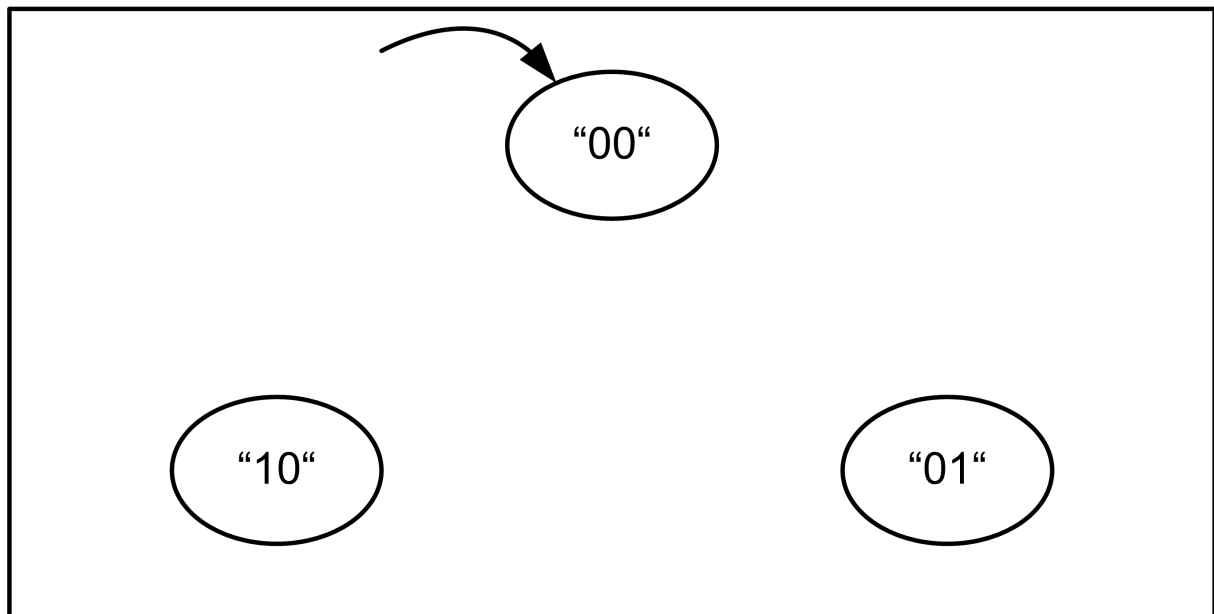


Abbildung 2: Automatengraph (Ersatz)

- (c) Die Abbildung 3 zeigt einige logische Gatter sowie zwei D-Flip-Flops. Verbinden Sie die Signale *S*, *N*, *Rst* und *Clk* so, dass die Schaltung die Spezifikation des Prozesses *P0_PROC* realisiert. Sie benötigen dafür nicht alle angegebenen Schaltungselemente. Falls Sie Konstanten benötigen, können Sie eine logische Null von dem Signal '0' abgreifen.

Prozess P0_PROC

```

P0_PROC: process (Clk , Rst)
begin
  if (Clk 'event and Clk = '1') then
    if (Rst = '1') then
      S <= "00";
    else
      S <= N;
    end if
  end if;
end process;
  
```

NAME:

Matrikelnummer:

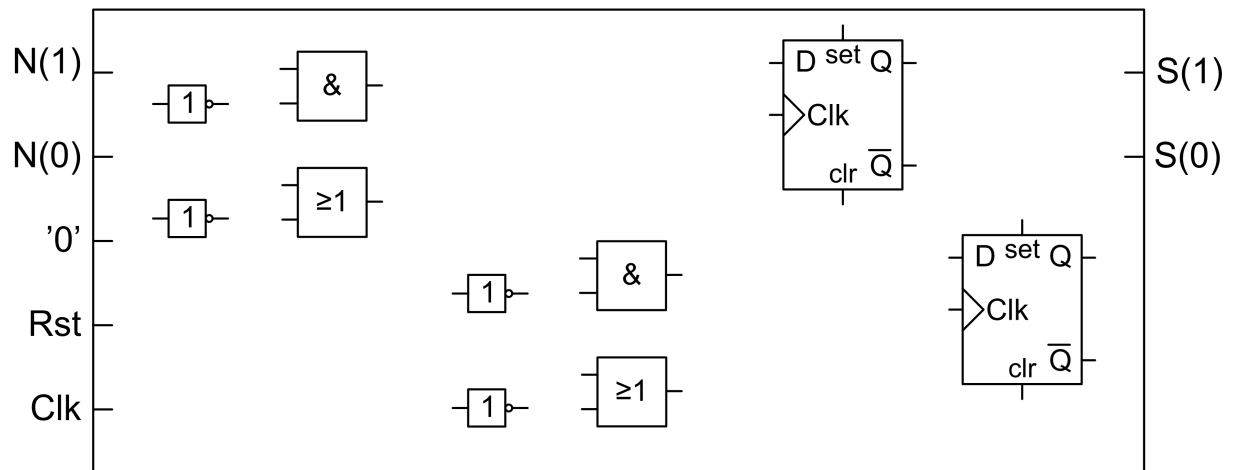


Abbildung 3: Schaltung für den Prozess P0_PROC

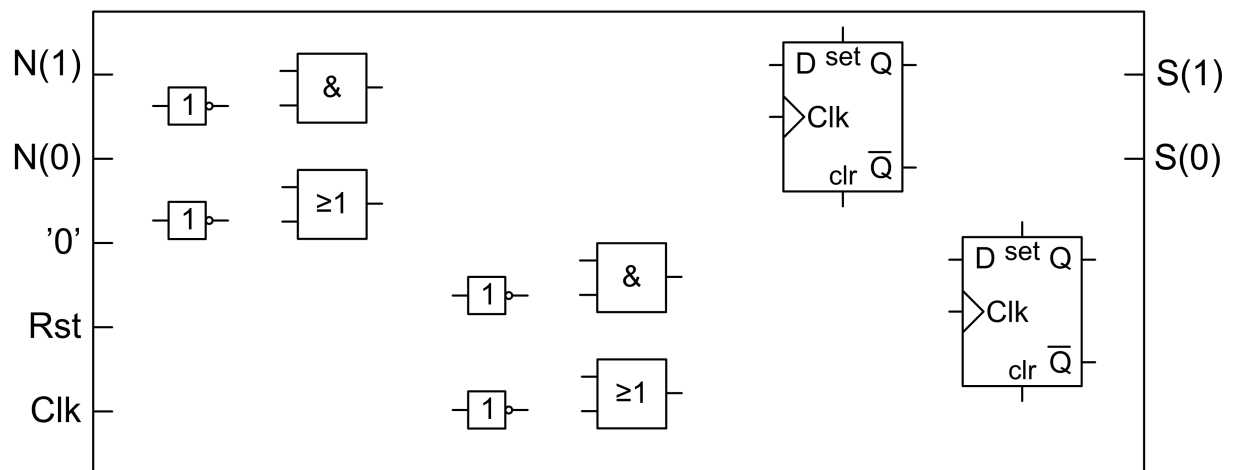


Abbildung 4: Schaltung für den Prozess P0_PROC (Ersatz)

Aufgabe 3 (Zustandsminimierung Prioritäts-Binärencoder) [15 Punkte]

Abbildung 5 zeigt einen Prioritäts-Binärencoder mit 3 Eingängen $x_2..x_0$, 2 Ausgängen $z_1..z_0$, einer „Enable“-Eingangsleitung EN und einer „Input Active“-Ausgangsleitung IA .

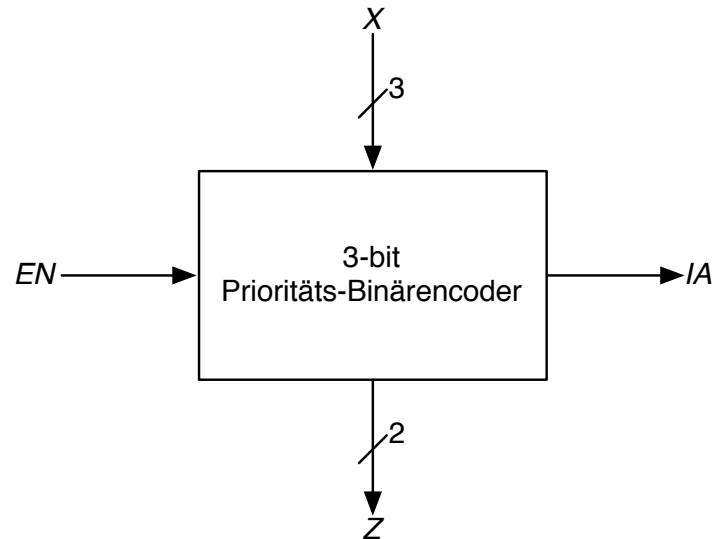


Abbildung 5: 3-bit Prioritäts-Binärencoder

Dieser 3-bit Prioritäts-Binärencoder hat ...

- ein 3-bit Wort X als Eingang, bei dem ein oder mehrere Bits x_i auf 1 gesetzt sein können.
- ein 2-bit Wort Z als Ausgang, das den größten Index i für $x_i = 1$ binär kodiert. D. h. die Eingänge haben Prioritäten, x_2 die höchste, x_0 die niedrigste.
- einen zusätzlichen Ausgang IA , um zu erkennen, dass alle Eingänge 0 sind.
 - $IA = \begin{cases} 0 & \text{wenn } \forall i \in \{0..2\} : x_i = 0 \\ 1 & \text{sonst} \end{cases}$
- einen zusätzlichen Eingang EN , um den Binärencoder zu aktivieren.
 - $EN = 0 \Rightarrow IA = 0 \wedge \forall i \in \{0..1\} : z_i = 0$

Entwerfen Sie im Folgenden für den in Abbildung 5 angegebenen 3-bit Prioritäts-Binärencoder eine vereinfachte Schaltung.

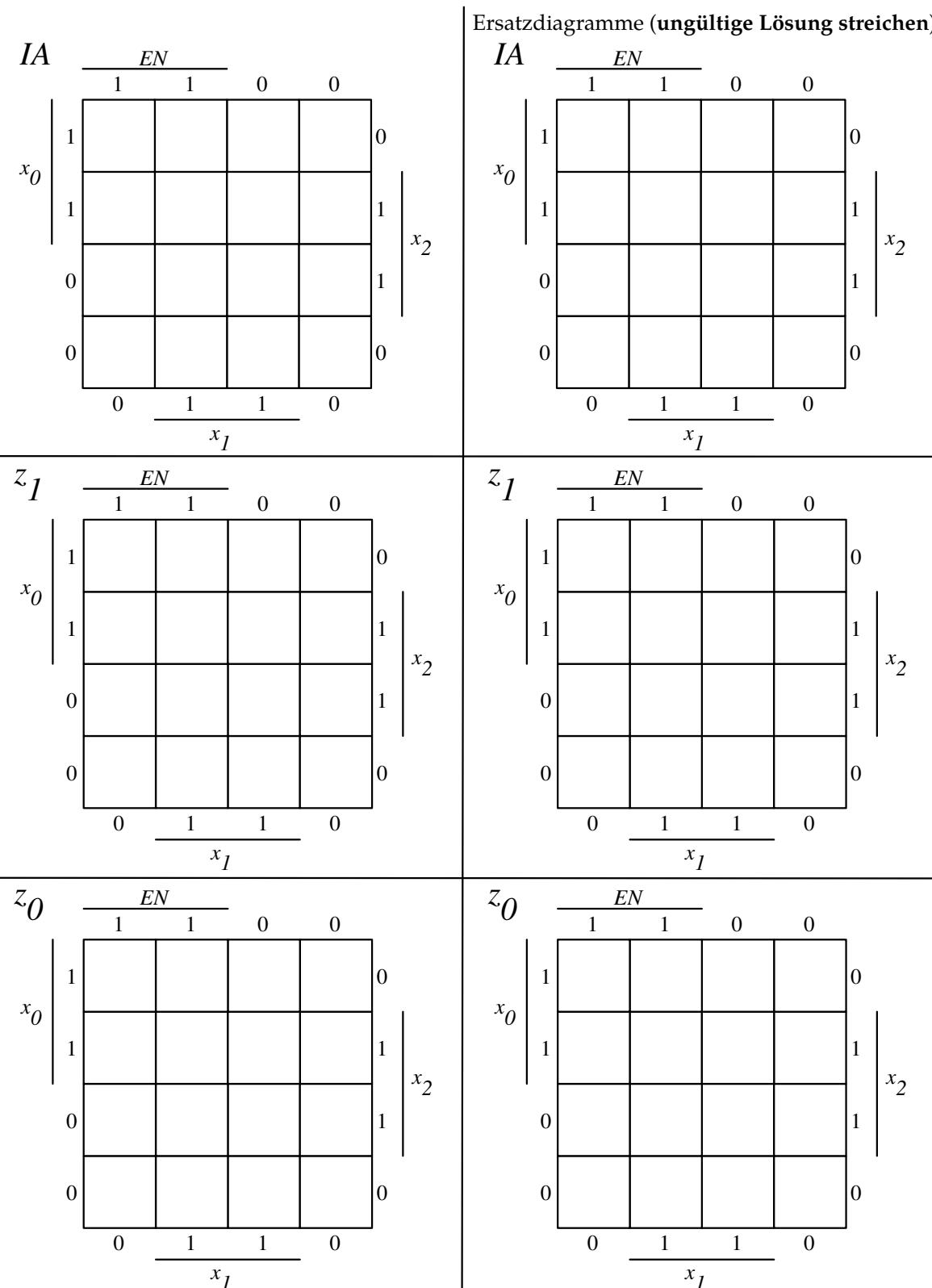
NAME:

Matrikelnummer:

- (a) Füllen Sie die angegebene Wahrheitstabelle mit den bool'schen Werten für den 3-bit Prioritäts-Binärencoder. Verwenden Sie „Don't Cares“ (X) in den Fällen, in denen es möglich ist.

EN	Eingang			Ausgang		
	x_2	x_1	x_0	IA	z_1	z_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			

- (b) Bestimmen Sie die minimalen SOP-Formen für die einzelnen Ausgaben IA , z_1 und z_0 mit Hilfe der unten angegebenen Karnaugh-Diagramme.



NAME:

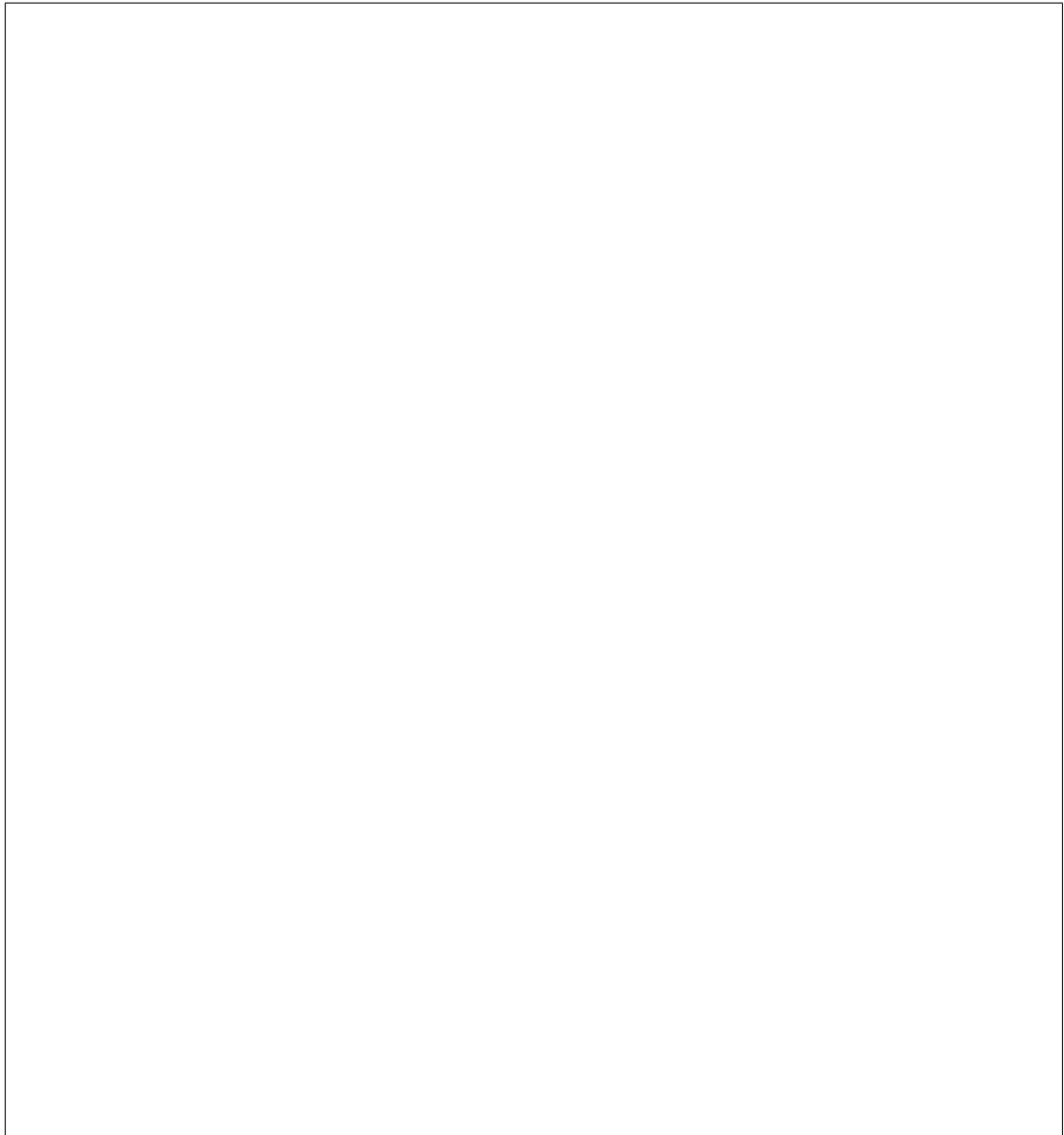
Matrikelnummer:

$IA =$ _____

$z_1 =$ _____

$z_0 =$ _____

- (c) Entwerfen Sie eine Schaltung für den 3-bit Prioritäts-Binärencoder. Verwenden Sie **ausschließlich** 2-AND (AND mit 2 Eingängen), 2-OR (OR mit 2 Eingängen) und NOT Gatter. Ihr Entwurf soll optimal sein, d. h. Sie sollten die **minimale** Anzahl an Gattern benutzen!



(d) Nehmen Sie an, dass Sie diesen 3-bit Prioritäts-Binärencoder in eine Schaltung einbauen, bei der Folgendes gilt:

- Am Eingang $x_2 \dots x_0$ liegt immer mindestens eine 1 an.
- Der Eingang EN liegt immer auf 1.

Wie ändert sich die Wertetabelle für den 3-bit Prioritäts-Binärencoder? Verwenden Sie „Don't Cares“ (X) in den Fällen, in denen es möglich ist.

EN	Eingang			Ausgang		
	x_2	x_1	x_0	IA	z_1	z_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			

NAME:

Matrikelnummer:

- (e) Bestimmen Sie die minimalen SOP-Formen für die einzelnen Ausgaben IA , z_1 und z_0 mit Hilfe der unten angegebenen Karnaugh-Diagramme für den unter (d) angegebenen Fall.

IA <div style="display: inline-block; text-align: center; margin-left: 20px;"> $\overbrace{\quad\quad\quad}^{EN}$ 1 1 0 0 </div>		Ersatzdiagramme (ungültige Lösung streichen)																															
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> x_0 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> x_0 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> z_1 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> z_1 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> z_0 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> z_0 1 1 0 0 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table> <div style="margin-left: 10px;"> 0 1 1 0 </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div></div> <div> $\overbrace{\quad\quad\quad}^{x_I}$ 0 1 1 0 </div> <div></div> </div>																

$$IA = \underline{\hspace{15cm}}$$

$$z_1 = \underline{\hspace{15cm}}$$

$$z_0 = \underline{\hspace{15cm}}$$

NAME:

Matrikelnummer:

Aufgabe 4 (Division - Array-Dividierer)

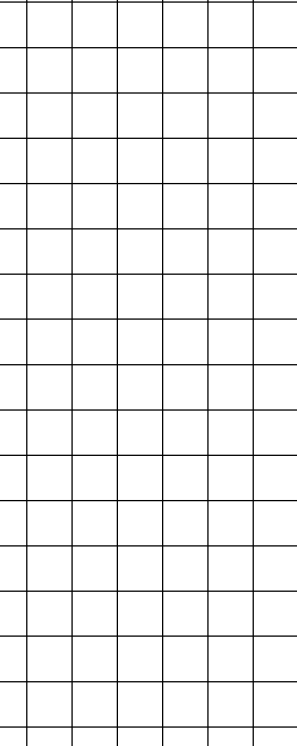
[20 Punkte]

- (a) Wandeln Sie die Zahlen 77_{10} und 6_{10} in das duale Zahlensystem um.

$77_{10} \leftrightarrow$ _____

$6_{10} \leftrightarrow$ _____

- (b) Führen Sie die schriftliche Division der Zahlen 77_{10} und 6_{10} ($77/6$) im dualen Zahlensystem durch. Benutzen Sie dafür das Schema in Abbildung 4. Beenden Sie Ihre Berechnung, wenn der Rest nicht mehr durch den Divisor ganzzahlig zu teilen ist. Schreiben Sie Quotient und Rest in die dafür vorgesehenen Felder. Überprüfen Sie das Ergebnis im Dezimalsystem.



/=

}

Quotient

}

Rest

Abbildung 6: Divisionsschema

NAME:

Matrikelnummer:

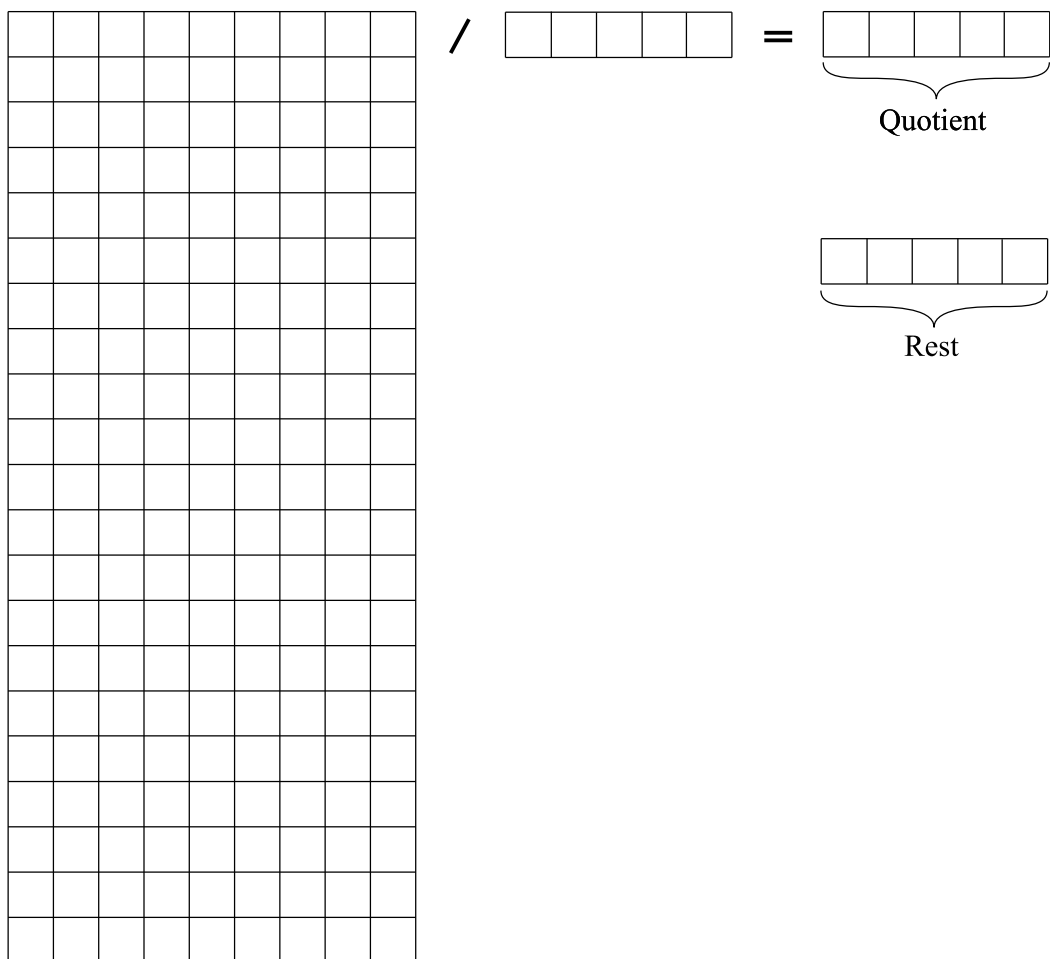


Abbildung 7: Divisionsschema (Ersatzdiagramm)

Eine Division kann in Hardware durch einen Array-Dividierer realisiert werden. Dazu wird der Vollsubtrahierer aus Abbildung 8 mit einem zusätzlichen Eingangssignal a versehen (vgl. Abb. 9).

- Bei einer '1' am Eingangssignal a ist der Ausgang d gleich dem Minuend x .
- Bei einer '0' an a ist $d = x - y - b_{i-1}$.
- Das Übertragsbit bzw. Borrow Bit b_i wird durch a nicht beeinflusst.

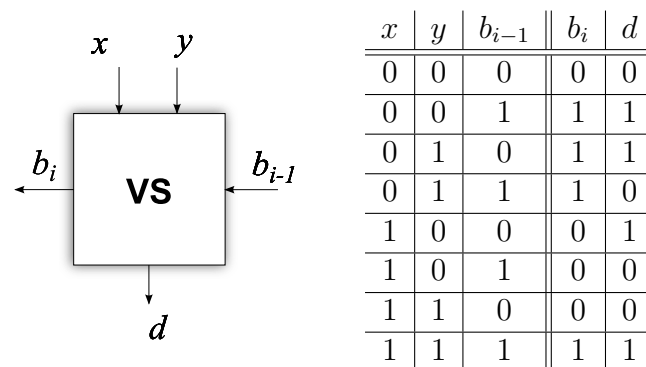


Abbildung 8: Vollsubtrahierer mit Wahrheitstabelle

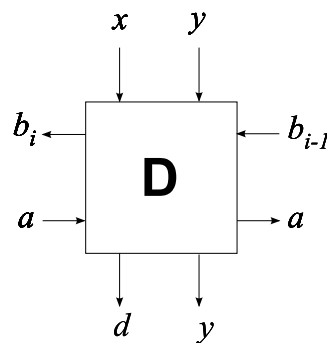


Abbildung 9: modifizierter Vollsubtrahierer

- (c) Vervollständigen Sie die Wahrheitstabelle des modifizierten Vollsubtrahierers in Tabelle 1.
- (d) Führen Sie die Division $x/y = q + (r/y)$ für $x = 7$ und $y = 3$ mit einem Array-Dividierer (Abb. 4) durch.
- Schreiben Sie die benötigten Namen ($x_2, x_1, x_0, y_2, y_1, y_0, q_2, q_1, q_0, r_2, r_1, r_0$) der Ein- und Ausgänge in die grauen Quadrate.
 - Legen Sie den Dividend und den Divisor korrekt an den Array-Dividierer an.
 - Tragen Sie in die grauen Kreise alle Ergebniswerte und Konstanten (0, 1) ein.

NAME:

Matrikelnummer:

x	y	b_{i-1}	a	b_i	d
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		

x	y	b_{i-1}	a	b_i	d
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: Wahrheitstabelle und Ersatz des modifizierten Subtrahierers. Ungültige Lösung streichen!

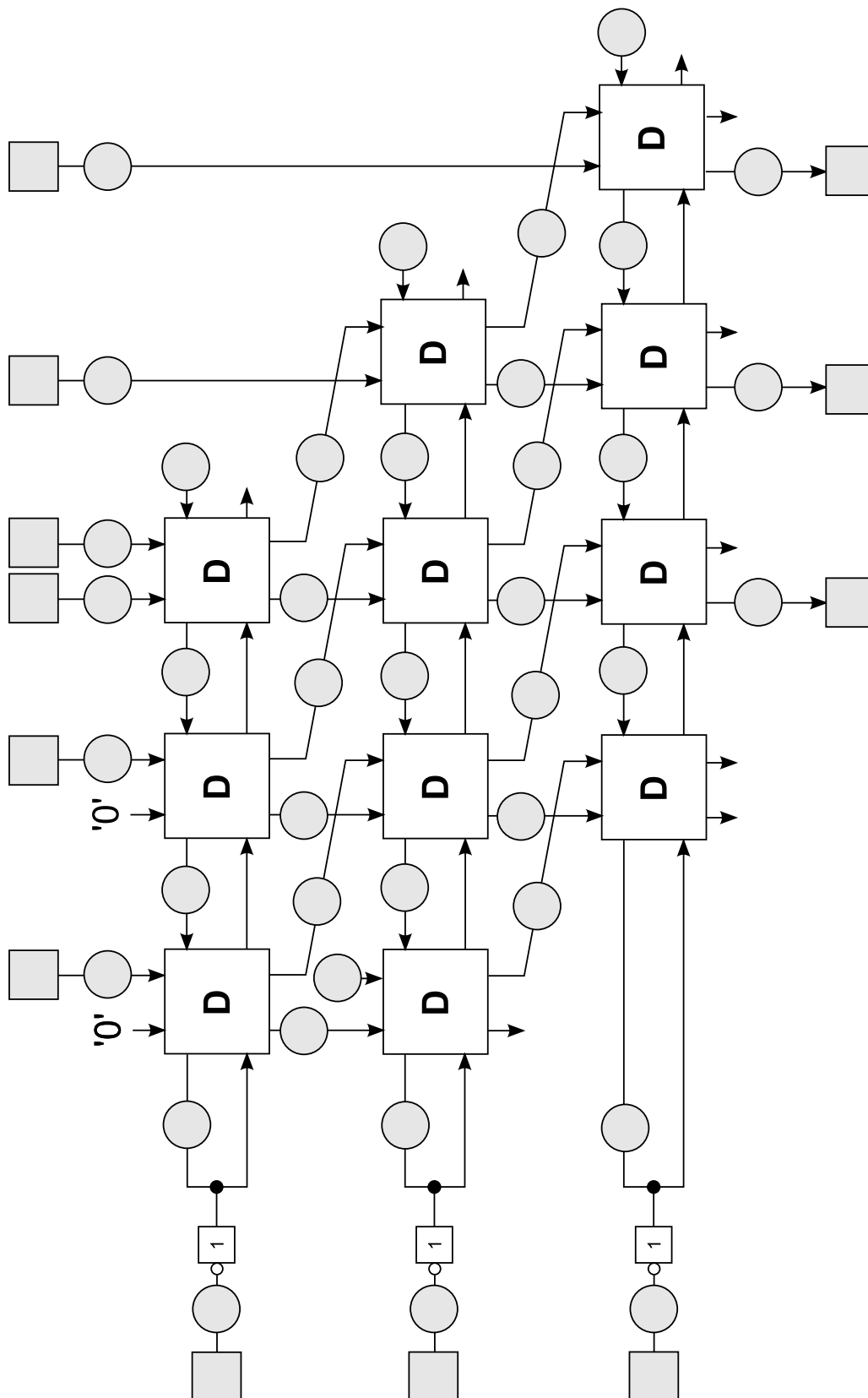


Abbildung 10: 3-Bit Array-Dividierer

NAME:

Matrikelnummer:

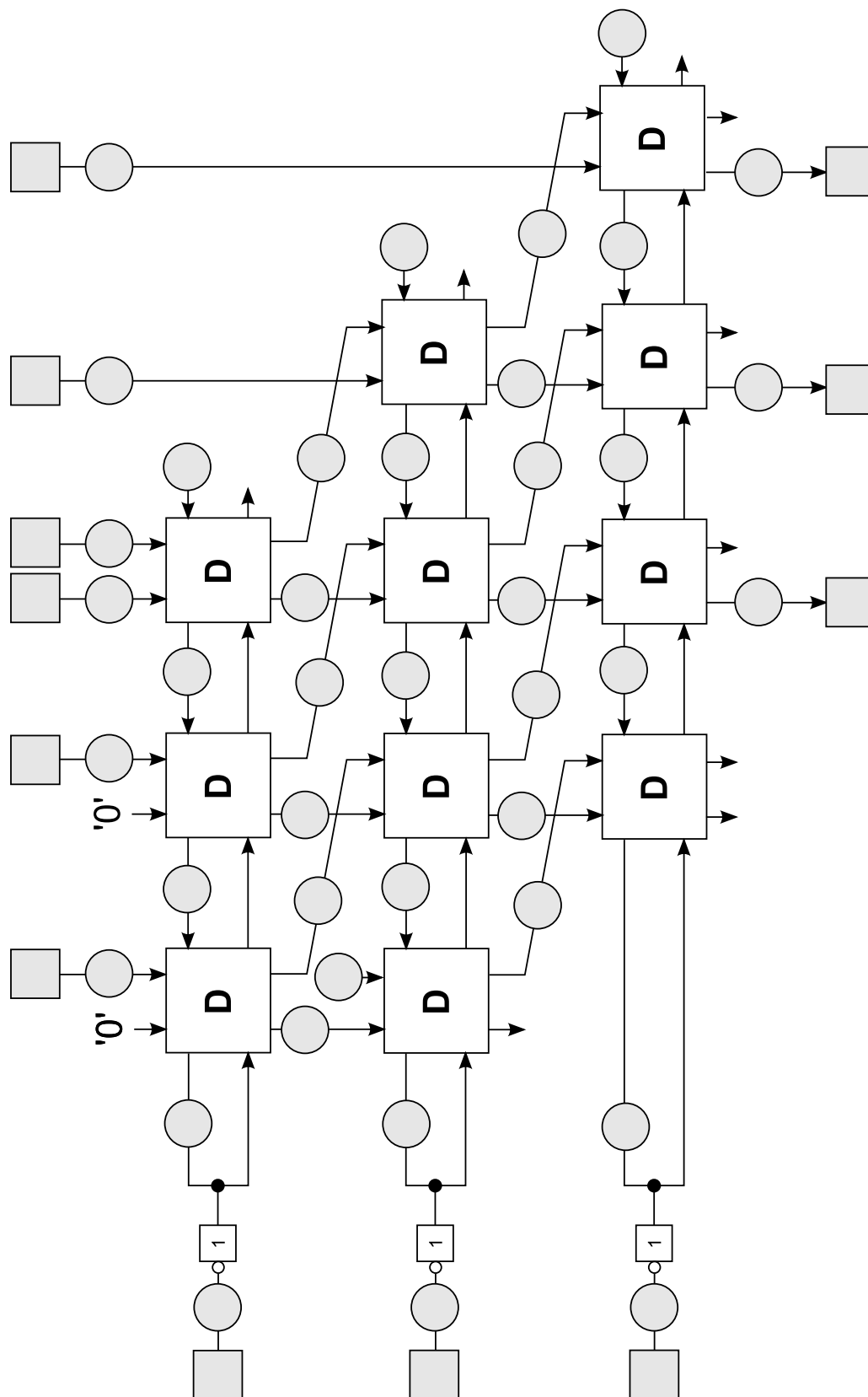


Abbildung 11: 3-Bit Array-Dividierer (Ersatzdiagramm)

Aufgabe 5 (RTL Entwurf)

[20 Punkte]

Abbildung 12 zeigt einen Schaltungsblock der zählt, wie oft ein bestimmtes Eingabewort im internen ROM vorhanden ist. Die Schaltung besitzt drei Eingänge: **X**, **clk**, **Start** und zwei Ausgänge: **Done**, **Y**.

- Am Eingang **X** wird das zu verarbeitende Wort (32-Bit) über den INBUS angelegt.
- Das **Start** Signal signalisiert, dass die Zählung beginnen soll.
- Der Ausgang **Done** signalisiert das Ende der Berechnung sowie ein gültiges Ergebnis **Y** (Anzahl der mit dem Eingabewort übereinstimmenden Worte im ROM).

Um die Anzahl übereinstimmender Worte zu ermitteln, vergleicht die Schaltung das über den Eingang **X** erhaltene Wort der Reihe nach mit allen im ROM gespeicherten Worten. In Abbildung 12 sind Teile der Realisierung der Schaltung abgebildet:

- Das Register **Ein_Wort** speichert das Eingabe-Wort.
 - Das ROM hat eine Speicherkapazität von 1024 kByte (1 Byte $\hat{=}$ 8 Bit).
 - Zwei Zähler, die über das **reset** Signal auf 0 zurückgesetzt werden können. Beide Zähler können über ein **enable** Signal aktiviert werden, d.h., sie zählen nur dann weiter, wenn das **enable** Signal auf 1 gesetzt ist. Der Zähler **Adr_Z** hat zusätzlich einen Ausgang **überlauf** der genau dann auf 1 gesetzt ist, wenn der Zähler von seinem Maximalwert auf 0 überläuft.
 - Der Komparator vergleicht zwei Worte und gibt genau dann eine 1 aus, wenn beide Worte übereinstimmen. Es kann angenommen werden, dass keine zeitliche Verzögerung vom Anlegen der ROM-Adresse bis zur Ausgabe des Komparators vorliegt.
 - Der Controller steuert den Ablauf der Zählung und setzt die notwendigen internen Steuersignale.
- (a) Vervollständigen Sie das Blockschaltbild aus Abbildung 12, indem Sie Ein- und Ausgänge der Komponenten und der externen Ein- und Ausgaben miteinander verbinden. Das **clk** Signal ist bereits angeschlossen und braucht daher nicht weiter berücksichtigt zu werden.
- (b) Tragen Sie in den grau hinterlegten Kreisen in Abbildung 12 die minimal benötigten Bitbreiten der internen Busse ein.
- (c) Abbildung 14 zeigt einen Moore-Automaten zur Steuerung des Controllers. Ergänzen Sie die Belegung der Ausgabesignale und die Bedingungen für Zustandsübergänge.
- (d) Ergänzen Sie entsprechend des in Aufgabenteil (c) erzeugten Automaten die in Abbildung 16 dargestellte Mikroprogrammsteuerung. Eintragungen müssen nur in den grau hinterlegten Flächen vorgenommen werden.

NAME:

Matrikelnummer:

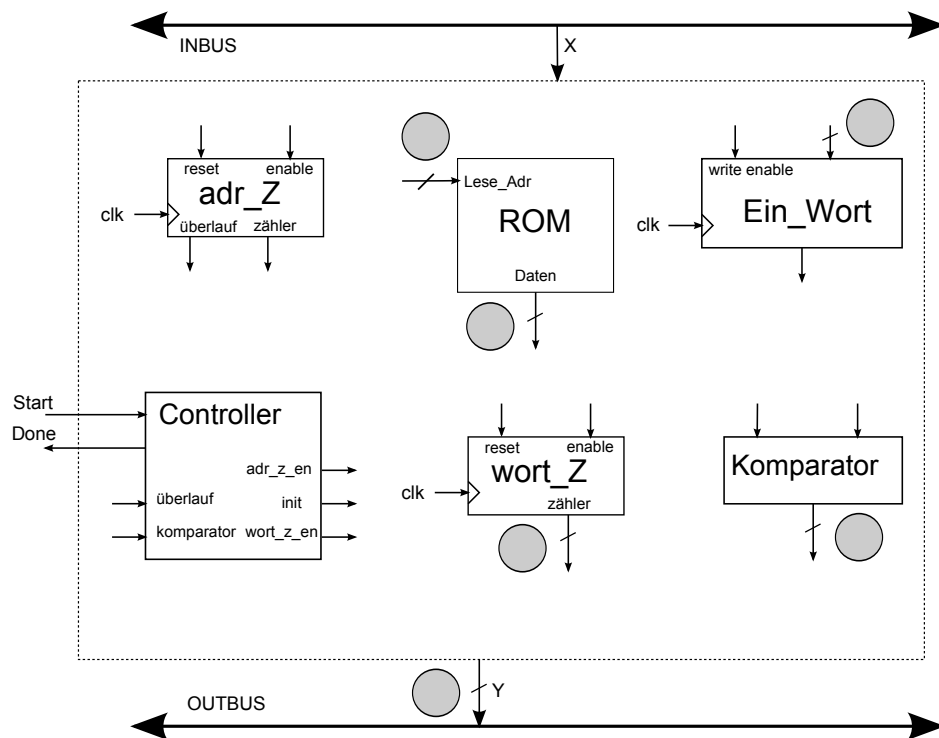


Abbildung 12: Blockschaltbild zu (a)

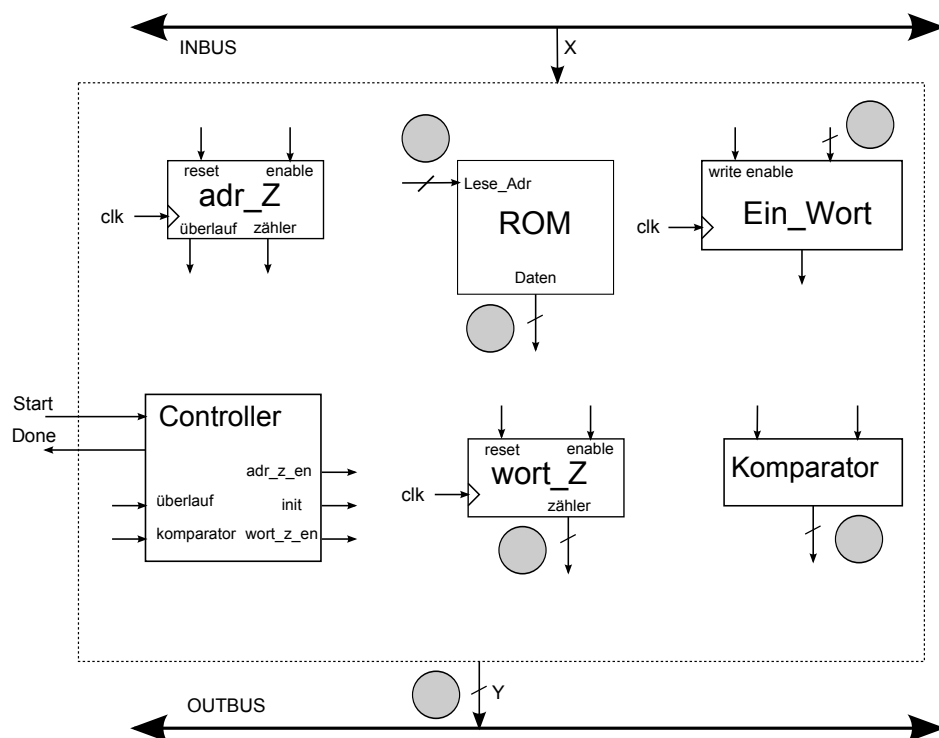


Abbildung 13: Blockschaltbild zu (a) (Ersatzdiagramm)

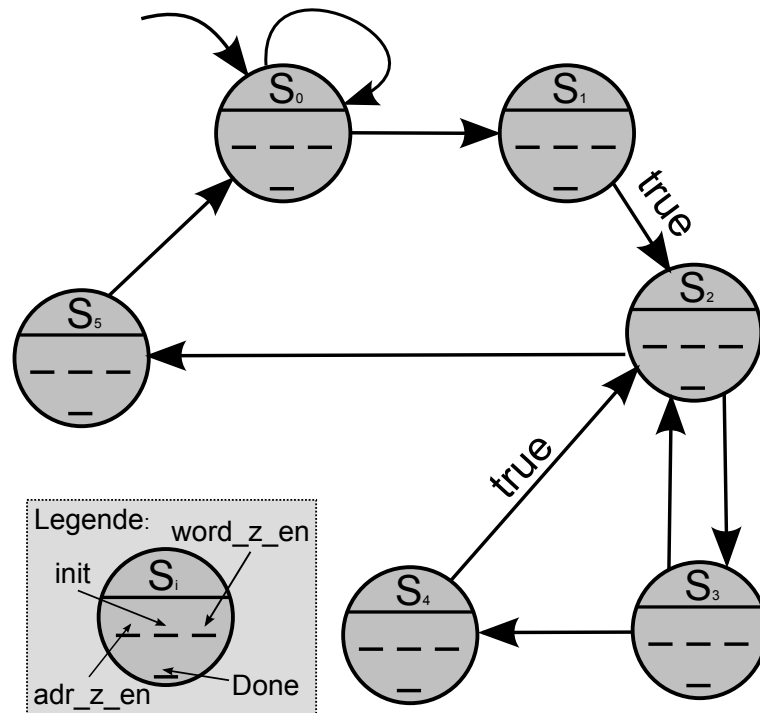


Abbildung 14: Moore-Automat zu Aufgabenteil (c)

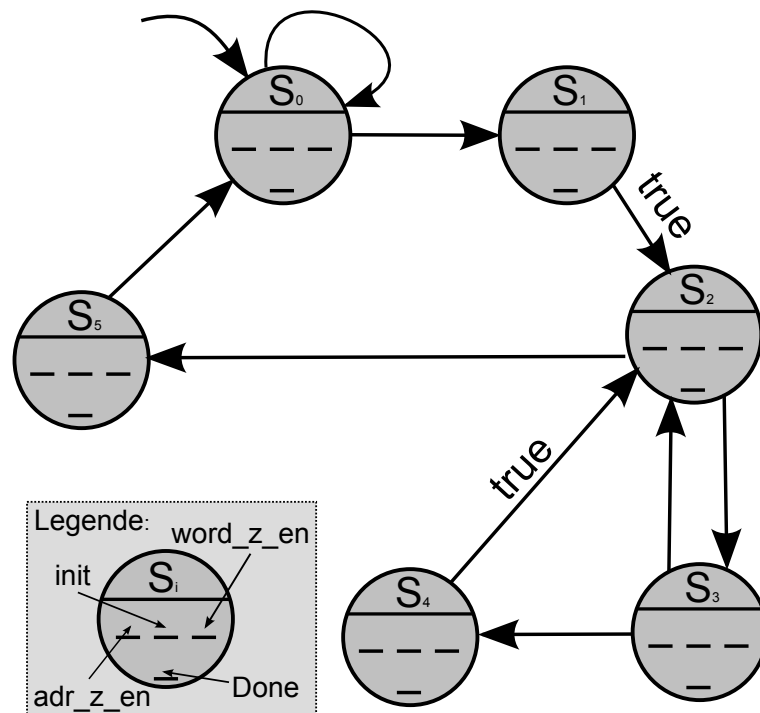


Abbildung 15: Moore-Automat zu Aufgabenteil (c) (Ersatzdiagramm)

NAME:

Matrikelnummer:

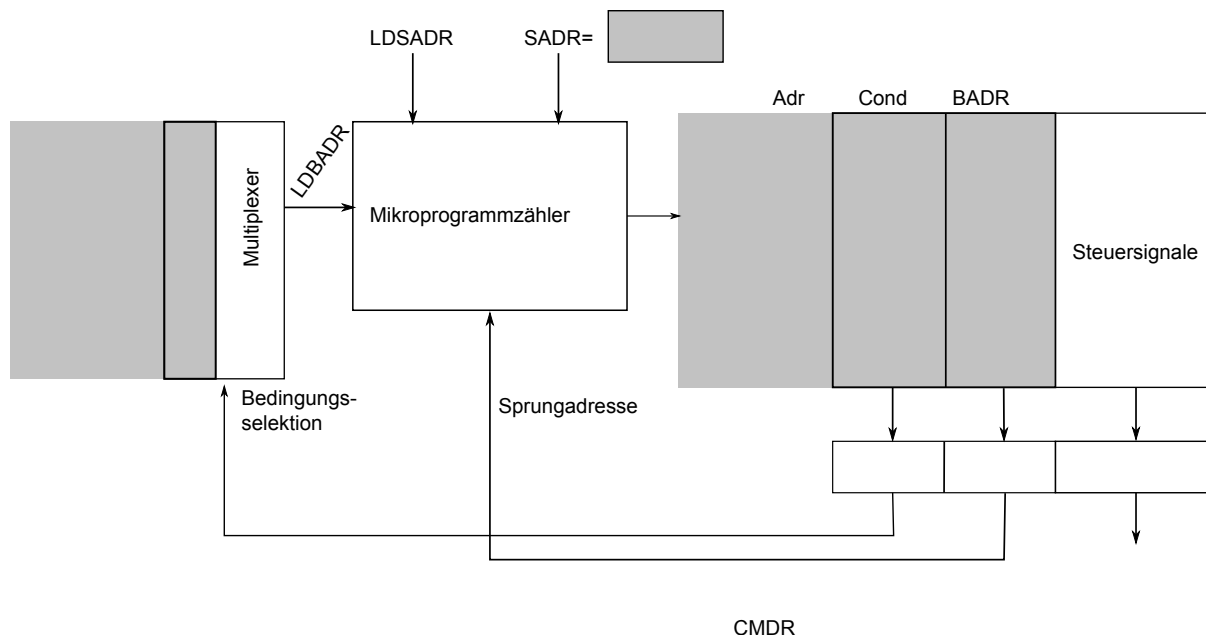


Abbildung 16: Mikroprogrammsteuerung zu Aufgabenteil (d)

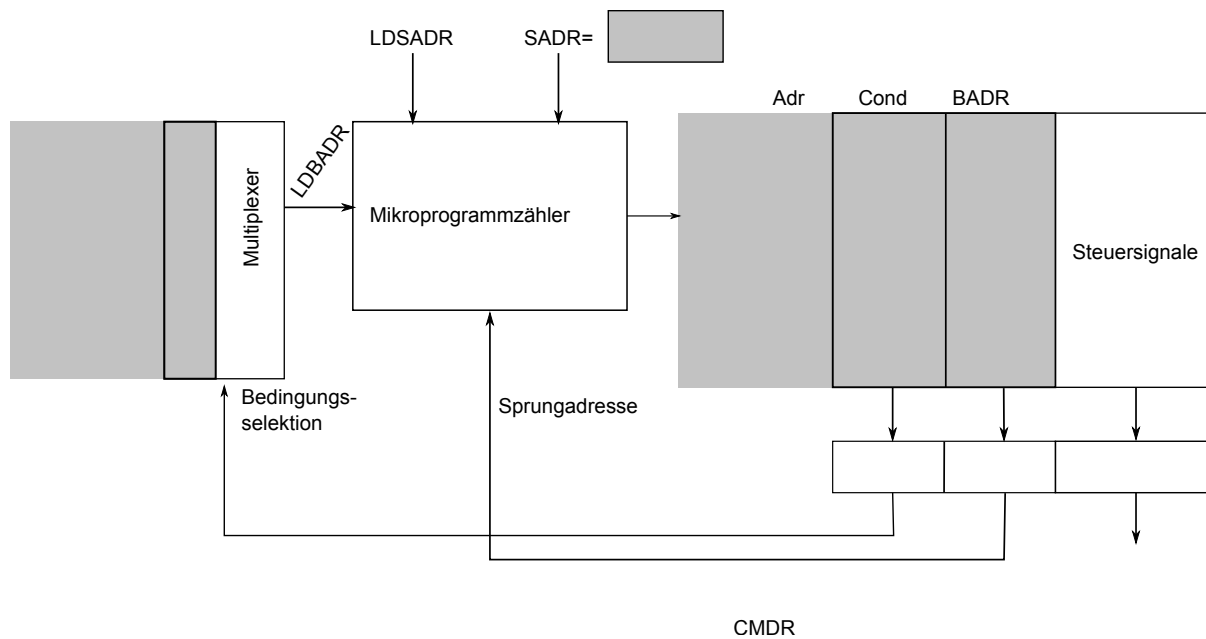


Abbildung 17: Mikroprogrammsteuerung zu Aufgabenteil (d) (Ersatzdiagramm)

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!

NAME:

Matrikelnummer:

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!

Konzeptpapier: Falls der Platz unter den einzelnen Aufgaben nicht ausreicht, können Sie diese Seiten für Zwischenrechnungen nutzen. Bitte Lösung und Lösungsweg eindeutig mit der Aufgabennummer markieren!