

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

**Klausur**  
**Digitaltechnik /**  
**Grundlagen Technische Informatik**

5. August 2011

Punkteverteilung								
Aufgabe	1	2	3	4	5	6	7	$\Sigma$
maximale Punkte	13	15	15	15	15	12	5	90
erreichte Punkte								

<b>Note:</b>	
--------------	--

Aufkleber

Name:	
Matrikelnummer:	
Studienrichtung:	

**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!

**Aufgabe 1:** (Zahldarstellung)

13 Punkte

- a) Gegeben sei die Zahl 1111 in 4-bit Zweierkomplementdarstellung. Welchen Wert hat die Zahl? Kreuzen Sie das richtige Ergebnis in folgender Tabelle an. (1 Punkt)

- ☐ 15  
☐ -1  
☐ -7  
☐ Keine Lösung ist richtig.

- b) Wandeln Sie die in hexadeximaler Schreibweise angegebenen Zahlen  $A$  und  $B$  in das duale Zahlensystem um.

(1 Punkt)

 $A = a5_{16} \leftrightarrow$  \_\_\_\_\_ $B = 1b_{16} \leftrightarrow$  \_\_\_\_\_

- c) Gegeben seien die 4-bit Zweierkomplementzahlen  $x=1111$  und  $y=0001$ . Die Addition liefert  $x+y=10000$ . Wie ist dieses Ergebnis zu interpretieren? (1 Punkt)

- ☐  $x+y$  hat den Wert 16.  
☐  $x+y$  hat den Wert 0.  
☐ Ungültiges Ergebnis wegen Overflow.  
☐ Keine Lösung ist richtig.

- d) Auch Brüche können als Zweierkomplementzahlen dargestellt werden. Bei einem Zweierkomplementbruch  $a_0, a_{-1}, a_{-2}, a_{-3}, \dots$  gibt  $a_0$  das Vorzeichen an, und die Komplementzahl wird durch bitweises Invertieren und Addition einer 1 am Bit mit der niedrigsten Wertigkeit berechnet. Welchen Wert hat der Zweierkomplementbruch  $1, 111$ ? (2 Punkte)

- ☐ 1,875  
☐  $-\frac{1}{7}$   
☐  $-\frac{1}{8}$   
☐ Keine Antwort ist richtig.

- e) Gegeben sei wieder der Zweierkomplementbruch 1,111. Wie sieht der Zweierkomplementbruch aus, wenn statt der 4-Bit Darstellung eine 8-Bit Darstellung verwendet werden soll? (1 Punkt)

- ☐ 1,1111111
- ☐ 1,1110000
- ☐ 1,0001111
- ☐ Keine Antwort ist richtig.

- f) Bei Umrechnung des Dezimalbruchs 0,3 in einen Binärbruch ergibt sich folgende Darstellung: (2 Punkte)

- ☐ 0,0100110011001...
- ☐ 0,11
- ☐ 0,110011001100...
- ☐ Keine Antwort ist richtig.

- g) Gegeben sei die Dezimalzahl 2,25. Wie sieht die Zahldarstellung im IEEE Gleitkommaformat aus? (2 Punkte)

Vorzeichen (1 bit):

---

Mantisse (23 bit):

---

Exponent (8 bit, Exzessdarstellung mit Verschiebungskonstante 127):

---

- h) Gegeben sei der Hamming-Code aus der Vorlesung. Die Codeworte haben die Form  $(x_1, x_2, x_3, x_4, p_1, p_2, p_3)$ , und für die Prüfbits gelte  $p_1 = x_1 \oplus x_2 \oplus x_3$ ,  $p_2 = x_1 \oplus x_2 \oplus x_4$ ,  $p_3 = x_1 \oplus x_3 \oplus x_4$ . Zu welchem Codewort wird die Information  $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$  ergänzt? (1 Punkt)

- ☐ (1, 0, 1, 1, 1, 1, 1)
- ☐ (1, 0, 1, 1, 0, 1, 1)
- ☐ (1, 0, 1, 1, 0, 1, 0)
- ☐ Keine Antwort ist richtig.

- i) Gegeben sei wieder der Hamming-Code aus der Vorlesung. Bei einer Datenübertragung erhält der Empfänger das Wort  $(1, 0, 0, 1, 1, 0, 1)$ . Kreuzen Sie alle richtigen Aussagen an: (2 Punkte)

- ☐ Bei einem Fehler im Prüfbit ist keine Korrektur möglich.
- ☐ Das Syndrom ist  $(0, 1, 1)$ .
- ☐ Das richtige Codewort ist  $(1, 0, 0, 1, 1, 0, 0)$ .
- ☐ Das Wort  $(1, 0, 0, 1, 1, 0, 1)$  ist ein Codewort, die Übertragung war fehlerfrei.
- ☐ Keine Antwort ist richtig.

### Aufgabe 2: (Automat)

15 Punkte

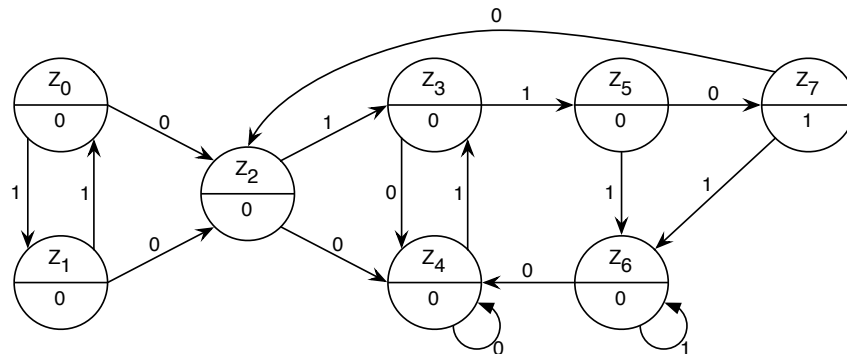


Abbildung 1: Moore-Automat

Gegeben sei der in Abbildung 1 dargestellte Moore-Automat. Im Folgenden soll dieser Moore-Automat vereinfacht werden.

- a) Wandeln Sie den in Abbildung 1 gegebenen Moore-Automat in einen Mealy-Automat  $A = (I, O, S, \delta, \lambda)$  um und füllen Sie die unten angegebene Automatentafel aus. (2 Punkte)

$$I = \{ \text{_____} \}, O = \{ \text{_____} \}$$
$$S = \{ \rule{10cm}{0.4pt} \}$$
[illegible]

- (8 Punkte)

[illegible]





$$S = \{ \underline{\hspace{10cm}} \}$$
[illegible]

[illegible][illegible]

$$S = \{ \underline{\hspace{10cm}} \}$$

[illegible]

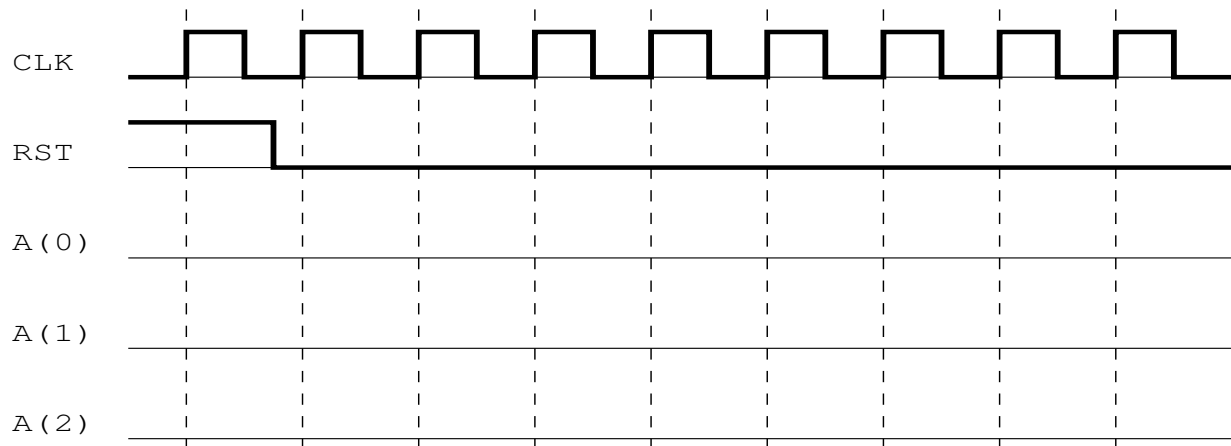
**Aufgabe 3: (VHDL)**

15 Punkte

- a) Im Folgenden finden Sie die VHDL-Beschreibung der Entity `Toplevel_A`. Diese generiert aus dem Taktsignal `CLK` das Ausgangssignal `A`. Vervollständigen Sie die Signalverläufe in Abbildung 2 (die Signale `CLK` und `RST` sind vorgegeben).

(Ersatzdiagramm auf Seite 13)

(8 Punkte)

Abbildung 2: Signalverläufe der Entity `Toplevel_A`

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Toplevel_A is
    port( CLK, RST : in  std_logic;
          A       : out std_logic_vector(2 downto 0));
end entity;

architecture Behavioral of Toplevel_A is
    signal sr : std_logic_vector(2 downto 0);
    signal invert : std_logic;
begin
    A <= sr;
    invert <= not sr(0);

    process(CLK,RST) begin
        if RST = '1' then
            sr <= "000";
        elsif CLK'event and CLK = '1' then
            sr <= invert & sr(2) & sr(1);
        end if;
    end process;
end architecture;

```

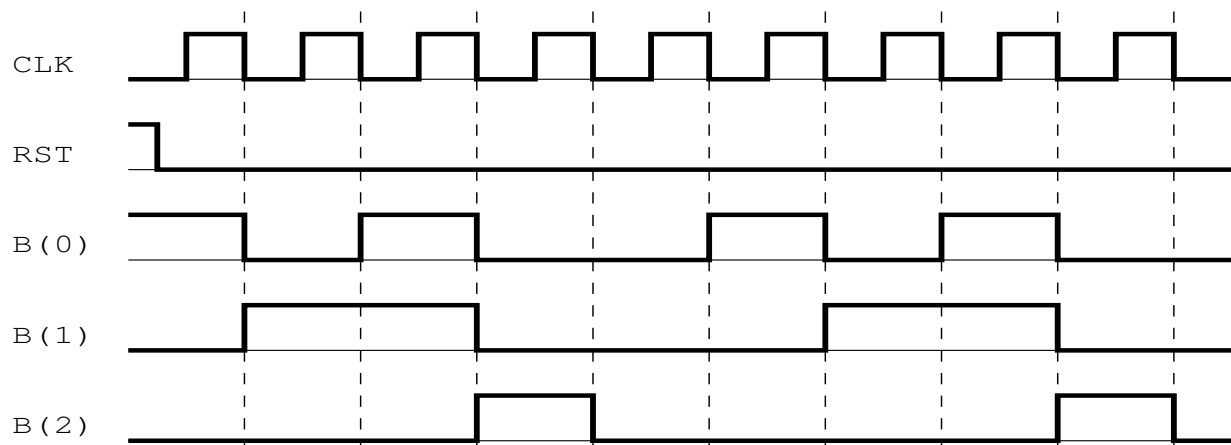


Abbildung 3: Signalverläufe der Entity Toplevel\_B

- b) Vervollständigen Sie das folgende Code-Fragment der Entity Toplevel\_B so, dass sich die in Abbildung 3 vorgegebenen Verläufe des Signals B einstellen. (7 Punkte)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Toplevel is
    port( CLK, RST : in  std_logic;
          B          : out std_logic_vector(2 downto 0));
end entity;

architecture Behavioral of Toplevel_B is
    signal ctr : _____;
begin
    B <= ctr;

    process(CLK,RST) begin
        if _____ then
            ctr <= _____;
        elsif CLK'event and _____ then
            _____;
            if _____ then
                ctr <= ctr + 1;
            end if;
        end if;
    end process;
end architecture;

```

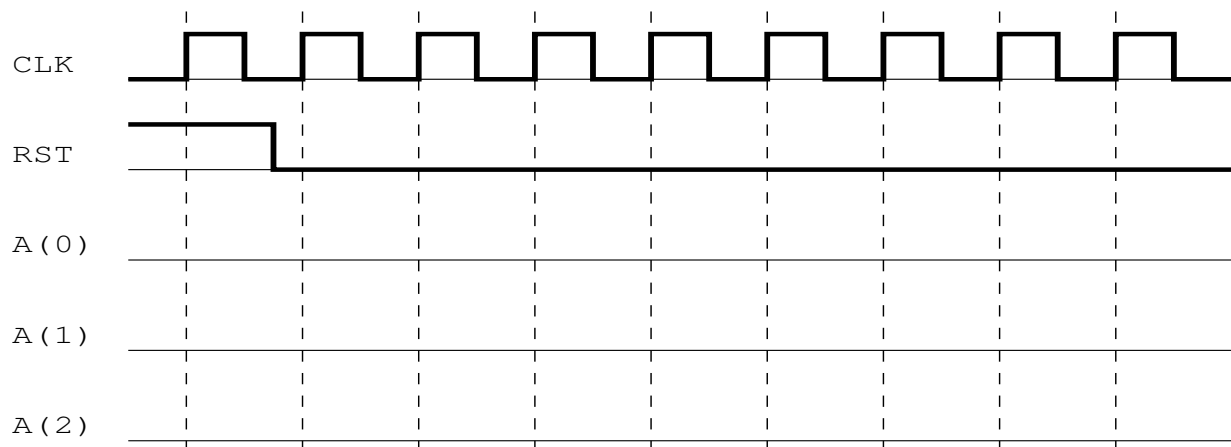


Abbildung 4: Ersatz Signalverläufe der Entity Toplevel\_A (**ungültige Lösung streichen!**)

Ersatz Code-Fragment der Entity Toplevel\_B (**ungültige Lösung streichen!**)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Toplevel is
    port( CLK, RST : in  std_logic;
          B          : out std_logic_vector(2 downto 0));
end entity;

architecture Behavioral of Toplevel_B is
    signal ctr : _____;
begin
    B <= ctr;

    process(CLK,RST) begin
        if _____ then
            ctr <= _____;
        elsif CLK'event and _____ then
            _____;
            if _____ then
                ctr <= ctr + 1;
            end if;
        end if;
    end process;
end architecture;
```

### Aufgabe 4: (Quine-McCluskey)

15 Punkte

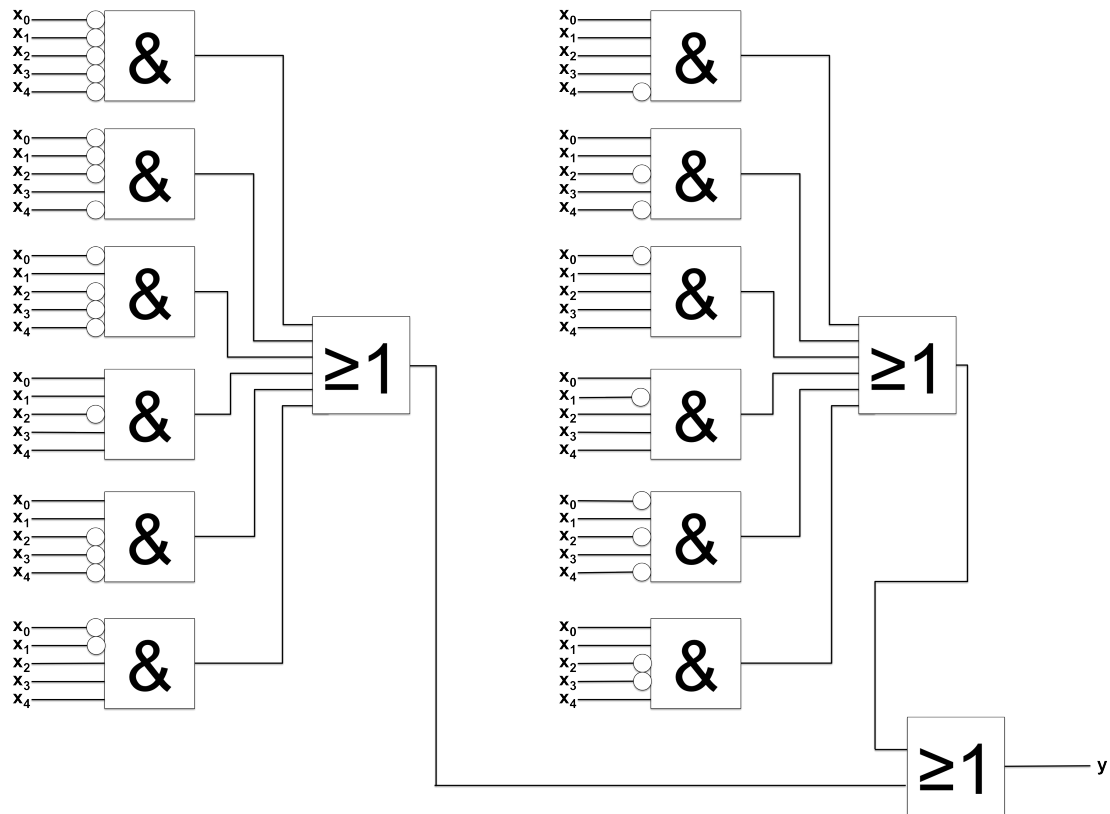


Abbildung 5: Logische Schaltung

Gegeben sei die Schaltung aus Abbildung 5 mit der Schaltungsfunktion  $y(x_0, x_1, x_2, x_3, x_4)$ . Ihr Ziel ist es, diese Schaltung mit Hilfe des Quine-McCluskey Verfahrens zu optimieren.

- a) Bestimmen Sie die Einstellenmenge  $\mathcal{E}$  der Schaltungsfunktion  $y$ .

(2 Punkte)

$$\mathcal{E} = \{ \rule{10cm}{0.4pt} \rule{10cm}{0.4pt} \rule{10cm}{0.4pt} \rule{10cm}{0.4pt} \}$$

- b) Ermitteln Sie mit Hilfe des Quine-McCluskey Verfahrens die Menge  $\mathcal{P}$  der Primimplikanten.

(6 Punkte)

$\mathcal{P} = \{$  \_\_\_\_\_  
\_\_\_\_\_  $\}$

- c) Gegeben sei jetzt eine andere Schaltungsfunktion  $y(x_0, x_1, x_2, x_3)$  mit den Primimplikanten:

$$\mathcal{P} = \{-000, 0-00, 01-0, -110, 10-0\}$$

Füllen Sie die Überdeckungstabelle aus und bestimmen Sie die minimale SOP-Form (**S**um **O**f **P**roducts).

(5 Punkte)

$P_0$								
$P_1$								
$P_2$								
$P_3$								
$P_4$								

Tabelle 1: Überdeckungstabelle

$P_0$								
$P_1$								
$P_2$								
$P_3$								
$P_4$								

Tabelle 2: Ersatz Überdeckungstabelle (**ungültige Lösung streichen!**)



Die minimale SOP-Form lautet:

$$y(x_0, x_1, x_2, x_3) = \underline{\hspace{15cm}}$$
  
$$\underline{\hspace{15cm}}$$

- d) Zeichnen Sie die minimierte Schaltung aus Aufgabenteil c). Benutzen Sie dazu sowohl UND- als auch ODER-Gatter, wobei die Eingänge der Gatter direkt mit den Schaltungseingängen beschriftet werden dürfen (ähnlich dem Schaltungsaufbau in Abbildung 5).

(2 Punkte)

**Aufgabe 5:** (Fehlerkorrigierende Codes)

15 Punkte

Gegeben sei der Code aus Tabelle 3 bestehend aus 4 Codeworten der Form  $(d_1, d_2, p_1, p_2, p_3)$ :

$d_1$	$d_2$	$p_1$	$p_2$	$p_3$
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

Tabelle 3: Code

- a) Wie viele gleichzeitige Bit-Fehler lassen sich mit diesem Code maximal *erkennen*?

(1 Punkt)

---

- b) Wie viele gleichzeitige Bit-Fehler lassen sich mit diesem Code maximal *korrigieren*?

(1 Punkt)

---

Der Code soll im Folgenden als fehlerkorrigierender Code eingesetzt werden. Die Datenbits  $d_1$  und  $d_2$  werden durch die 3 Paritätsbits  $p_1$ ,  $p_2$  und  $p_3$  abgesichert.

- c) Stellen Sie die Gleichungen für die Berechnung von  $p_1$ ,  $p_2$  und  $p_3$  in Abhängigkeit von  $d_1$  und  $d_2$  auf.

(1,5 Punkte)

---

---

---

- d) Das Syndrom  $(e_1, e_2, e_3)$  vergleicht die übertragenen mit den erwarteten Prüfbits. Stellen Sie nun die Gleichungen für die Berechnung der „*Error Flags*“  $e_1$ ,  $e_2$  und  $e_3$  auf.

(1,5 Punkte)

---



---



---

- e) Füllen Sie Tabelle 4 zur Bestimmung des fehlerhaften Bits anhand der soeben gewonnenen Gleichungen aus. Gehen Sie davon aus, dass maximal ein Bit gekippt ist. Sollte ein Fehler nur durch einen Mehrfachfehler erklärbar sein, markieren Sie die Zeile mit X.

(4 Punkte)

$e_1$	$e_2$	$e_3$	Fehlerquelle
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Tabelle 4: Fehlerquellen

$e_1$	$e_2$	$e_3$	Fehlerquelle
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Tabelle 5: Ersatz Fehlerquellen  
(ungültige Lösung streichen!)

- f) Implementieren Sie die Hardware für die Korrekturschaltung. Die Schaltung erhält als Eingabe die empfangenen Bits  $c_1$  und  $c_2$  sowie die berechneten Error-Flags  $e_1, e_2$  und  $e_3$  und gibt die korrigierten Daten  $d_1$  und  $d_2$  aus. Das Signal *fail* wird gesetzt, wenn mehr als ein Fehler aufgetreten sein muss.

(6 Punkte)

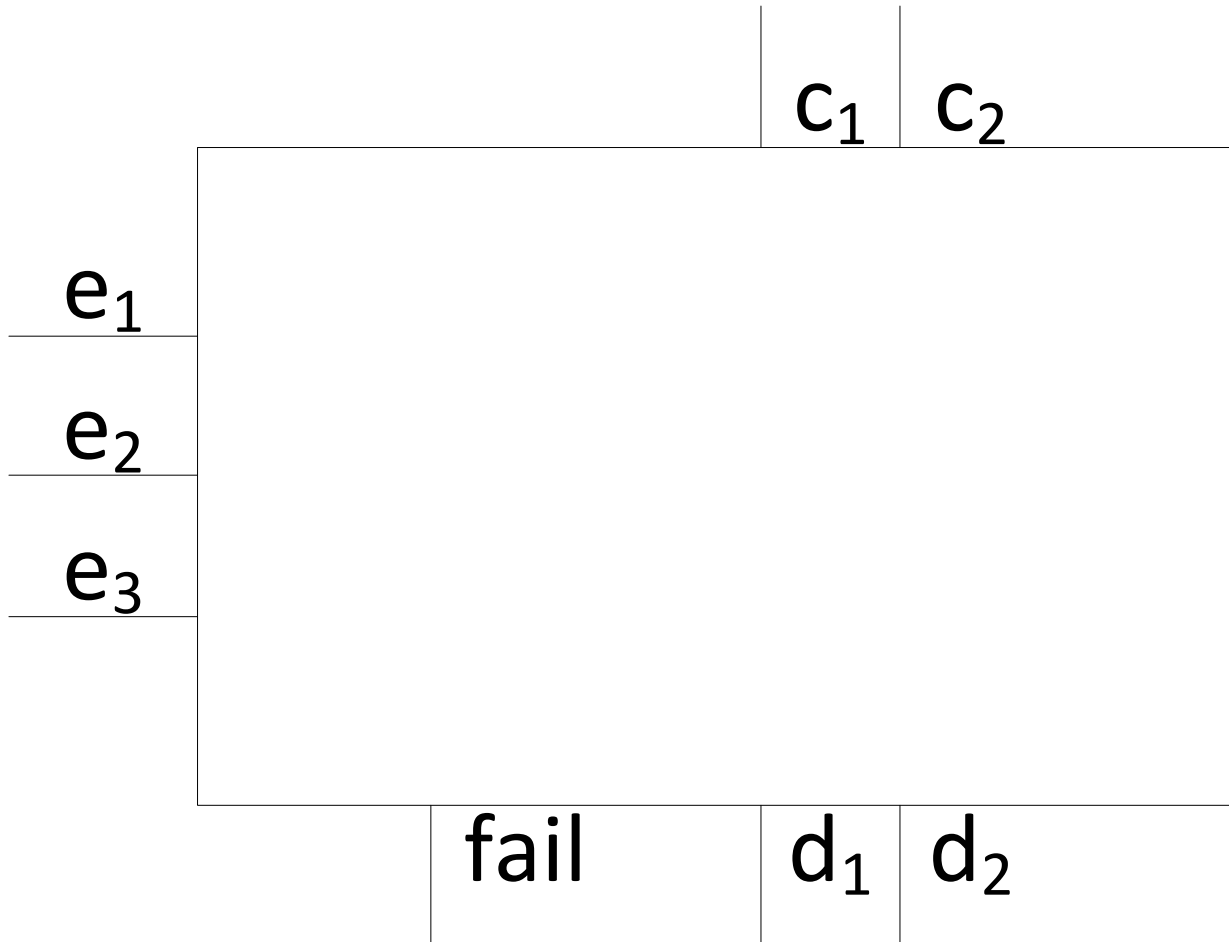


Abbildung 6: Fehlerkorrekturschaltung

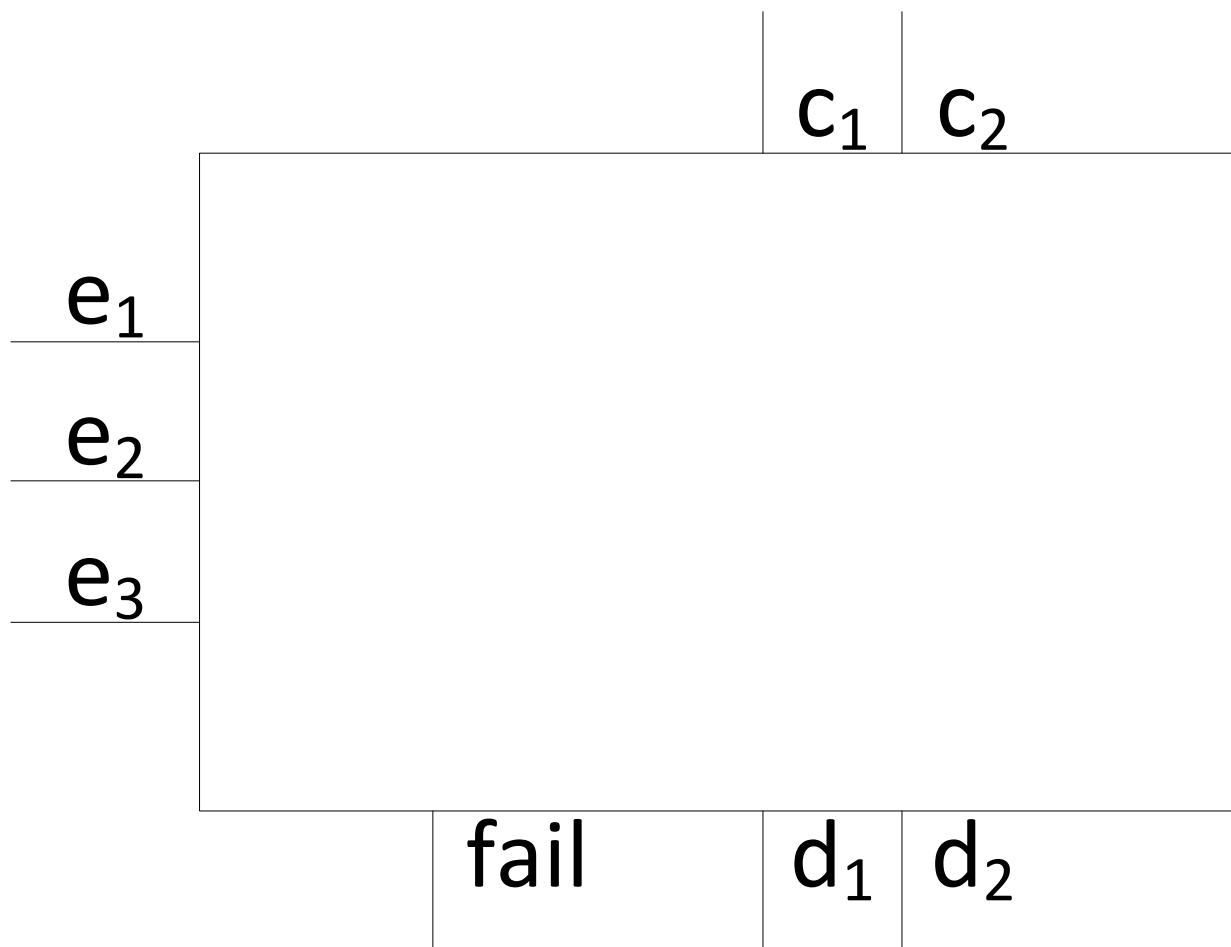


Abbildung 7: Ersatz für Fehlerkorrekturschaltung (**ungültige Lösung streichen!**)

**Aufgabe 6:** (RTL Design)

12 Punkte

Gegeben sind die zwei 8-bit Zweierkomplementzahlen:

$$A = 00111000_2, B = 10011000_2$$

Das Produkt  $P = A \cdot B$  soll mit Booth'scher Multiplikation berechnet werden. Die Register A, B und P seien dazu wie in Abbildung 8 initialisiert.

- a) Ergänzen Sie die folgende Tabelle und tragen Sie für jede Kombination von  $a_0, a_{-1}$  die jeweils auszuführende Operation (add, sub, - ) in die Spalte OP ein.

(1 Punkt)

$a_0$	$a_{-1}$	OP
0	0	
0	1	
1	0	
1	1	

- b) Berechnen Sie  $A \cdot B$ . Benutzen Sie das Schema in Abbildung 8 (Seite 23) . Geben Sie dabei für jeden Schritt die Zwischenwerte und die Operation OP an. Markieren Sie außerdem das Ergebnis.

(6 Punkte)

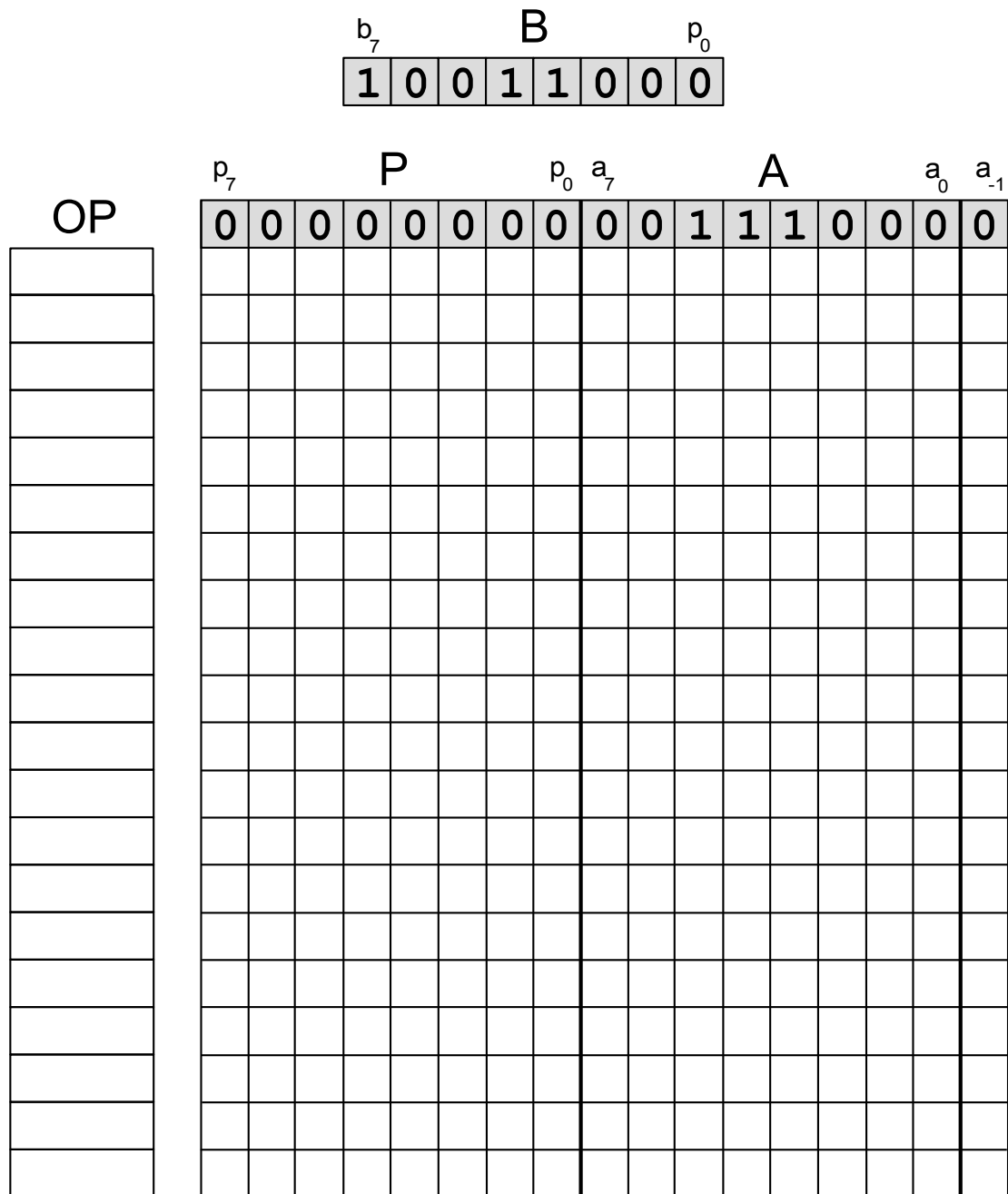


Abbildung 8: Booth Schema (**Ersatztablelle auf Seite 26**)





- d) Tragen Sie in Tabelle 6 zu den Steuersignalen  $c_1..c_7$  des Booth Multiplizierers (Abbildung 9) jeweils eine kurze Funktionsbeschreibung ein.

(4 Punkte)

Kontrollpunkt	Funktionsbeschreibung
$c_1$	
$c_2$	
$c_3$	
$c_4$	
$c_5$	
$c_6$	
$c_7$	

Tabelle 6: Kontrollpunktbeschreibung

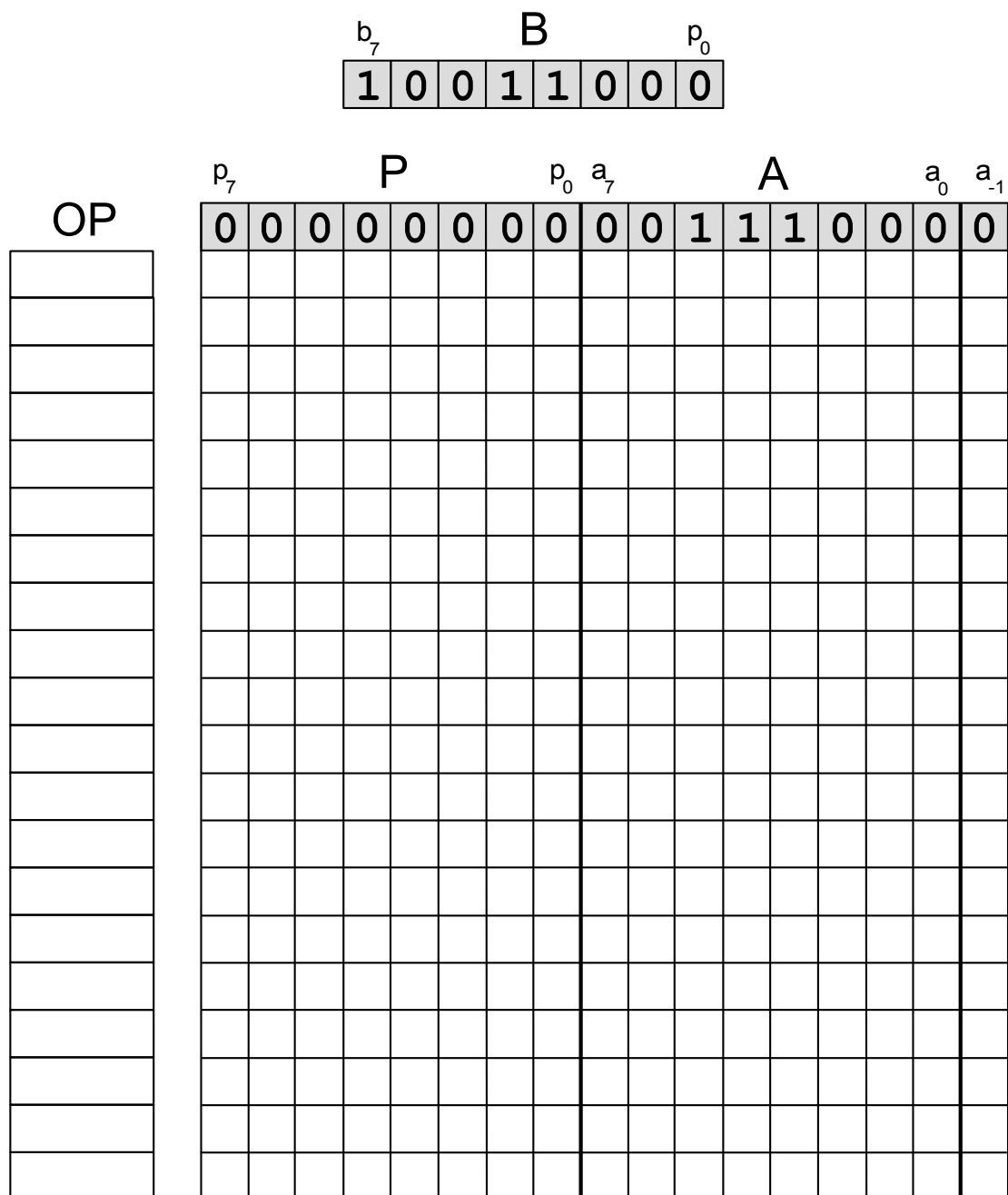


Abbildung 10: Ersatz Booth Schema (**ungültige Lösung streichen!**)

**Aufgabe 7:** (Mikroprogrammierbares Steuerwerk)

5 Punkte

Der Automat in Abbildung 11 soll als Mikroprogramm realisiert werden. Geben Sie die Speicherbelegung für dieses Programm in Abbildung 12 an.

**Hinweis:** Eine '1' am Eingang load lädt die jeweilige Sprungadresse in den 3-bit Zähler.

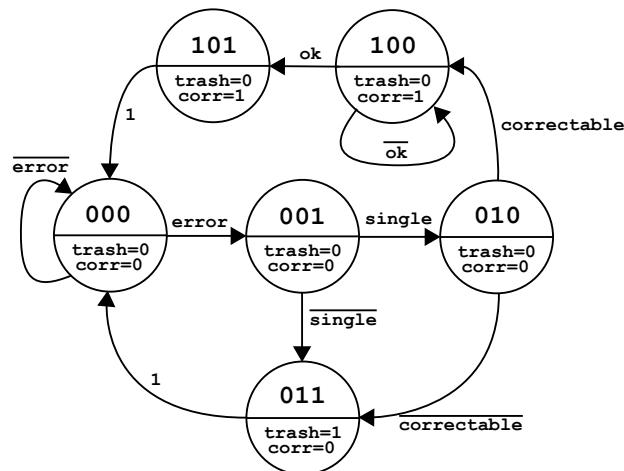


Abbildung 11: Automat des µProgramms

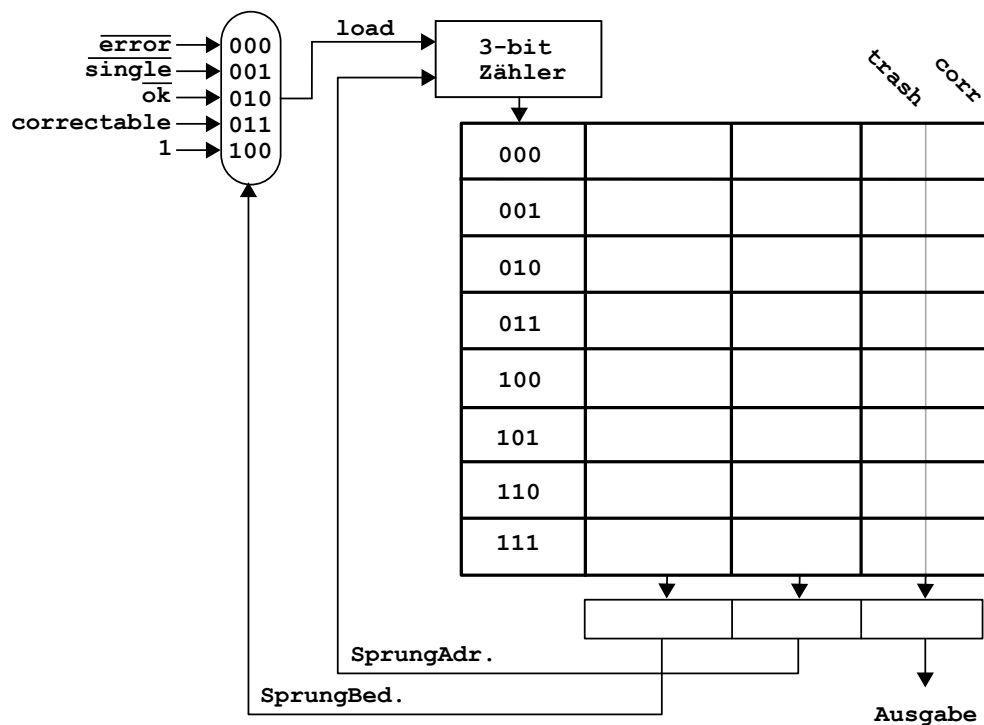
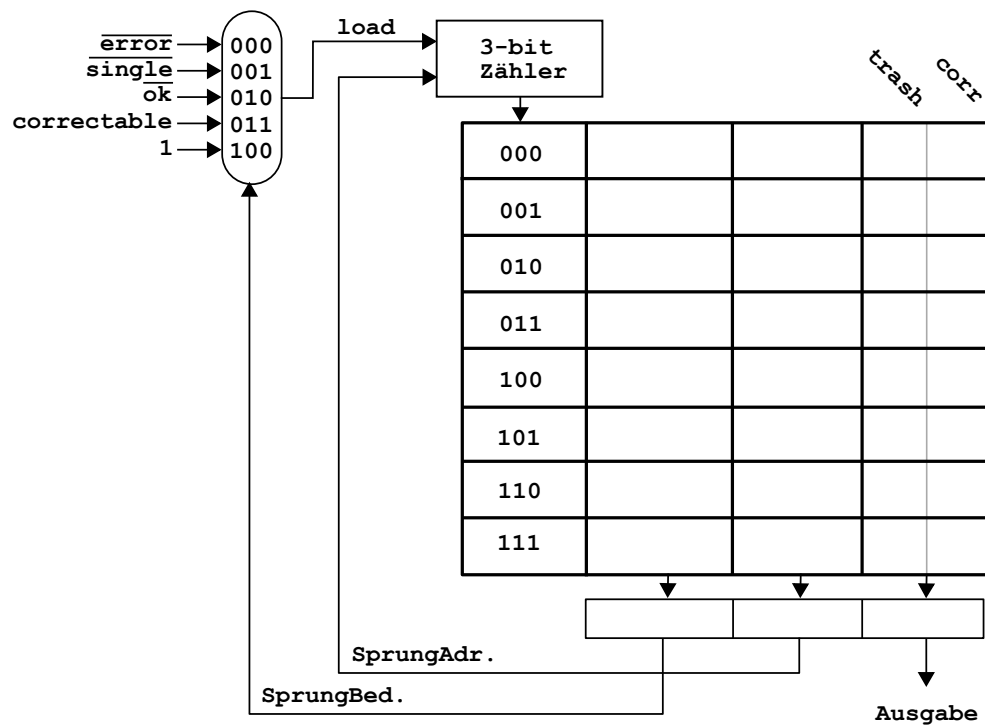


Abbildung 12: Schaltbild µProgramm

Abbildung 13: Ersatz Schaltbild  $\mu$ Programm (ungültige Lösung streichen!)





