

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

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

30. März 2012

Punkteverteilung							
Aufgabe	1	2	3	4	5	6	$\Sigma$
maximale Punkte	12	15	15	14	14	20	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 und Kodierung)

12 Punkte

- a) Gegeben sei die Zahl 1110 in 4-Bit Zweierkomplementdarstellung. Welchen Wert hat die Zahl? Kreuzen Sie das richtige Ergebnis in folgender Tabelle an. (1 Punkt)

- ☐ 14  
☐ -2  
☐ -6  
☐ Keine Lösung ist richtig.

- b) Gegeben sei wieder die Zahl 1110 in 4-Bit Zweierkomplementdarstellung. Die Zahldarstellung soll jetzt auf eine 8-Bit Zweierkomplementdarstellung erweitert werden. Wie sieht die korrekte Zahldarstellung aus? Kreuzen Sie das richtige Ergebnis in folgender Tabelle an. (1 Punkt)

- ☐ 00001110  
☐ 11111110  
☐ 10001110  
☐ Keine Lösung ist richtig.

- c) Gegeben sei jetzt die Zahl 1110 in 4-Bit Exzessdarstellung mit Verschiebungskonstante 8. Welchen Wert hat die Zahl? Kreuzen Sie das richtige Ergebnis in folgender Tabelle an. (1 Punkt)

- ☐ 6  
☐ -6  
☐ 14  
☐ Keine Lösung ist richtig.

- d) Bei Umrechnung des Dezimalbruchs 0,8 in einen Binärbruch ergibt sich folgende Darstellung: (2 Punkte)

- ☐ 0,01100110011....  
☐ 0,1000  
☐ 0,10011001100....  
☐ Keine Lösung ist richtig.

- e) Es werden nun die Zahlen  $x = 1001$  und  $y = 1011$  in 4-Bit Zweierkomplementdarstellung betrachtet. Bei der Addition der beiden Zahlen ergibt sich  $x + y = 10100$ . Kreuzen Sie bitte alle richtigen Aussagen in der Tabelle an. (1 Punkt)

- ☐ 10100 ist keine gültige 4-Bit Zweierkomplementzahl (wegen Überlauf).
- ☐ In 4-Bit Zweierkomplementdarstellung lautet das Ergebnis 0100.
- ☐ In 4-Bit Betrag/Vorzeichen Darstellung ist das Ergebnis 1100.
- ☐ Keine Antwort ist richtig.

- f) 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 Addieren einer 1 am Bit mit der niedrigsten Wertigkeit berechnet. Für den Bruch  $-3/4$  ergibt sich folgende 4-Bit Darstellung: (1 Punkt)

- ☐ 1,110
- ☐ 1,011
- ☐ 1,010
- ☐ Keine Antwort ist richtig.

- g) Gegeben sei der Zweierkomplementbruch 1,101. Wie sieht der Zweierkomplementbruch aus, wenn statt der 4-Bit Darstellung eine 8-Bit Darstellung verwendet werden soll? (1 Punkt)

- ☐ 1,1011111
- ☐ 1,1010000
- ☐ 1,1111101
- ☐ Keine Antwort ist richtig.

- h) Gegeben sei die Dezimalzahl -2,75. Wie sieht die Zahldarstellung im IEEE Gleitkommaformat aus? (2 Punkte)

Vorzeichen (1 Bit):

---

Mantisse (23 Bit):

---

Exponent (8 Bit, Exzessdarstellung mit Verschiebungskonstante 127):

---

- i) 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  $p_1 = x_1 \oplus x_2 \oplus x_3$ ,  $p_2 = x_1 \oplus x_2 \oplus x_4$ ,  $p_3 = x_1 \oplus x_3 \oplus x_4$ . Bei einer Datenübertragung erhält der Empfänger das Wort (1, 0, 1, 1, 1, 0, 1). Kreuzen Sie alle richtigen Aussagen an: (2 Punkte)

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

## Aufgabe 2: (Automat)

15 Punkte

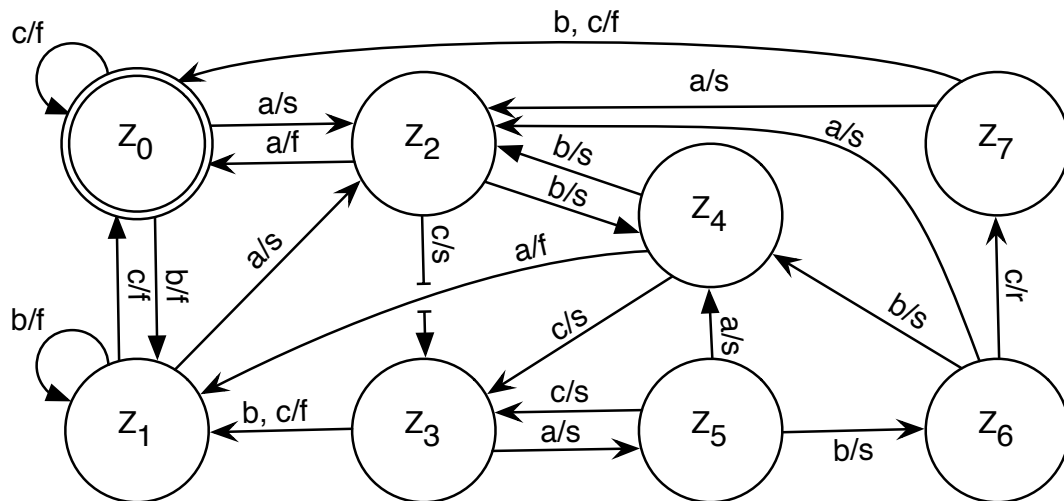


Abbildung 1: Mealy-Automat

Gegeben sei der in Abbildung 1 dargestellte Mealy-Automat mit dem Startzustand  $z_0$ . Im Folgenden soll dieser Mealy-Automat vereinfacht werden.

- a) Geben Sie den in Abbildung 1 gegebenen Mealy-Automat  $A = (X, Y, S, \delta, \lambda, s_1)$  an und füllen Sie die unten angegebene Automatentafel aus.

(3 Punkte)

$$X = \{ \text{_____} \}, Y = \{ \text{_____} \}$$

$$S = \{ \text{_____} \}, s_1 = \{ \text{_____} \}$$

[illegible]

[illegible]



c) Geben Sie den Automatengraph des von Ihnen minimierten Mealy-Automaten an.

(2 Punkte)



**Aufgabe 3:** (VHDL)

15 Punkte

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

(10 Punkte)

---

```
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 x,y,z : std_logic;
begin
    A <= x & y & z;

    process(CLK,RST) begin
        if RST = '1' then
            x <= '0';
            y <= '0';
            z <= '0';
        elsif CLK'event and CLK = '1' then
            x <= not x;
            y <= x;
            z <= (y and z) or (x and not z);
        end if;
    end process;
end architecture;
```

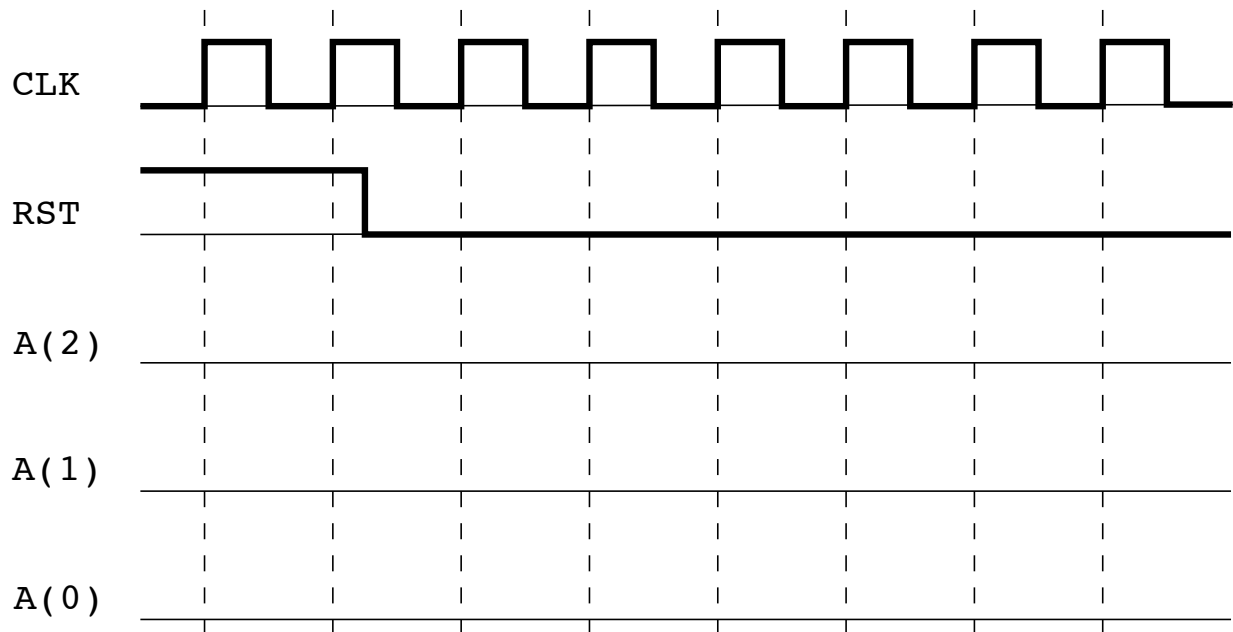
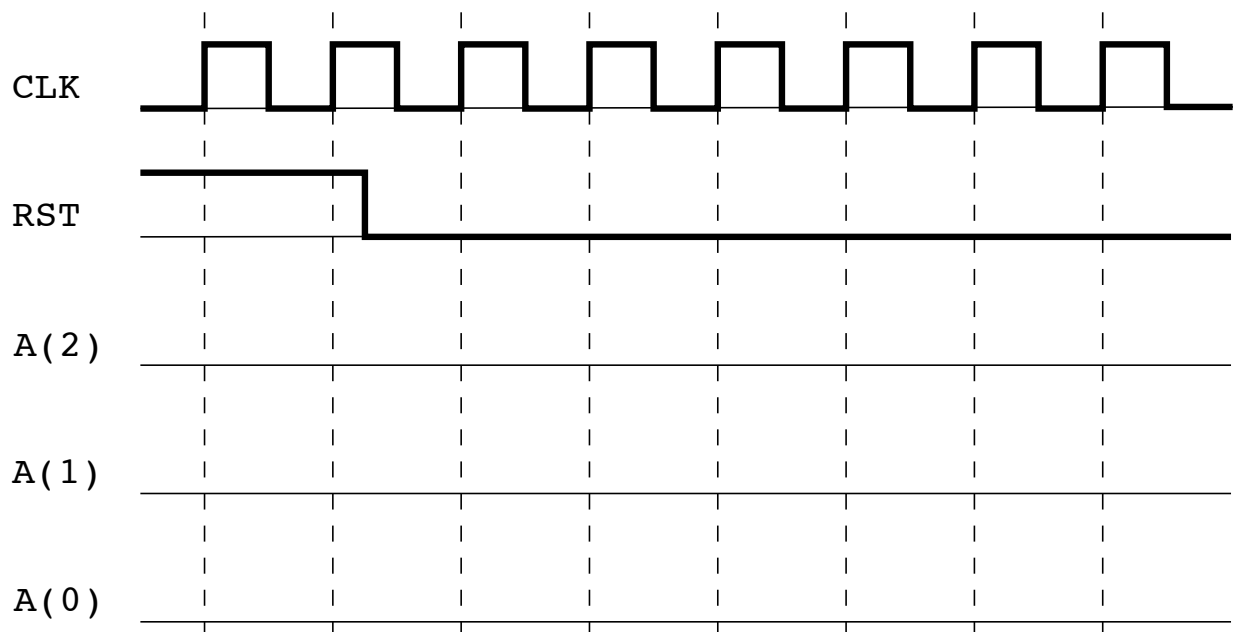


Abbildung 2: Signalverläufe der Entity Toplevel\_A

Abbildung 3: Ersatz Signalverläufe der Entity Toplevel\_A, **ungültige Lösung streichen!**

b) Vervollständigen Sie das folgende Code-Fragment der Entity `Toplevel_B` so, dass sich die in Abbildung 4 vorgegebenen Signalverläufe für P und B einstellen.

(5 Punkte)

---

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

entity Toplevel_B is
    port( CLK : in  std_logic;
          RST : in  std_logic;
          B   : out std_logic_vector(1 downto 0);
          P   : out std_logic);
end entity;

architecture Behavioral of Toplevel_B is
    signal c : std_logic_vector(1 downto 0);
begin
    B <= _____;
    P <= _____;

    process (CLK, RST) is
    begin
        if RST = '1' then
            c <= _____;
        elsif _____ then
            if c = "11" then
                c <= "00";
            else
                c <= c + 1;
            end if;
        end if;
    end process;
end Behavioral;
```

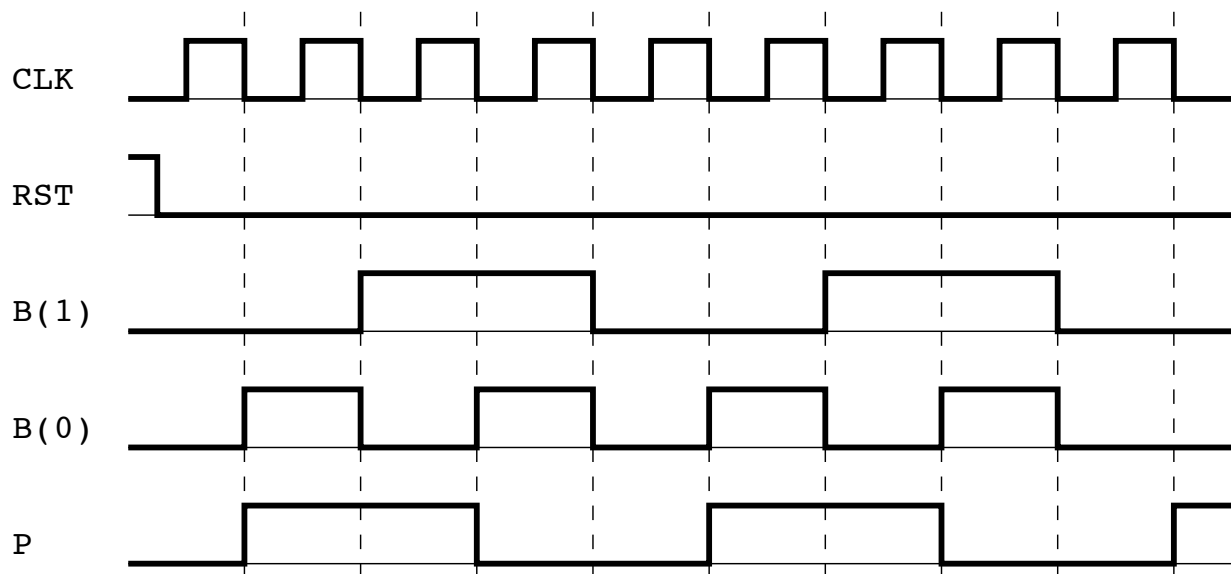


Abbildung 4: Signalverläufe der Entity Toplevel\_B

Ersatz (nur Architecture), **ungültige Lösung streichen!**

```

architecture Behavioral of Toplevel_B is
    signal c : std_logic_vector(1 downto 0);
begin
    B <= -----;
    P <= -----;

    process (CLK, RST) is
    begin
        if RST = '1' then
            c <= -----;
        elsif ----- then
            if c = "11" then
                c <= "00";
            else
                c <= c + 1;
            end if;
        end if;
    end process;

end Behavioral;

```

**Aufgabe 4:** (Logikminimierung)

14 Punkte

Gegeben sei die Schaltung in Abbildung 5.

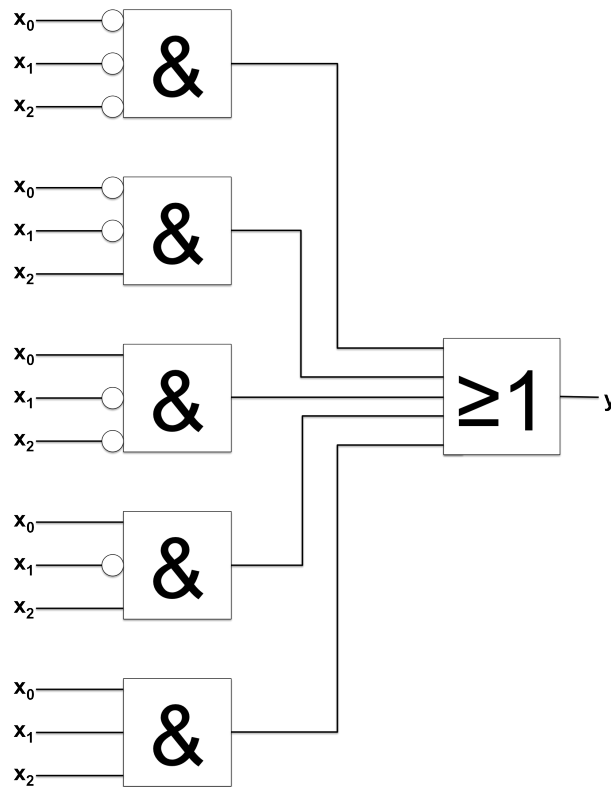


Abbildung 5: Logische Schaltung

a) Geben sie die disjunkte Normalform (*DNF*) der Schaltung an.

(2 Punkte)

$y(x_0, x_1, x_2) =$  \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

- b) Optimieren Sie die Schaltungsfunktion mit Hilfe Boolescher Umformungen und geben Sie die minimale Funktion an.

(3 Punkte)

$y(x_0, x_1, x_2) =$  \_\_\_\_\_

\_\_\_\_\_

- c) Zeichnen Sie die minimale Schaltungsfunktion. Ihnen stehen *UND*- und *ODER*-Gatter (mit jeweils 2 Eingängen), sowie *INVERTER* zur Verfügung.

(2 Punkte)

- d) Die Schaltung wird im Folgenden um verschiedene Funktionen erweitert. Gegeben sei die Menge der Primimplikanten der neuen Schaltungsfunktion durch:

$$\mathcal{P} = \{0 - -0, 0 - 0 -, 01 - -, 10 - 1, -001\}$$

Bestimmen Sie mit Hilfe nachfolgender Tabelle die minimale SOP-Form (**S**um **O**f **P**roducts).

(7 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:

$$y(x_0, x_1, x_2, x_3) = \underline{\hspace{10cm}}$$

**Aufgabe 5:** (Division)

14 Punkte

Abbildung 6 zeigt den Aufbau eines sequentiellen Dividierers für ganzzahlige positive Zahlen. Für die Division werden die Operandenregister **A** und **B** mit dem Dividenden und dem Divisor geladen. Nach Abarbeitung der Division steht der Quotient in **A** und der Rest der Division in **Y**.

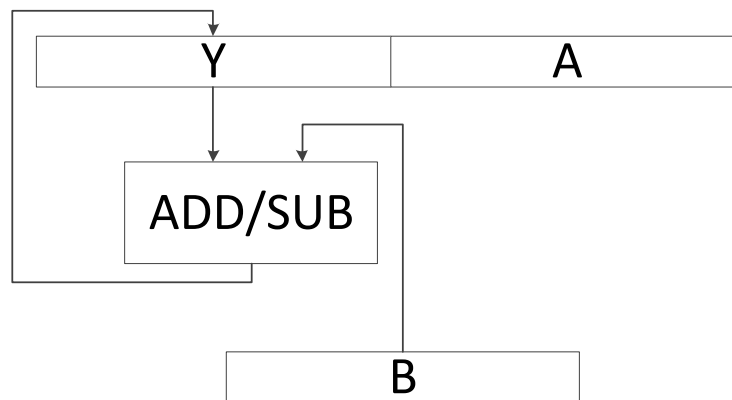


Abbildung 6: Schaltbild eines sequentiellen Dividierers

- a) Wie groß müssen die Register **A**, **B** und **Y** gewählt werden um Zahlen im Bereich von 0 bis 255 zu dividieren?

(1 Punkt)

A: \_\_\_\_\_ B: \_\_\_\_\_ Y: \_\_\_\_\_

- b) Wandeln Sie die Zahlen  $43_{10}$  und  $3_{10}$  in das duale Zahlensystem um.

(1 Punkt)

$43_{10} \leftrightarrow$  \_\_\_\_\_

$3_{10} \leftrightarrow$  \_\_\_\_\_

- c) Tragen Sie die fehlenden Aktionen in den unten gegebenen Algorithmus für die sequentielle Division **ohne** Rückspeichern ein:

(4 Punkte)

1. Operandenregister B laden

2.  $Y \leftarrow 0$ , A laden

3. führe n mal aus:

\* schiebe \_\_\_\_\_ um \_\_\_\_\_ nach \_\_\_\_\_

\* wenn  $Y$  \_\_\_\_\_ dann  $Y \leftarrow$  \_\_\_\_\_  
sonst  $Y \leftarrow$  \_\_\_\_\_

\* wenn  $Y$  \_\_\_\_\_ dann  $A(\text{_____}) \leftarrow$  \_\_\_\_\_  
sonst  $A(\text{_____}) \leftarrow$  \_\_\_\_\_

4. wenn  $Y$  \_\_\_\_\_ dann  $Y \leftarrow$  \_\_\_\_\_

5. Quotient steht in A, der Rest in Y

- d) Führen Sie nun die schriftliche Division ohne Rückspeichern mit den Zahlen  $43_{10}$  und  $3_{10}$  durch. Benutzen Sie dafür das Schema in Abbildung 7. Geben Sie in der ersten Spalte die Operation (shift, add, sub) an, die Sie benutzt haben um auf das jeweilige Zwischenergebnis zu kommen.

(8 Punkte)





**Aufgabe 6:** (Steuerung)

20 Punkte

Der in Abbildung 9 spezifizierte Automat soll durch eine speicherprogrammierte Steuerung realisiert werden. Der Zustand S0 bezeichnet den Startzustand. Die Ausgabe  $C_i$  bedeutet, dass die  $i$ -te Steuerleitung 1 ist, die restlichen 0. Ein \* bedeutet, dass alle Steuerleitungen auf 0 gesetzt werden.

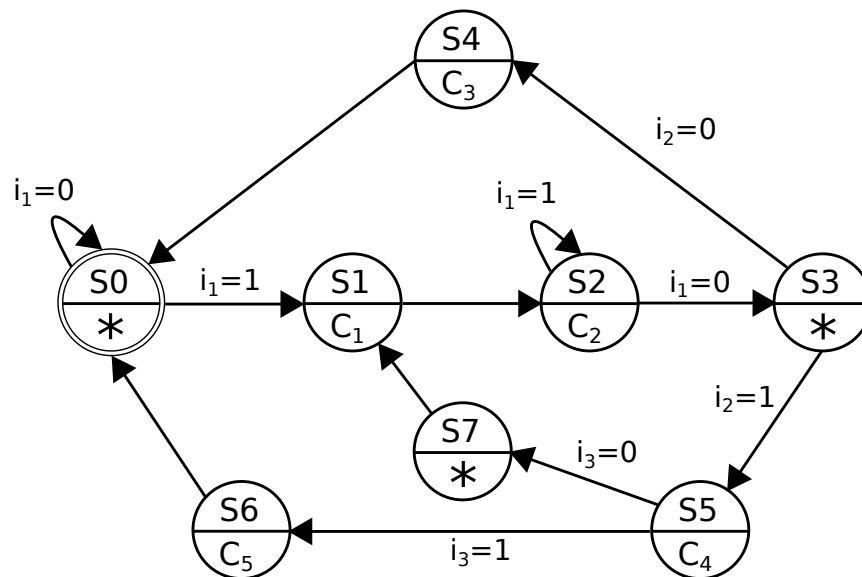


Abbildung 9: Automat

- a) Um welchen Automatentyp handelt es sich? Begründen Sie Ihre Antwort, ansonsten gibt es keinen Punkt. (1 Punkt)

---



---

- b) Geben Sie für den in Abbildung 9 dargestellten Automaten alle notwendigen Sprungbedingungen für eine speicherprogrammierte Steuerung an. (3 Punkte)

---



---



---



---



---



---

- c) Geben Sie die minimale Anzahl an Bits an, die man zur Kodierung der Sprungbedingungen benötigt. (1 Punkte)

- d) Der Automat in Abbildung 10 soll ebenfalls als speicherprogrammierte Steuerung realisiert werden. Der Zustand S0 bezeichnet den Startzustand. Die Ausgabe  $C_i$  bedeutet, dass die  $i$ -te Steuerleitung 1 ist, die restlichen 0. Ein \* bedeutet, dass alle Steuerleitungen auf 0 gesetzt werden. Geben Sie in Abbildung 11 die fehlenden minimalen Bitbreiten bzw. die minimale Speichergröße an. (5 Punkte)

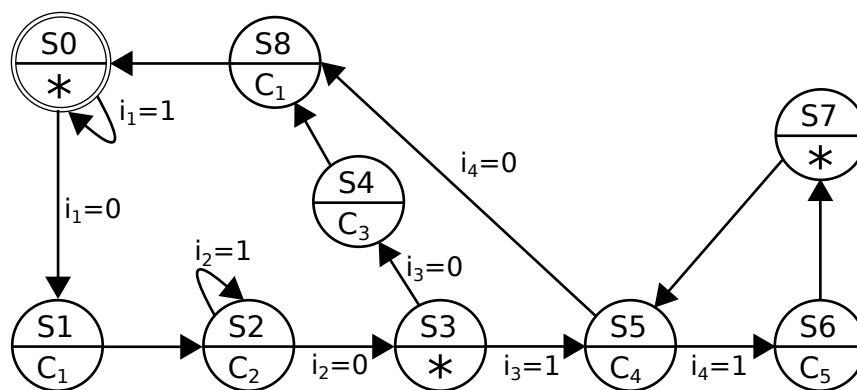


Abbildung 10: Automat

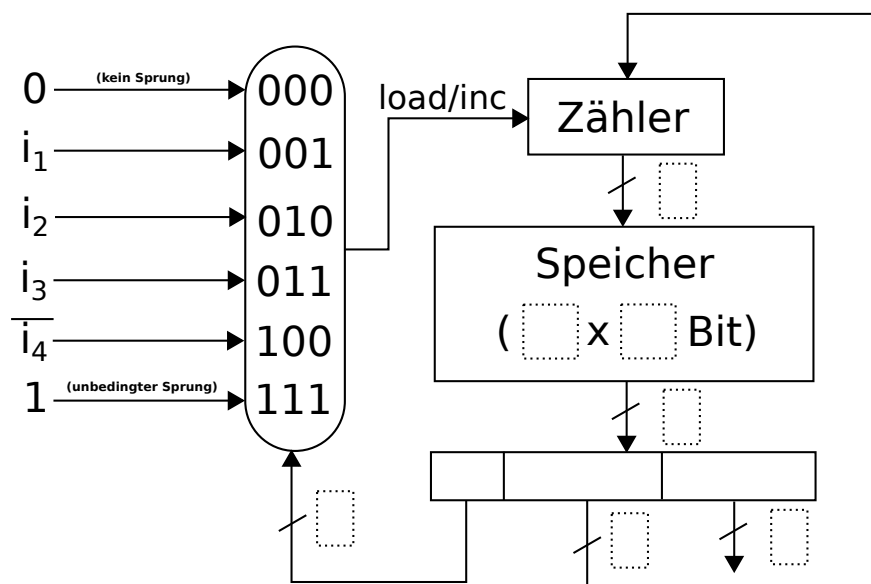


Abbildung 11: Struktur der speicherprogrammierten Steuerung

- e) Beschreiben Sie, was der Zähler bei  $\text{load/inc}=0$  und  $\text{load/inc}=1$  macht. (3 Punkte)

$\text{load/inc}=0$ :

---



---

$\text{load/inc} = 1$ :

---



---

- f) Geben Sie den exakten Speicherinhalt an, so dass der Automat in Abbildung 10 als speicherprogrammierte Steuerung realisiert wird. (7 Punkte)

Speicheradresse	Bedingungsauswahl	Sprungadresse	Steuersignale	Bemerkung

Tabelle 3: Speicherinhalt

Speicheradresse	Bedingungsauswahl	Sprungadresse	Steuersignale	Bemerkung

Tabelle 4: Ersatz Speicherinhalt, **ungültige Lösung streichen!**







