

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

Klausur
Digitaltechnik /
Grundlagen Technische Informatik

1. April 2010
Bearbeitungsdauer: 90 Minuten

Punkteverteilung							
Aufgabe	1	2	3	4	5	6	Σ
maximale Punkte	15	15	10	10	15	25	90
erreichte Punkte							

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

Aufkleber

Name:	
Matrikelnummer:	
Studienrichtung:	

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

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

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

Es sind keine Hilfsmittel zugelassen!

Aufgabe 1: (Quine-McCluskey)

15 Punkte

a) Gegeben sei die Einstellenmenge:

$$\mathcal{E} = \{0000, 0001, 0110, 0111, 1000, 1001, 1100, 1110, 1111\}$$

der Funktion $z = f(x_3, x_2, x_1, x_0)$. Ermitteln Sie mit Hilfe des Quine-McCluskey Verfahrens die Menge \mathcal{P} der Primimplikanten. (8 Punkte)

b) Gegeben sei eine Menge von Primimplikanten (\mathcal{P}_0 bis \mathcal{P}_4) durch:

$$\mathcal{P} = \{00 - 0, -000, 0 - 1-, - - 11, -1 - 1\}$$

Bestimmen Sie mit Hilfe nachfolgender Tabelle die minimale SOP-Form (**Sum Of Products**) der zugehörigen Funktion. (7 Punkte)

Minimale SOP-Form: _____

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

Tabellen zu b)

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

m												
\mathcal{P}_0												
\mathcal{P}_1												
\mathcal{P}_2												
\mathcal{P}_3												
\mathcal{P}_4												

Ersatztabellen zu b)
(ungültige Lösung streichen!)

Aufgabe 2: (VHDL)

15 Punkte

Auf der nächsten Seite finden Sie die VHDL Entity *Toplevel* sowie die zugehörige Architecture *Behavioral*. Die Architecture implementiert drei Prozesse, die auf die Ausgangssignale x , y , und z sowie auf das lokale Signal a schreibend zugreifen.

Abbildung 1 zeigt ein Timing-Diagramm für die Entity *Toplevel*. Die Werte der Eingangssignale clk , rst , und $input$ sind vorgegeben, die Signale x , y , a , und z starten mit dem Wert 0.

Vervollständigen Sie die Signalverläufe für die Signale x , y , a , und z in Abbildung 1.

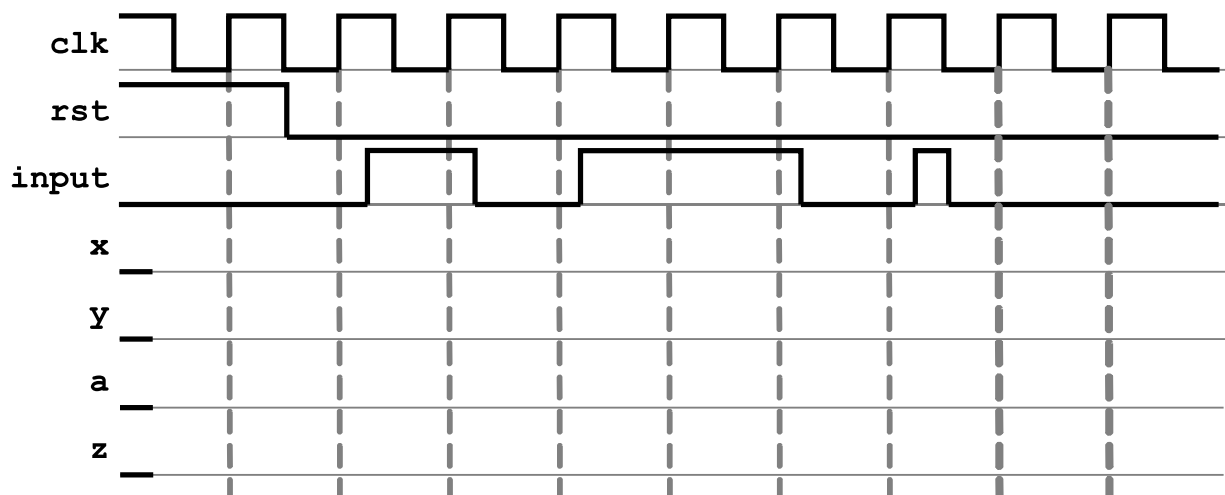
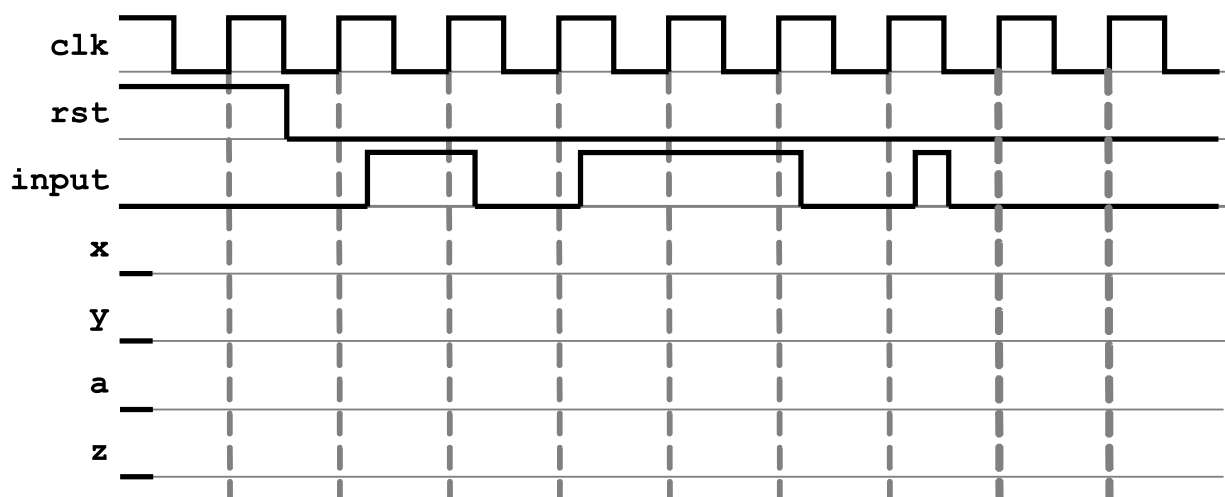


Abbildung 1: Timing-Diagramm für die Entity Toplevel



Timing-Diagramm für die Entity Toplevel (Ersatz)
(ungültige Lösung streichen!)

Entity Toplevel

```
entity Toplevel is
port(rst : in   std_logic;
      clk  : in   std_logic;
      input : in   std_logic;
      x     : out  std_logic;
      y     : out  std_logic;
      z     : out  std_logic);
end Toplevel;

architecture Behavioral of Toplevel is
  signal a : std_logic;
begin

  proc_x : process (rst , input)
  begin
    if rst = '1' then
      x <= '0';
    else
      x <= input;
    end if;
  end process;

  proc_y : process (clk , rst , input)
  begin
    if clk'event and clk = '1' then
      if rst = '1' then
        y <= '0';
      else
        y <= not input;
      end if;
    end if;
  end process;

  proc_{za} : process (clk , rst , input , a)
  begin
    if rst = '1' then
      a <= '0';
      z <= '0';
    else
      if clk'event and clk = '1' then
        a <= input;
        z <= a;
      end if;
    end if;
  end process;

end Behavioral;
```

Aufgabe 3: (Zahldarstellung)

10 Punkte

- a) Gegeben sei die Zahl 1001 in 4-bit Zweierkomplementdarstellung. Welchen Wert hat die Zahl?

Kreuzen Sie das richtige Ergebnis in folgender Tabelle an.

(1 Punkt)

- ☐ -7
☐ 9
☐ -1
☐ Keine Lösung ist richtig

- b) Gegeben sei wieder die Zahl 1001 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)

- ☐ 10001001
☐ 00001001
☐ 11111001
☐ Keine Lösung ist richtig

- c) Bei Umrechnung des Dezimalbruchs 0,7 in einen Binärbruch ergibt sich folgende Darstellung:

(2 Punkte)

- ☐ $0,10\overline{110}$ (0,1011001100....)
☐ 0,111
☐ $0,01\overline{001}$ (0,0100110011....)
☐ Keine Antwort ist richtig.

- d) Gegeben sei die Dezimalzahl 8,25. Wie sieht die Zahldarstellung im IEEE-754 Gleitkommaformat aus?

(2 Punkte)

Vorzeichen (1 bit): _____

Mantisse (23 bit): _____

Exponent (8 bit, Exzessdarstellung mit Verschiebungskonstante 127): _____

- e) In einem Rechnersystem soll die Übertragung von Daten (64 Informationsbits) zwischen den einzelnen Komponenten mit einem fehlerkorrigierenden Code abgesichert werden. Der Code soll 2-fach Fehler korrigieren können. Wie groß muss der Minimumabstand für den Code gewählt werden?

Kreuzen Sie in der Tabelle alle richtigen Antworten an:

(1 Punkt)

- ☐ 3
- ☐ 5
- ☐ Bei dieser Datenbreite ist keine Fehlerkorrektur mehr möglich, nur Fehlererkennung
- ☐ Keine Antwort ist richtig.

- f) Über eine unsichere Datenleitung sollen Worte bestehend aus 4 Informationsbits x_1, x_2, x_3, x_4 und 3 Prüfbits p_1, p_2, p_3 übertragen werden. Zur Kodierung wird ein Hammingcode verwendet. Es gilt:

$$p_1 = x_1 \oplus x_2 \oplus x_3$$

$$p_2 = x_1 \oplus x_2 \oplus x_4$$

$$p_3 = x_1 \oplus x_3 \oplus x_4$$

Dabei entspricht \oplus einer XOR-Verknüpfung. Die Information (x_1, x_2, x_3, x_4) wird als Codewort ($x_1, x_2, x_3, x_4, p_1, p_2, p_3$) übertragen. Bei einer Datenübertragung kommt das Wort ($x_1, x_2, x_3, x_4, p_1, p_2, p_3$) = (1,1,0,1,1,0,0) beim Empfänger an. Welche Aussagen in der Tabelle treffen zu?

Hinweis: Gehen Sie davon aus, dass höchstens 1 Bit gekippt ist.

(3 Punkte)

- ☐ Das übertragene Wort ist fehlerfrei.
- ☐ Bei der Datenübertragung ist ein Fehler passiert, das richtige Wort nach der Korrektur ist (1,0,0,1,1,0,0).
- ☐ Bei der Datenübertragung ist ein Fehler passiert, das richtige Wort nach der Korrektur ist (1,1,0,1,0,1,0).
- ☐ Keine Antwort ist richtig.

Aufgabe 4: (Fehlererkennung und -korrektur)

10 Punkte

Betrachten Sie die in Tabelle 1 dargestellte Sicherungsmethode.

	Datenwort				Parität
	0	1	1	0	0
	0	0	1	0	1
	1	1	0	0	0
	1	0	1	1	1
Parität	0	0	1	1	

Tabelle 1: Kreuzsicherung eines Datenblocks

Es handelt sich hierbei um die Kreuzsicherung, bei der Daten in einem rechteckigen Block angeordnet werden und anschließend die Parität über jede Zeile und jede Spalte gebildet wird.

- a) Welche Hammingdistanz hat der Code? Welche Eigenschaften für die Fehlerkorrektur und die Fehlererkennung (dieses Codes) lassen sich aus dem Hammingabstand ableiten?

Hinweis: Überlegen Sie, wie viele Codebits sich ändern, wenn man ein Informationsbit ändert. (3 Punkte)

- b) Das 16 Bit Datenwort soll nun, wie beim Hammingcode aus der Vorlesung, mit einer minimalen Zahl von Paritätsbits ergänzt werden, so dass Einfachfehler korrigiert werden können. Wie viele Paritätsbits werden mindestens benötigt? (1 Punkte)

- c) Anstatt des gesicherten Datenwortes aus Tabelle 1 werden die unten aufgeführten Daten empfangen (geänderte Bits sind unterstrichen). Wenden Sie darauf den folgenden Fehlerkorrekturalgorithmus an und kreuzen Sie das entsprechende Ergebnis an.
- Falls sowohl Spalten- als auch Zeilenparitäten zu den Informationsbits passen, nimm eine korrekte Übertragung an.
 - Falls nur eines der Paritätsbits nicht zu den Informationsbits passt, korrigiere nur das Paritätsbit und lasse alle übrigen Daten unangetastet.
 - Falls jeweils eine Zeilenparität und eine Spaltenparität nicht zu den Informationsbits passen und Zeile i und Spalte j betroffen sind, dann korrigiere den Dateneintrag an der Position (i,j) .
 - Alle übrigen Fehler in den Paritätsbits führen zu einer Fehlererkennung, aber zu keiner Korrektur.
- (6 Punkte)

	Datenwort				Par.
	0	1	1	0	0
	<u>1</u>	0	1	0	1
	<u>0</u>	1	0	0	0
	1	0	1	1	1
Parität	0	0	1	1	

Der Fehler in der Nachricht wird

nicht erkannt

☐

nur erkannt (nicht korrigiert)

☐

korrekt korrigiert

☐

falsch korrigiert

☐

	Datenwort				Par.
	0	1	1	0	0
	0	<u>1</u>	1	0	<u>0</u>
	1	1	0	0	0
	1	0	1	1	1
Parität	0	<u>1</u>	1	1	

Der Fehler in der Nachricht wird

nicht erkannt

☐

nur erkannt (nicht korrigiert)

☐

korrekt korrigiert

☐

falsch korrigiert

☐

	Datenwort				Par.
	0	1	1	0	0
	0	0	1	0	<u>0</u>
	1	1	0	0	0
	1	0	1	1	1
Parität	<u>1</u>	0	1	1	

Der Fehler in der Nachricht wird

nicht erkannt

☐

nur erkannt (nicht korrigiert)

☐

korrekt korrigiert

☐

falsch korrigiert

☐

	Datenwort				Par.
	0	1	1	0	0
	0	0	<u>0</u>	0	1
	1	1	0	0	0
	1	0	1	1	1
Parität	0	0	<u>0</u>	1	

Der Fehler in der Nachricht wird

nicht erkannt

☐

nur erkannt (nicht korrigiert)

☐

korrekt korrigiert

☐

falsch korrigiert

☐

Aufgabe 5: (Automat)

15 Punkte

Gegeben sei der durch Abbildung 1 spezifizierte Mealy-Automat, bei dem aber die Ausgaben fehlen. Der Automat erkennt eine spezielle Eingabefolge von Bits und gibt jedes Mal, wenn er in den Zustand S_3 oder S_5 gelangt, eine '1' aus. Damit signalisiert der Automat, dass eine gültige Bitfolge erkannt wurde. Ansonsten wird '0' ausgegeben.

- a) Ergänzen Sie die richtigen Ausgaben in Abbildung 1. (2 Punkte)

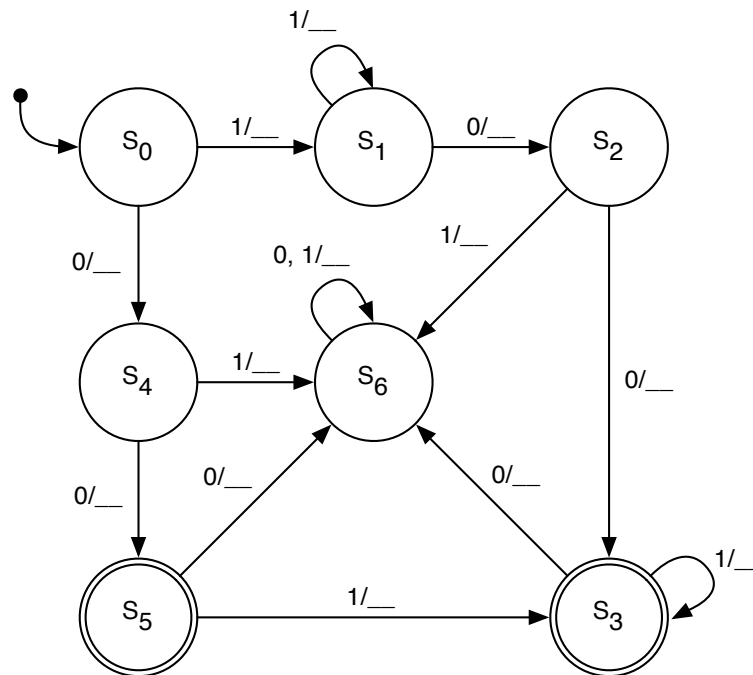


Abbildung 1: Mealy-Automat

- b) Füllen Sie die gegebene Automatentafel zu dem in Abbildung 1 spezifizierten Mealy-Automaten aus. (2 Punkte)

[illegible]

- c) Bestimmen Sie den äquivalenten zustandsminimalen Mealy-Automaten mit Hilfe des Ginsburg/Huffman Verfahrens (Benutzen Sie die gegebenen Tabellen). (6 Punkte)

δ/λ	0	1

δ/λ	0	1

δ/λ	0	1

δ/λ	0	1

Die Automatentafel des zustandsminimierten Automaten:

δ/λ	0	1

Ersatztabellen (ungültige Lösungen streichen!)

δ/λ	0	1

δ/λ	0	1

δ/λ	0	1

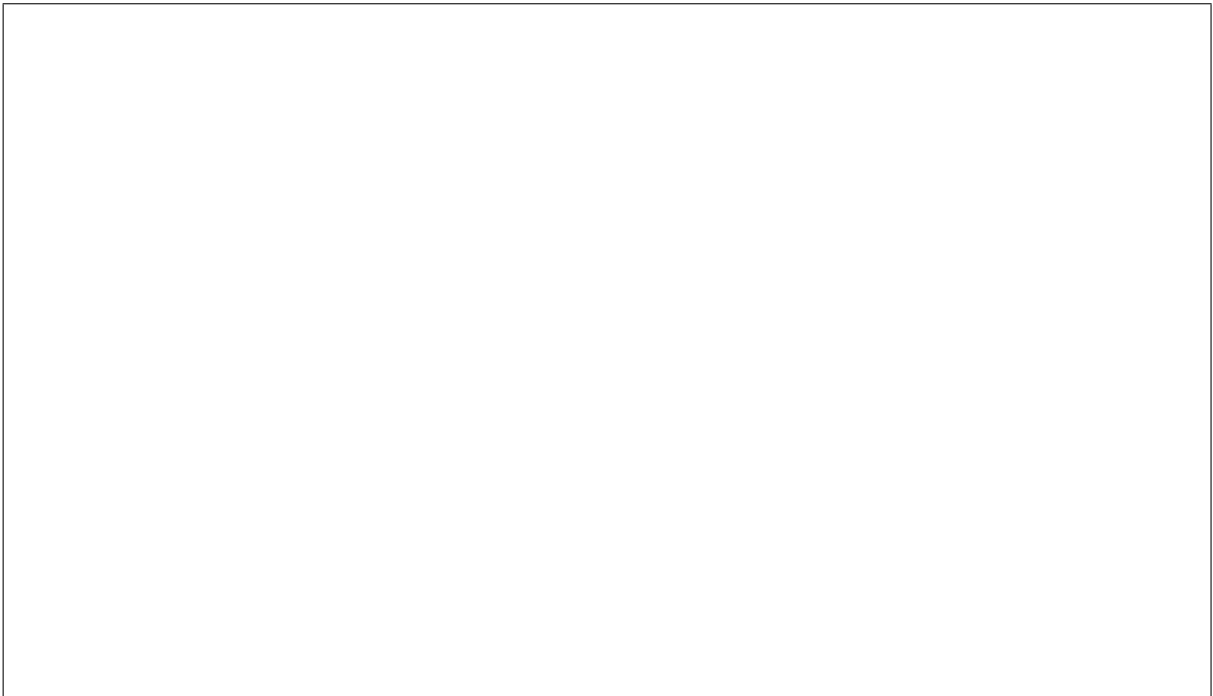
δ/λ	0	1

Die Automatentafel des zustandsminimierten Automaten (Ersatz):

δ/λ	0	1

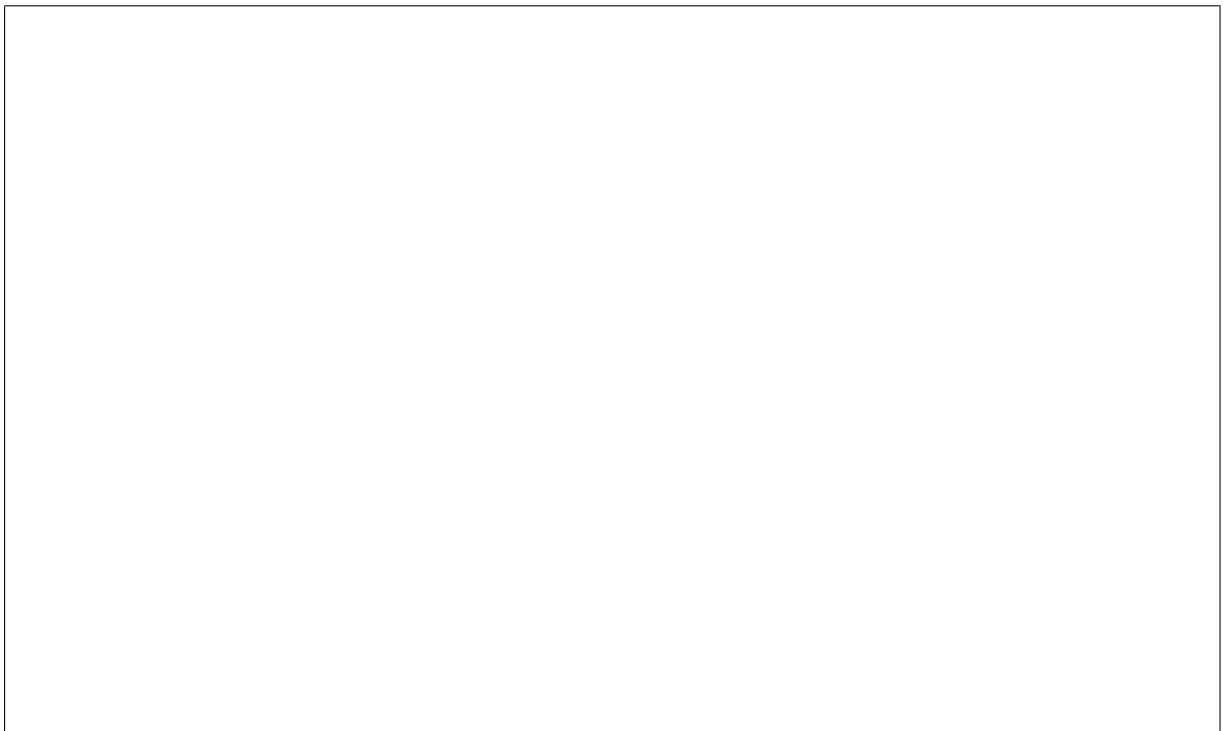
d) Zeichnen Sie den zustandsminimalen Mealy-Automaten:

(3 Punkte)



e) Wandeln Sie den zustandsminimalen Mealy-Automaten in einen Moore-Automaten um und zeichnen Sie diesen:

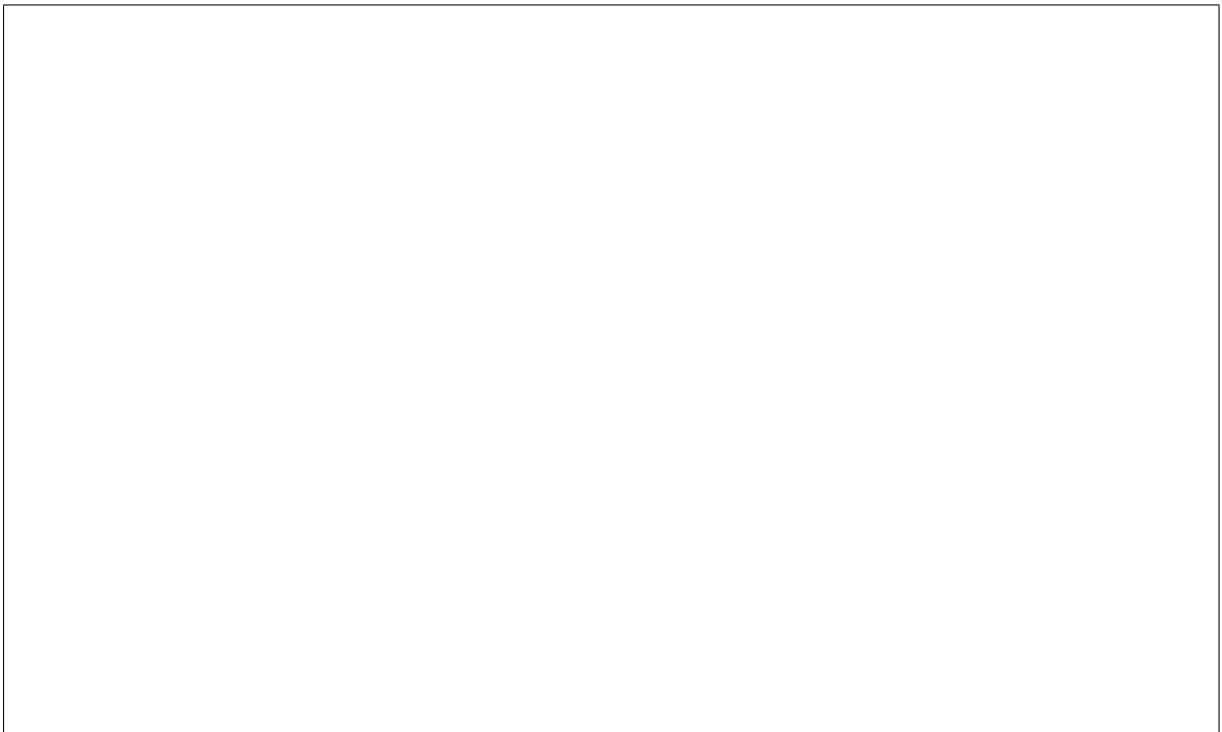
(2 Punkte)



Ersatztabellen (ungültige Lösungen streichen!)



_____ -Automat



_____ -Automat

Aufgabe 6: (RTL)

25 Punkte

- a) Für eine Hardwarekomponente sei bereits die Spezifikation der Steuerung in Form eines Mealy Automaten (vgl. Abbildung 1, nächste Seite) gegeben. Die Steuerung soll als festverdrahtetes Steuerwerk implementiert werden. Die Zustände sind laut Spezifikation mit 4 Bits folgendermaßen kodiert:

Zustand	Kodierung (Z_3, Z_2, Z_1, Z_0)
IDLE	0001
CHECK	0010
CALC	0100
WAIT	1000

Tabelle 1: Zustandskodierung

Wie groß ist die minimale Anzahl von Bits, die für die Kodierung entgegen der oben genannten Spezifikation ausreichen würde? (1 Punkt)

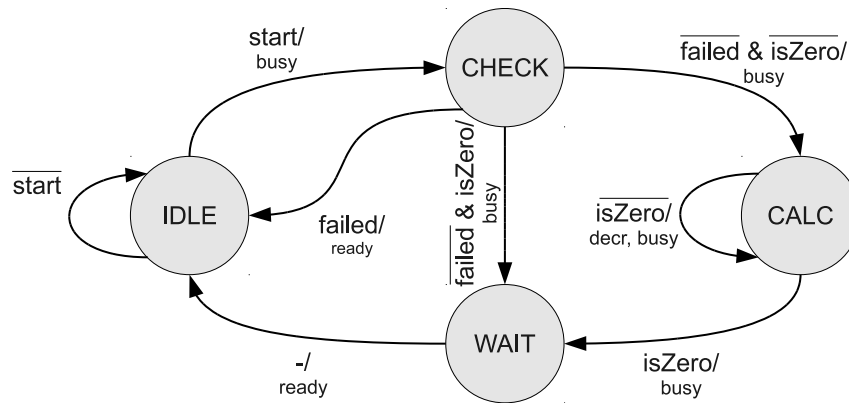


Abbildung 1: Zustandsübergangsgraph

- b) Vervollständigen Sie die Logiktable der Zustandsübergangsfunktion $\delta : Z \times I \rightarrow Z$ für den Automat in Abbildung 1 für ein festverdrahtetes Steuerwerk (Abbildung 2). Benennen sie die ersten drei Spalten der Tabelle. Steuerleitungen, die für den Zustandswechsel nicht benötigt werden, müssen mit X gekennzeichnet werden.

(4 Punkte)

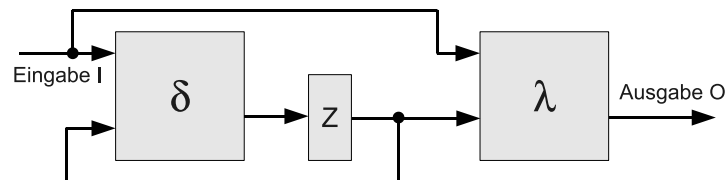


Abbildung 2: Festverdrahtetes Steuerwerk

			Z_3	Z_2	Z_1	Z_0	Z'_3	Z'_2	Z'_1	Z'_0
			0	0	0	1	0	0	0	1
			0	0	0	1	0	0	1	0
			0	0	1	0	1	0	0	0
			0	0	1	0	0	1	0	0
			0	0	1	0	0	0	0	1
			0	1	0	0	0	1	0	0
			0	1	0	0	1	0	0	0
			1	0	0	0	0	0	0	1

Tabelle 2: Logiktablelle

			Z_3	Z_2	Z_1	Z_0	Z'_3	Z'_2	Z'_1	Z'_0
			0	0	0	1	0	0	0	1
			0	0	0	1	0	0	1	0
			0	0	1	0	1	0	0	0
			0	0	1	0	0	1	0	0
			0	0	1	0	0	0	0	1
			0	1	0	0	0	1	0	0
			0	1	0	0	1	0	0	0
			1	0	0	0	0	0	0	1

Ersatztablelle 2: Logiktablelle (ungültige Lösung streichen!)

Im Folgenden soll nun eine Schaltung zur sequenziellen Berechnung der Fakultät einer ganzen Zahl $n \in \mathbb{N}_0$ erstellt werden: $n! = \prod_{i=1}^n i$ mit $0! \doteq 1$.

Zur Realisierung des Datenpfads stehen Ihnen außer den 8 Bit Registern **a** und **b** zwei Multiplexer und jeweils genau ein Multiplizierer (mult) und ein Subtrahierer (sub) zur Verfügung. Der Wert von n wird über einen 8 Bit Bus (INBUS) zur Verfügung gestellt. Das Ergebnis soll im Register **a** gespeichert und an einen zweiten 8 Bit breiten Bus (OUTBUS) ausgegeben werden. Die Multiplikation und die Subtraktion benötigen nur einen Taktzyklus.

- c) Welchen Wert darf n maximal annehmen, damit kein Überlauf entsteht, d.h. dass das Ergebnis der Fakultätsberechnung im Ergebnisregister **a** korrekt gespeichert werden kann. Begründen Sie Ihre Antwort. (1 Punkt)

- d) Vervollständigen Sie den Datenpfad in Abbildung 3 so, dass die Fakultät berechnet werden kann. Ergänzen Sie, falls notwendig, Konstanten an den Eingängen von Komponenten.

Hinweis: Bedenken Sie, dass $0! = 1$ ist.

(8 Punkte)

- e) Zeichnen Sie zur Realisierung der Steuerung in den Datenpfad (Abbildung 3) die notwendigen Kontrollpunkte ein und benennen Sie diese. Füllen Sie zusätzlich nachfolgende Tabelle 3 aus.

Hinweis: Bedenken Sie, dass die Steuerleitungen der Multiplexer ebenfalls Kontrollpunkte sind.

(3 Punkte)

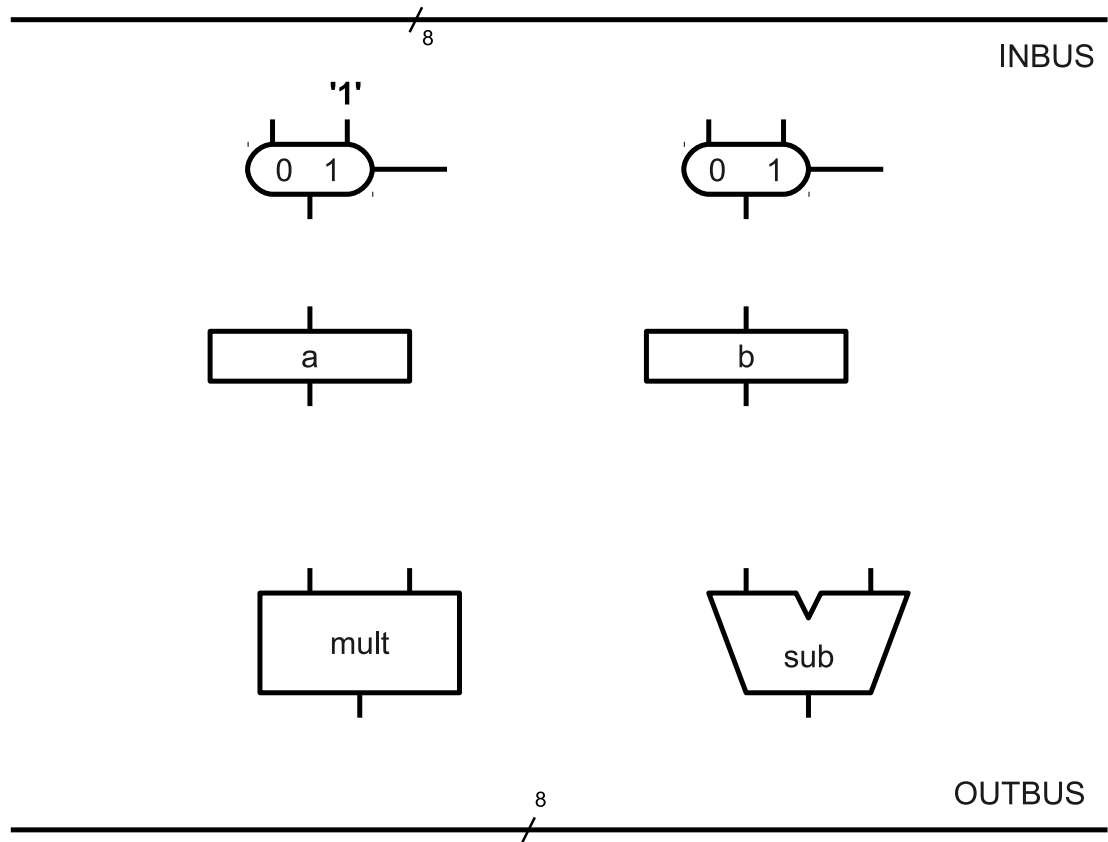
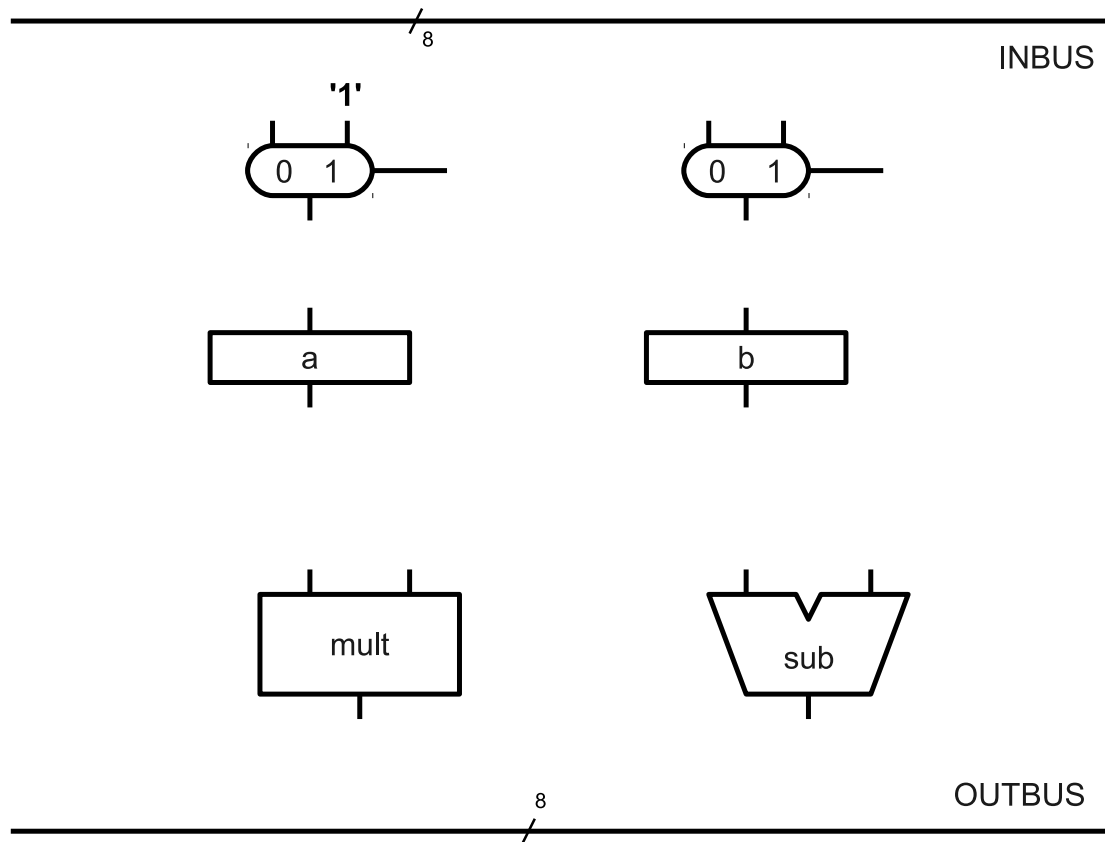


Abbildung 3: Datenpfad

Kontrollpunktname	Funktionsbeschreibung

Tabelle 3: Kontrollpunkttable



Ersatzabbildung 3: Datenpfad (ungültige Lösung streichen!)

Kontrollpunktname	Funktionsbeschreibung

Ersatztable 3: Kontrollpunkttable (ungültige Lösung streichen!)

- f) Die Steuerung besitzt zusätzlich zu den von Ihnen in Aufgabenteil e) festgelegten Signalen die Steuer- und Statusleitungen **start** und **ready**. Bei **start**=1 beginnt die Berechnung, **ready**=1 signalisiert das Ende der Berechnung. Zeichnen Sie den Zustandsübergangsgraph der Steuerung als endlichen Automat (Moore oder Mealy) in den dafür vorgesehenen Freiraum (Abbildung 4). (8 Punkte)



Abbildung 4: Zustandsübergangsgraph



Ersatzabbildung 4: Zustandsübergangsgraph (ungültige Lösung streichen!)

