

Klausur zur Vorlesung
Digitaltechnik (DT)

Prof. Marco Platzner
Fachgebiet Technische Informatik
Universität Paderborn

27.03.2019

- 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	15	25	15	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) Die Boolesche Funktionsalgebra F_3 ... (3 Punkte)

- ☐ hat 8 Elemente in ihrer Trägermenge.
- ☐ hat 256 Elemente in ihrer Trägermenge.
- ☐ hat alle Booleschen Funktionen von bis zu 3 Variablen in ihrer Trägermenge.
- ☐ definiert keine Inverse für die Null-Funktion.

(b) Ein Maxterm $M_i(x_1, x_2, \dots, x_n)$, $n > 1$, ... (3 Punkte)

- ☐ kann durch ein n -OR Gatter und optionale Inverter realisiert werden.
- ☐ hat immer mehr Literale als ein Minterm $m_i(x_1, x_2, \dots, x_n)$.
- ☐ ist eine Funktion deren Wahrheitstabelle '1'-en in jeder Zeile hat.
- ☐ hat immer mehr '1'-en als ein Minterm $m_i(x_1, x_2, \dots, x_n)$.

(c) Ein $(\text{mod } 2^n)$ Zähler, der als asynchrone Schaltung realisiert ist, gibt zyklisch folgende Sequenz von Worten aus: $0, 1, 2, \dots, 63$. (3 Punkte)

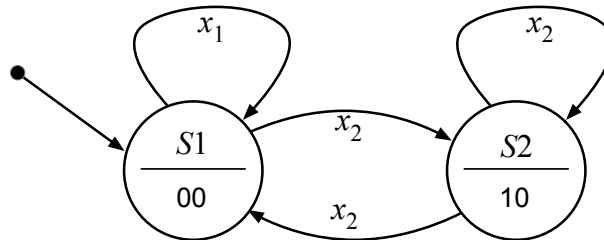
- ☐ Der Zähler hat genau 5 Flip-Flops.
- ☐ Der Zähler ist ein Schieberegister mit 6 Flip-Flops.
- ☐ Der Zähler hat genau 6 D-Latches.
- ☐ Alle Speicherelemente des Zählers werden mit '0' initialisiert.

NAME:

Matrikelnummer:

- (d) Der folgende Moore-Automat $A = \{X, Y, S, \delta, \mu, s_I\}$ mit $X = \{x_1, x_2\}$...

(3 Punkte)



- ☐ ist vollständig.
- ☐ ist deterministisch.
- ☐ ist vollreduziert.

- (e) Die Parität eines Codewortes $X = (x_1, x_2, \dots, x_n)$...

(3 Punkte)

- ☐ ändert sich, wenn man ein Bit von '0' auf '1' ändert.
- ☐ kann man aus der Anzahl von '0'-Bits ableiten.
- ☐ ist ungerade, wenn $\text{XOR}(x_1, x_2, \dots, x_n) = '0'$.

Aufgabe 2 (VHDL)

[15 Punkte]

Abbildung 1 zeigt die Schnittstellen einer Kaffeemaschinensteuerung, welche Sie entwickeln sollen. Die Steuerung hat vier Eingänge vom Typ *std_logic*: einen Takteingang (*clk*), einen Eingang zur Auswahl der Kaffeestärke (*w*), einen Eingang zum Bestätigen der Auswahl (*b*) und einen Eingang, um das Abschließen des Brühvorgangs zu überprüfen (*d*). Dazu kommen zwei Ausgänge vom Typ *std_logic*: einen Ausgang, um eine valide Auswahl anzuzeigen (*v*) und einen Ausgang zum Ausgeben der Auswahl (*a*).

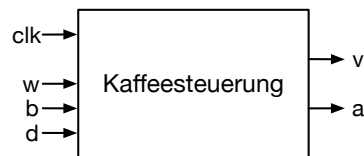


Abbildung 1: Entity der Kaffeemaschinensteuerung

Abbildung 2 zeigt den ereignisgesteuerten Zustandsautomaten der Steuerung. Zur Vereinfachung sind die Transitionen nur mit den jeweils relevanten Eingängen versehen. Die Maschine wartet im Zustand *IDLE* auf einen Kunden, der den Prozess startet. Der Kunde hat die Wahl zwischen einem schwachen und einem starken Kaffee. Nachdem die Auswahl bestätigt wurde, beginnt der Brühvorgang.

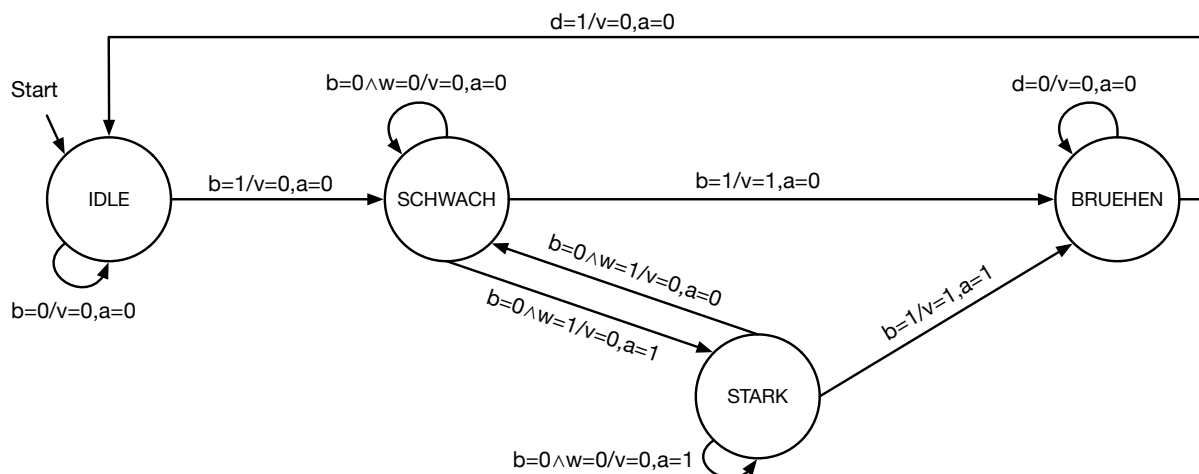


Abbildung 2: Ereignisgesteuerter Zustandsautomat der Kaffeemaschinensteuerung

- (a) Handelt es sich bei dem angegebenen Zustandsautomaten um einen Mealy- oder einen Moore-Automaten? (1 Punkt)

☐ Mealy

☐ Moore

- (b) Vervollständigen Sie den VHDL Code in Listing 1, sodass die oben beschriebene Funktionalität realisiert wird. (14 Punkte)

Listing 1: VHDL Code der *Kaffeesteuerung*

```

architecture Behavioral of Kaffeesteuerung is
  type state_t is ( _____
                    _____
                    _____
                    _____);

  signal state : state_t := _____;
  signal next_state : state_t := IDLE;
  signal v_i, a_i : std_logic := '0';
begin
  proc_state : process(_____) begin
    if rising_edge(clk) then
      state <= next_state;
      v <= v_i; a <= a_i;
    end if;
  end process proc_state;

  proc_next_state : process(_____) begin
    v_i <= '0'; a_i <= '0';

    _____

    case state is
      when IDLE =>
        if _____ then
          _____
          _____
          _____
        else
          _____
          _____
          _____
        end if;
      when SCHWACH =>
        if _____ then
          _____
          _____
          _____
        elsif _____ then
          _____
          _____
          _____
        else
          _____
          _____
          _____
        end if;

```

```
when STARK =>
    if _____ then
        _____
        _____
    elsif _____ then
        _____
        _____
        _____
    else
        _____
        _____
        _____
    end if;
when BRUEHEN =>
    if _____ then
        _____
        _____
        _____
    else
        _____
        _____
        _____
    end if;
when others =>
    next_state <= IDLE;
end case;
end process proc_next_state;
end Behavioral;
```

Listing 2: Ersatz-VHDL Code der *Kaffeesteuerung*, **ungültige Lösung streichen!**

```

architecture Behavioral of Kaffeesteuerung is
    type state_t is ( _____
                      _____
                      _____
                      _____);

    signal state : state_t := _____;
    signal next_state : state_t := IDLE;
    signal v_i, a_i : std_logic := '0';
begin
    proc_state : process(_____) begin
        if rising_edge(clk) then
            state <= next_state;
            v <= v_i; a <= a_i;
        end if;
    end process proc_state;

    proc_next_state : process(_____) begin
        v_i <= '0'; a_i <= '0';

        _____

        case state is
            when IDLE =>
                if _____ then
                    _____
                    _____
                    _____
                else
                    _____
                    _____
                    _____
                end if;
            when SCHWACH =>
                if _____ then
                    _____
                    _____
                    _____
                elsif _____ then
                    _____
                    _____
                    _____
                else
                    _____
                    _____
                    _____
                end if;

```

```
when STARK =>
    if _____ then
        _____
        _____
    elsif _____ then
        _____
        _____
        _____
    else
        _____
        _____
        _____
    end if;
when BRUEHEN =>
    if _____ then
        _____
        _____
        _____
    else
        _____
        _____
        _____
    end if;
when others =>
    next_state <= IDLE;
end case;
end process proc_next_state;
end Behavioral;
```


NAME:

Matrikelnummer:

Aufgabe 3 (Logikminimierung)

[25 Punkte]

Gegeben sei folgende Wahrheitstabelle einer Logikschaltung mit den Eingängen a , b , c , d , und Ausgängen x_1 und x_2 .

a	b	c	d	x_1	x_2
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	1	0

NAME:

Matrikelnummer:

- (a) Berechnen Sie eine minimale Summe von Primimplikanten für x_1 mithilfe des Quine-McCluskey Verfahrens. (15 Punkte)

Phase 1:

L_1

L_2

L_3

Gruppe	Codewort	✓
Grp. 0		
Grp. 1		
Grp. 2		
Grp. 3		
Grp. 4		

Gruppe	Codewort	✓
Grp. 0		
Grp. 1		
Grp. 2		
Grp. 3		
Grp. 4		

Gruppe	Codewort	✓
Grp. 0		
Grp. 1		
Grp. 2		
Grp. 3		
Grp. 4		

Geben Sie die Menge der Primimplikanten in Form von Codeworten an:

{ _____ }

Ersatz für Phase 1 (**Ungültige Lösung streichen**, sonst sind beide Lösungen ungültig!):

L_1			L_2			L_3		
Gruppe	Codewort	✓	Gruppe	Codewort	✓	Gruppe	Codewort	✓
Grp. 0			Grp. 0			Grp. 0		
Grp.1			Grp.1			Grp.1		
Grp. 2			Grp. 2			Grp. 2		
Grp. 3			Grp. 3			Grp. 3		
Grp. 4			Grp. 4			Grp. 4		

Geben Sie die Menge der Primimplikanten in Form von Codeworten an:

{ _____ }

NAME:

Matrikelnummer:

Phase 2:

Überdeckungstabelle:

P	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$

Ersatz für Überdeckungstabelle (**Ungültige Lösung streichen**, sonst sind beide Lösungen ungültig!):

P	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$	$m_{__}$

Geben Sie die im ersten Schritt gefundenen essentiellen Primimplikanten in Form von Codeworten an:

{ _____ }

Verbleiben nach dem Entfernen der Zeilen essentieller Primimplikanten (und der von ihnen überdeckten Spalten) noch Primimplikanten? ☐ Ja ☐ Nein

Geben Sie abschließend die gefundene minimale SOP-Form als Logikfunktion mit Variablen a , b , c , d an:

- (b) Beim Ermitteln einer minimalen Menge von Primimplikanten für x_2 hat ihr Kollege in Phase 1 des Verfahrens einen Fehler gemacht, und mehrere Implikanten als Primimplikanten aufgenommen, die keine sind. Zeigen Sie, dass in Phase 2 des Quine-McCluskey Verfahrens trotzdem eine minimale Menge von Primimplikanten gefunden wird. Füllen Sie hierfür die folgende Überdeckungstabelle aus und beantworten Sie die Fragen auf der folgenden Seite, indem Sie jeweils Mengen von Codewörtern angeben. (10 Punkte)

P'	m_0	m_4	m_7	m_8	m_{12}	m_{14}
	0000	0100	0111	1000	1100	1110
1110						
0-00						
0111						
11-0						
1-00						
- -00						

Ersatz (**Ungültige Lösung streichen**, sonst sind beide Lösungen ungültig!):

P'	m_0	m_4	m_7	m_8	m_{12}	m_{14}
	0000	0100	0111	1000	1100	1110
1110						
0-00						
0111						
11-0						
1-00						
- -00						

NAME:

Matrikelnummer:

Geben Sie die im ersten Schritt von Phase 2 des Verfahrens gefundenen essentiellen Primimplikanten an:

{ _____ }

Geben Sie die dominierten Zeilen an:

{ _____ }

Schlussfolgern Sie die nach dem Löschen der dominierten Zeilen resultierenden essentiellen Primimplikanten:

{ _____ }

Welche der sechs von ihrem Kollegen angegebenen Implikanten waren keine Primimplikanten?

{ _____ }

Aufgabe 4 (Zahlendarstellung)

[15 Punkte]

- (a) Gegeben sei die 2-stellige Dezimalzahl 53_{10} . Geben Sie die Zahl als Binärzahl ohne Vorzeichen an. (3 Punkte)

Lösung: _____

- (b) Gegeben sei die 6-stellige Binärzahl 101100_{2C} in 2er-Komplement Darstellung. Geben Sie die Zahl als 6-stellige Binärzahl in Vorzeichen/Betrag Darstellung an. (3 Punkte)

Lösung: _____

- (c) Gegeben sei die 8-stellige Binärzahl $10110011_{Excess-32}$ in Excess-32 Darstellung. Geben Sie die Zahl als Dezimalzahl an. (3 Punkte)

Lösung: _____

NAME:

Matrikelnummer:

- (d) Gegeben sei die 4-stellige Oktalzahl 3175_8 . Geben Sie die Zahl zur Basis 16 an. (3 Punkte)

Lösung: _____

- (e) Gegeben sei die 32-stellige Binärzahl $10111111111000000000000000000000_2$ im IEEE 754 Gleitkommaformat. Geben Sie die Zahl als Dezimalzahl an. (3 Punkte)

Lösung: _____

Aufgabe 5 (Sequentieller Multiplizierer)

[20 Punkte]

Gegeben sei folgender sequentieller Multiplizierer (Abbildung 3), zur Multiplikation zweier vorzeichenloser Zahlen X und Y in Binärdarstellung.

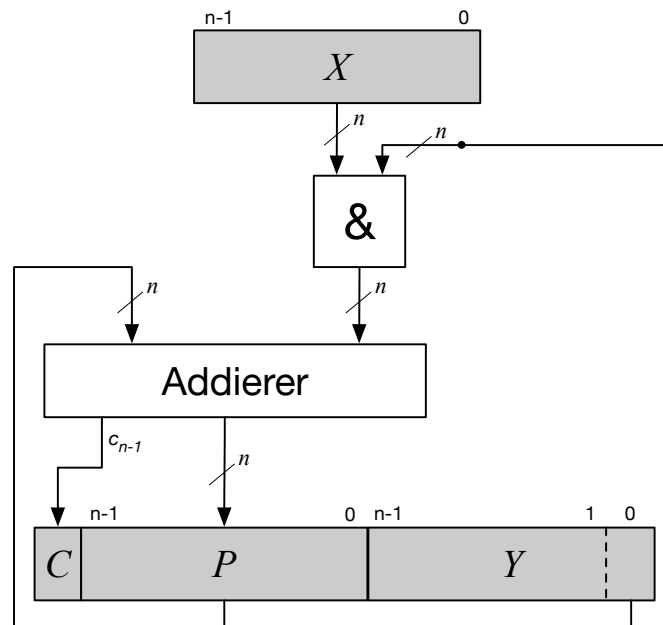


Abbildung 3: Sequentieller Multiplizierer

(a) Geben Sie die Schritte des Algorithmus zur Multiplikation an.

(4 Punkte)

1. Operandenregister X, Y laden

2. $C \leftarrow 0, P \leftarrow 0$

3.

4. Resultat steht in (P, Y)

Algorithmus zum sequentiellen Multiplizieren

NAME:

Matrikelnummer:

1. Operandenregister X, Y laden

2. $C \leftarrow 0, P \leftarrow 0$

3.

4. Resultat steht in (P, Y)

Ersatzschema für den Algorithmus, **ungültige Lösung streichen!**

- (b) Führen Sie nun die (vorzeichenlose) Multiplikation $Z_2 = X_2 \cdot Y_2$ mit Hilfe des in Abbildung 3 angegebenen sequentiellen Multiplizierers für die beiden vorzeichenlosen Zahlen

$X_2 = 011110_2$ und $Y_2 = 100011_2$ durch.

Füllen Sie hierzu die Tabelle 1 aus und geben Sie das Ergebnis Z_2 an. (12 Punkte)

	C	P					Y					Beschreibung
												Initialisieren
+												
+												
+												
+												
+												
+												
+												

Tabelle 1: Einzelschritte der sequentiellen Multiplikation

$$Z_2 = \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{}$$

NAME:

Matrikelnummer:

	C	P						Y						Beschreibung
														Initialisieren
+														
+														
+														
+														
+														
+														
+														
+														

Tabelle 2: Ersatzschema für (b), **ungültige Lösung streichen!**

$$Z_2 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline & & & & & & & & & & & & \\ \hline \end{array}$$

Ersatztable für (b), **ungültige Lösung streichen!**

- (c) Berechnen Sie zur Kontrolle die Dezimaldarstellung der Faktoren X , Y und des Ergebnisses der Multiplikation Z aus (b). (4 Punkte)

$$X_{10} = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{} \quad Y_{10} = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{} \quad Z_{10} = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}$$

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!