

Klausur zur Vorlesung

Grundlagen der Technischen Informatik / Digitaltechnik (GTI/DT)

Prof. Marco Platzner
Fachgebiet Technische Informatik
Universität Paderborn

02.08.2016

- 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 | 15 | 25 | 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) Welche der folgenden Aussagen sind in der Booleschen Algebra gültig?

☐ $a \cdot (a + b) = a \cdot b$

☐ $\overline{(a \cdot b)} \cdot c = \overline{(a \cdot c)} \cdot b$

☐ $\forall a \in B : \exists \bar{a} : a + \bar{a} = 0$

☐ $a \cdot b + a \cdot \bar{b} = a$

(b) Sind zwei Zustände s_1 und s_2 eines deterministischen Mealy-Automaten äquivalent ($s_1 \equiv s_2$), dann gilt:

☐ $s_1 \equiv_{\tau} s_2$

☐ $\forall x \in X : \lambda(s_1, x) = \lambda(s_2, x)$

☐ $\forall x \in X : |\delta(s_1, x)| \leq 1$

☐ $\forall x \in X : \delta(s_1, x) \equiv \delta(s_2, x)$

(c) Bei einem Code mit der Hammingdistanz von 5 sind folgenden Aussagen korrekt:

☐ Man kann 5 Fehler erkennen.

☐ Die Hammingdistanz zweier beliebiger Codeworte ist mindestens 5.

☐ Man kann 2 Fehler korrigieren.

NAME:

Matrikelnummer:

(d) Welche Aussagen sind bzgl. der linearen Kaskadierung von Komponenten der RTL-Ebene korrekt?

- ☐ Ein Ripple-Borrow Subtrahierer entsteht durch lineare Kaskadierung.
- ☐ Multiplexer lassen sich linear kaskadieren.
- ☐ Parallelregister lassen sich linear kaskadieren.
- ☐ Schieberegister lassen sich linear kaskadieren.

(e) Welche Aussagen treffen auf Halbleiter zu?

- ☐ Die Bandlücke zwischen Leitungs- und Valenzband ist größer als die von Isolatoren, aber kleiner als die von Leitern.
- ☐ Keramiken sind Halbleiter.
- ☐ Durch Einbringen von Fremdatomen (Dotierung) wird die Leitfähigkeit von Halbleitern gezielt verändert.
- ☐ Aufgrund thermischer Energie gelangen bei Raumtemperatur keine Elektronen ins Leitungsband.

Aufgabe 2 (VHDL)

[20 Punkte]

- (a) Im Folgenden sind Ihnen drei VHDL-Quelltexte (Q1 bis Q3), drei Blockschaltbilder (B1 bis B3) und drei Signalverläufe (S1 bis S3) gegeben. Welches Blockschaltbild und welcher Signalverlauf passt zu welchem Quelltext? Wenn kein Blockschaltbild oder kein Signalverlauf passt, dann streichen Sie bitte das entsprechende Tabellenfeld durch!

| Quelltext | Blockschaltbild | Signalverlauf |
|-----------|-----------------|---------------|
| Q1 | | |
| Q2 | | |
| Q3 | | |

- (b) In den Quelltexten Q1, Q2 und Q3 sind in den jeweiligen Zeilen 4, 5 und 7 die Sensitivitätslisten angegeben. Die Listen enthalten überflüssige Signale, die für eine korrekte Simulation nicht nötig sind. Welche sind das?

Q1: _____

Q2: _____

Q3: _____

Q1

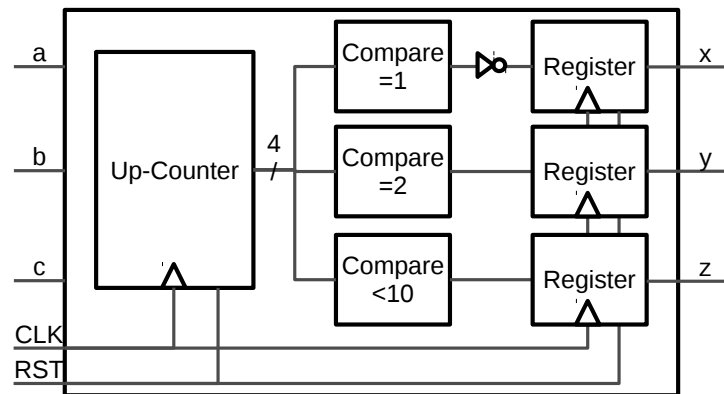
```
1 architecture Q1 of Test_Q1 is
2 begin
3
4     a_proc : process (CLK, RST, a, b, c) is
5     begin
6         x <= not a;
7         y <=    b;
8         z <=    '1';
9     end process;
10 end architecture;
```

Q2

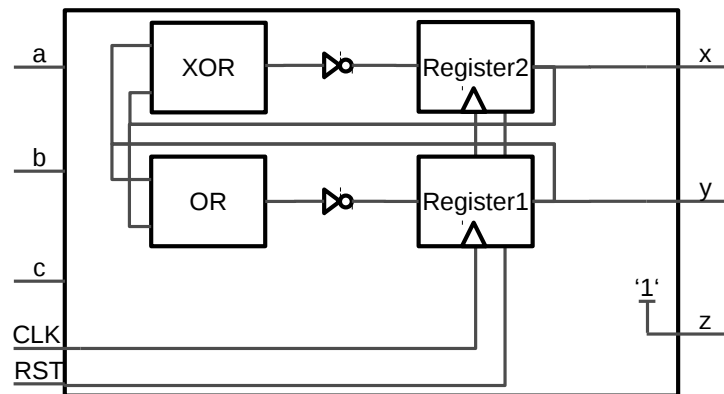
```
1 architecture Q2 of Test_Q2 is
2     signal i:integer range 0 to 15;
3 begin
4
5     a_proc : process (CLK, RST, a, b, c) is
6     begin
7         if RST='1' then
8             i <= 0; x <= '1'; y <= '0'; z <= '1';
9         elsif CLK'event and CLK='1' then
10            i <= i+1;
11            if i = 1 then x <= '0'; else x <= '1'; end if;
12            if i = 2 then y <= '1'; else y <= '0'; end if;
13            if i < 10 then z <= '1'; else z <= '0'; end if;
14        end if;
15    end process;
16 end architecture;
```

Q3

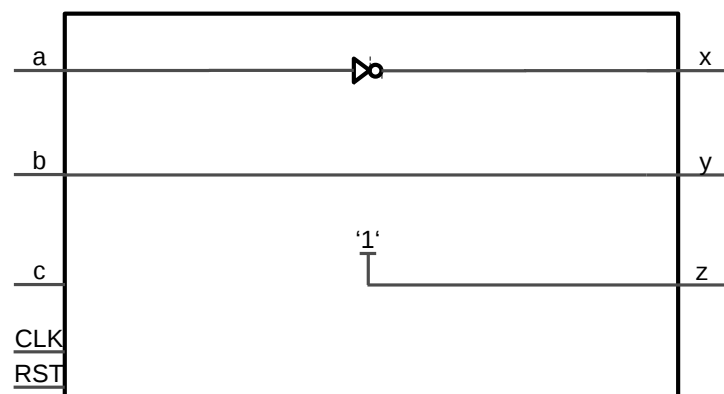
```
1 architecture Q3 of Test_Q3 is
2     signal i: std_logic_vector(2 downto 0);
3 begin
4
5     x <= i(2); y <= i(1); z <= i(0);
6
7     a_proc : process (CLK, RST, a, b, c) is
8     begin
9         if RST = '1' then
10            i <= "101";
11        elsif (CLK'event and CLK = '1') then
12            case i is
13                when "101" => i <= "001";
14                when "001" => i <= "111";
15                when "111" => i <= "101";
16                when others => i <= "101";
17            end case;
18        end if;
19    end process;
20 end architecture;
```



B1



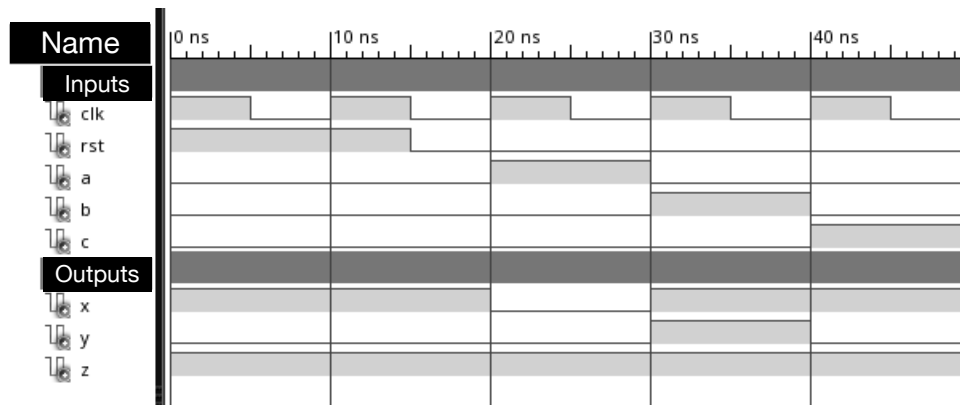
B2



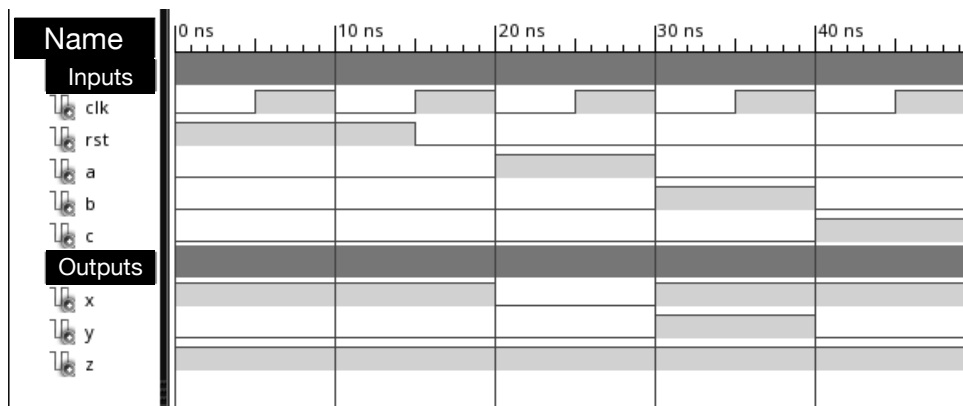
B3

NAME:

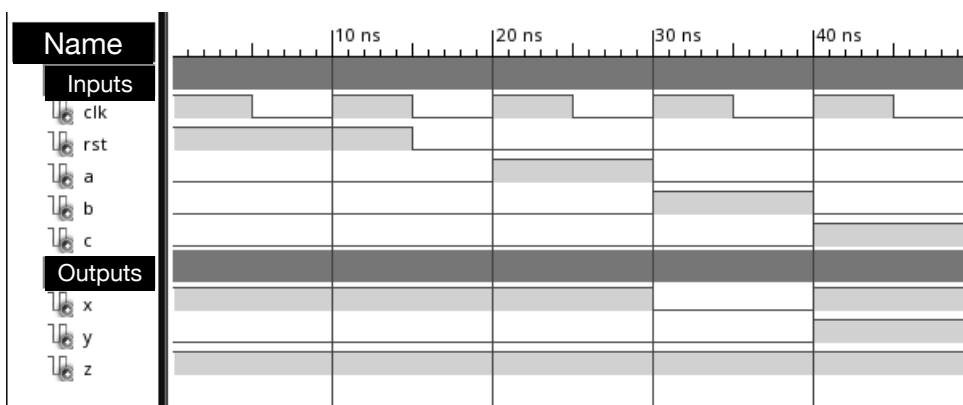
Matrikelnummer:



S1



S2



S3

Aufgabe 3 (Logikminimierung)

[15 Punkte]

In dieser Aufgabe soll ein 2-Bit Komparator implementiert werden.

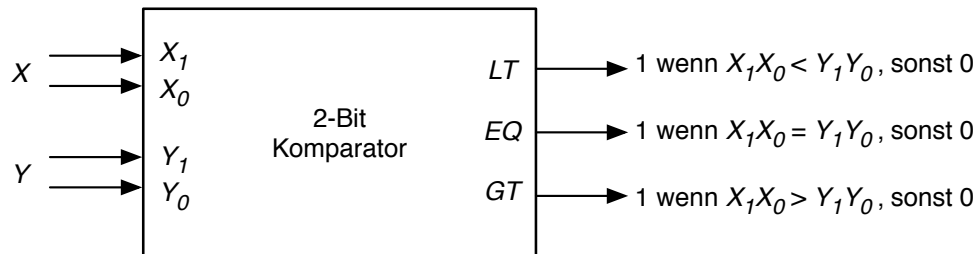


Abbildung 7: 2-Bit Komparator

Ein 2-Bit Komparator nimmt zwei 2-Bit Worte X und Y entgegen, die aus den Bits X_1X_0 und Y_1Y_0 bestehen. Die Komponente belegt immer nur einen der Ausgänge LT (less than), EQ (equals), GT (greater than) mit 1 abhängig davon, ob X kleiner als, gleich oder größer als Y ist. Die anderen Ausgänge sind jeweils mit 0 belegt, wie in Abbildung 7 dargestellt.

- (a) Füllen Sie die folgende Wahrheitstabelle mit den Booleschen Werten für den 2-Bit Komparator aus.

| X_1 | X_0 | Y_1 | Y_0 | LT | EQ | GT |
|-------|-------|-------|-------|------|------|------|
| 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:

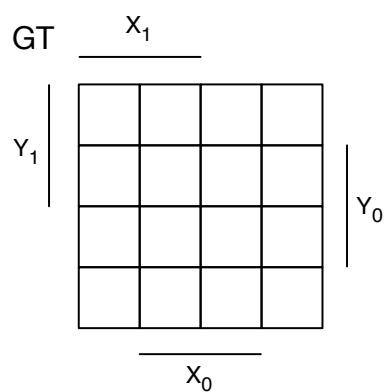
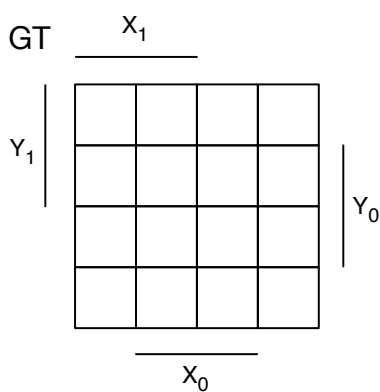
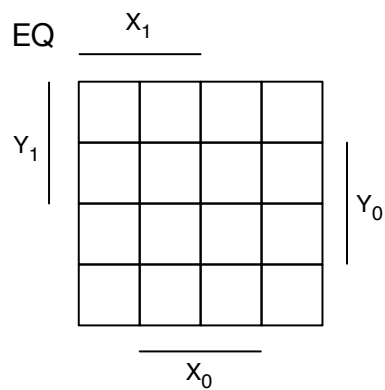
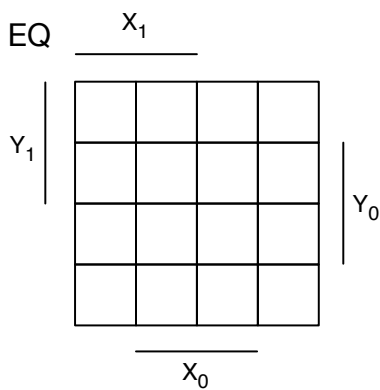
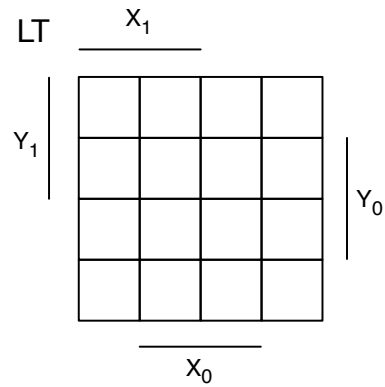
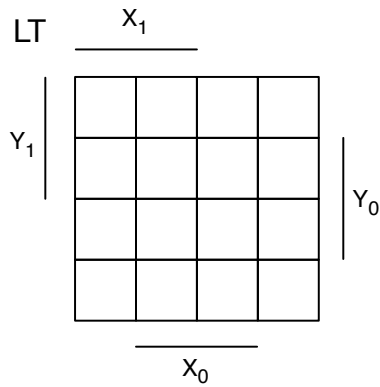
Matrikelnummer:

Ersatztable. Bitte ungültige Lösung streichen!

| X_1 | X_0 | Y_1 | Y_0 | LT | EQ | GT |
|-------|-------|-------|-------|------|------|------|
| 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 LT , EQ und GT mit Hilfe der unten angegebenen Karnaugh-Diagramme.

Ersatzdiagramme, ungültige Lösung streichen!



NAME:

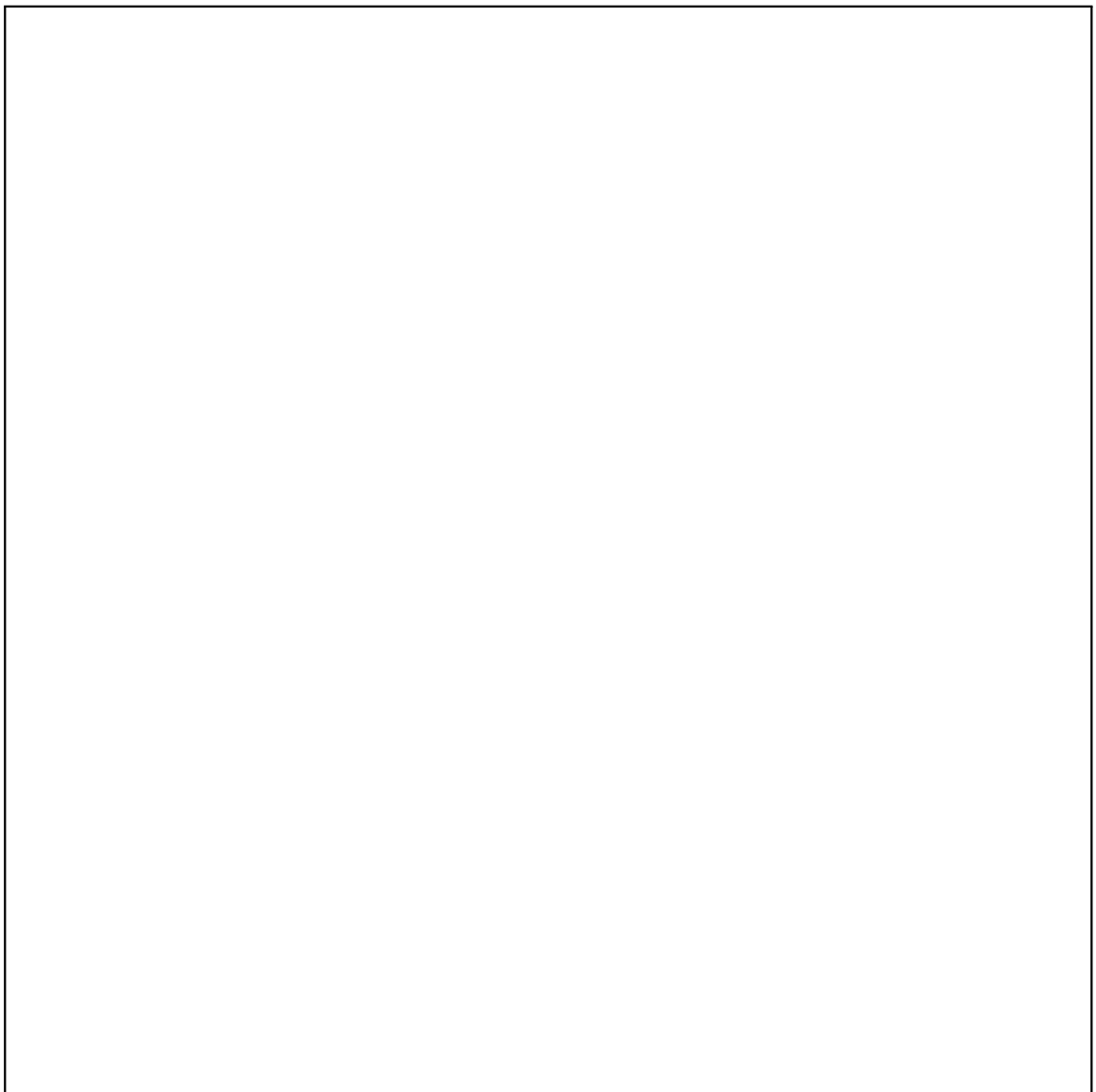
Matrikelnummer:

$LT =$ _____

$EQ =$ _____

$GT =$ _____

- (c) Entwerfen Sie eine Schaltung für den 2-bit Komparator. Benutzen Sie nur NAND Gatter und Inverter.



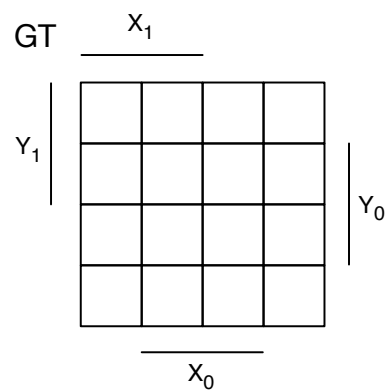
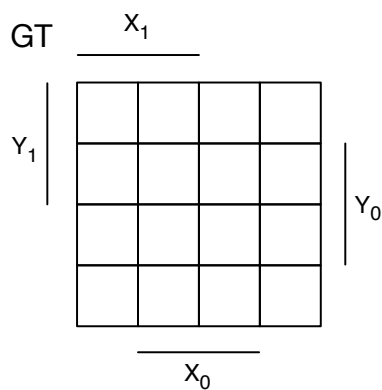
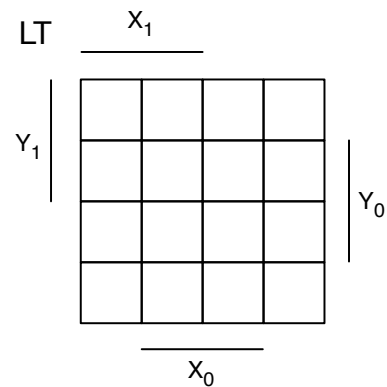
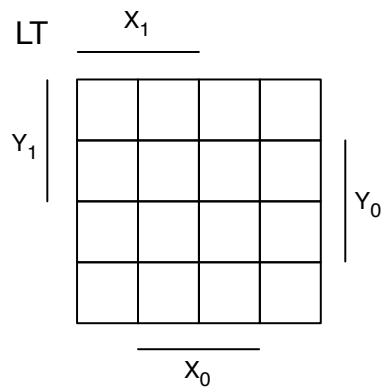
Ersatzdiagramm. Bitte ungültige Lösung streichen!

NAME:

Matrikelnummer:

- (d) Gehen Sie nun davon aus, dass gleiche Eingangswerte für X und Y ausgeschlossen werden. Bestimmen Sie die minimalen SOP-Formen für LT und GT für die dadurch vereinfachte Schaltung mithilfe der unten angegebenen Karnaugh Diagramme.

Ersatzdiagramme, ungültige Lösung streichen!



$LT =$ _____

$GT =$ _____

Aufgabe 4 (Vorzeichenbehaftete Zahlen und Darstellungsarten) [15 Punkte]

Seien $X = 0101_{2C}$ und $Y = 1010_{2C}$ zwei 4-bit Zahlen im Zweierkomplement.

- (a) Geben Sie die Zahlen X und Y in der Dezimaldarstellung an.

- (b) Geben Sie die Zahlen X und Y in der binären Vorzeichen/Betrag Darstellung als 4-bit Zahlen an.

- (c) Sei A ein 4-bit Register. Berechnen Sie $A_{2C} = X_{2C} + Y_{2C}$ im Zweierkomplement mittels der Addition nach der Schulmethode. Kommt es zu einem overflow? Wenn ja, welche Folge hat das für den Wert im Register A ?

NAME:

Matrikelnummer:

- (d) Sei B ein 4-bit Register. Berechnen Sie $B_{2C} = X_{2C} - Y_{2C}$ im Zweierkomplement mittels der Subtraktion nach der Schulmethode. Kommt es zu einem Overflow? Wenn ja, welche Folge hat das für den Wert im Register B ?
- (e) Um einen Overflow zu verhindern, sollen die Register auf 8 bit erweitert werden. Stellen Sie dazu die Zahlen X und Y im 8-bit Zweierkomplement dar.
- (f) Welcher Dezimalzahl entspricht folgende Binärzahl im IEEE 754, single precision, Gleitkommaformat: 1 10000000 00100000000000000000000? Der Exponent ist im Exzess-127 Code dargestellt. Die Mantisse ist auf 1 normiert und besitzt daher ein hidden bit.

Aufgabe 5 (Mikroprogrammierung)**[25 Punkte]**

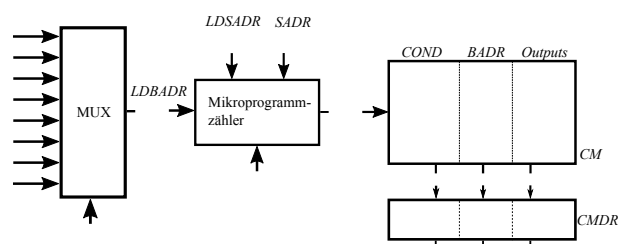
Für einen Kaffeeautomaten soll ein Mikroprogramm-Steuerwerk entworfen werden. Der Automat soll Kaffee in den Stärken *Normal* und *Stark* anbieten können. Zur Bedienung verfügt der Automat über ein Display, einen Bedienknopf (engl. button) und einen Kippschalter (engl. switch), um die Stärke des Kaffees auszuwählen. In Abbildung 8(a) ist eine Skizze des Bedienmoduls des Kaffeeautomaten zu sehen. Der Bestellablauf ist wie folgt:

Zu Beginn befindet sich der Automat im Ruhemodus. Nachdem der Kunde den Bedienknopf betätigt hat, kann er die Stärke des Kaffees auswählen. Hierzu wird auf dem Display „Stärke?“ angezeigt. Um die Stärke auszuwählen, stellt der Kunde den Kippschalter auf die gewünschte Stärke ein und bestätigt die Auswahl durch Drücken des Bedienknopfs. Je nach ausgewählter Stärke wird eine unterschiedliche Menge Kaffeebohnen gemahlen (engl. to grind) und so die Stärke des Kaffees bestimmt. Anschließend wird der Kaffee unabhängig von der Stärke des Kaffees gekocht (engl. to brew). Während der Kaffee gemahlen und gekocht wird, wird auf dem Display „Kaffee kommt...“ angezeigt. Nachdem der Kaffee gekocht wurde, wird auf dem Display „Kaffee entnehmen!“ angezeigt. Sobald der Kaffee entnommen wurde, geht der Automat zurück in den Ruhezustand. Wenn das Mahlen oder Kochen des Kaffees abgeschlossen ist, wird dies durch ein DONE-Signal angezeigt. Um festzustellen ob der Kaffee entnommen wurde, wird ein Sensor verwendet, der ein ENTNOMMEN-Signal liefert, sobald der Kaffee entnommen wurde.

- (a) Spezifizieren Sie geeignete Eingangssignale und weisen diesen eine Adresse am Multiplexer zu. Achten Sie darauf, dass die Eingangssignale mit einem Mikroprogramm-Steuerwerk (siehe Abbildung 8(b)) verwendet werden sollen. Tragen Sie die Zuordnung in die Tabelle 1 ein.



(a) Skizze des Bedienmoduls des Kaffeeautomaten



(b) Übersicht über ein Mikroprogramm-Steuerwerk

Abbildung 8: Skizze eines Kaffeeautomaten und Übersicht eines Mikroprogramm-Steuerwerks

| Adresse | Bezeichnung |
|---------|-------------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Tabelle 1: Tabelle für **(a)**, Adressbelegung des Multiplexers

| Adresse | Bezeichnung |
|---------|-------------|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Tabelle 2: Ersatztablette für **(a)**, **ungültige Lösungen streichen!**

NAME:

Matrikelnummer:

- (b) Entwerfen Sie einen Moore-Automaten, der den oben genannten Bestellablauf abbildet. Verwenden Sie hierzu folgende Ausgangssignale:

| | | | |
|----|---------------------------|----|---|
| MN | Mahle normal viele Bohnen | DS | Zeige im Display „Stärke?“ an |
| MS | Mahle viele Bohnen | DK | Zeige im Display „Kaffee kommt...“ an |
| K | Koche den Kaffee | DE | Zeige im Display „Kaffee entnehmen!“ an |

Hinweis: Fügen Sie ggf. Dummy-Zustände ein.



Ersatzautomat, ungültige Lösung streichen!



NAME:

Matrikelnummer:

- (c) Der in (b) entworfene Moore-Automat soll mit Hilfe eines Mikroprogramm-Steuerwerks realisiert werden. Geben Sie eine geeignete Codierung für die verwendeten Zustände an. Füllen Sie hierzu die Tabelle 3 aus.

| Codierung | Bezeichnung | Beschreibung |
|-----------|-------------|--------------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Tabelle 3: Tabelle für (c), Codierung der Zustände

| Codierung | Bezeichnung | Beschreibung |
|-----------|-------------|--------------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Tabelle 4: Ersatztabelle für (c), **ungültige Lösungen streichen!**

- (d) Vervollständigen Sie das in Abbildung 9 gezeigte Mikroprogramm-Steuerwerk, indem Sie die benötigten Leitungen einzeichnen und die Bitbreiten angeben.

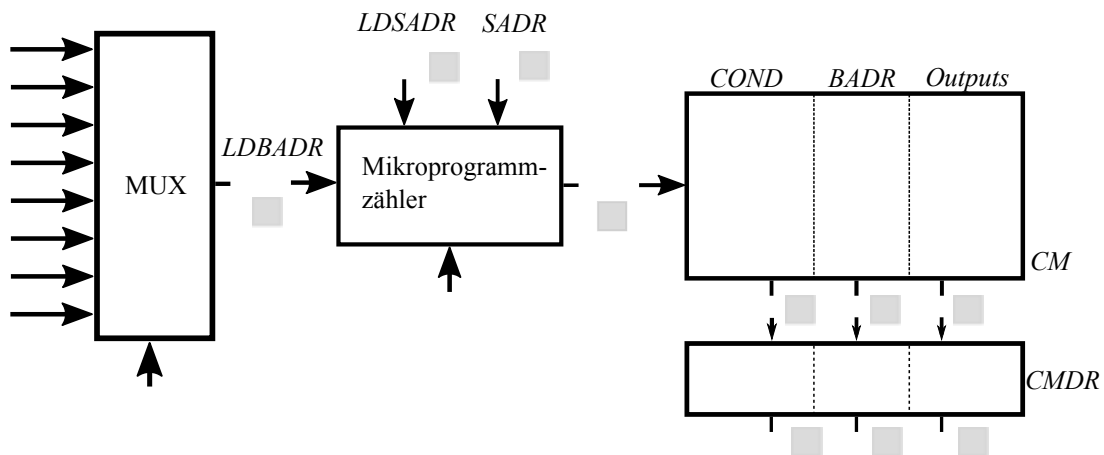


Abbildung 9: Vorlage für (d), Mikroprogramm-Steuerwerk

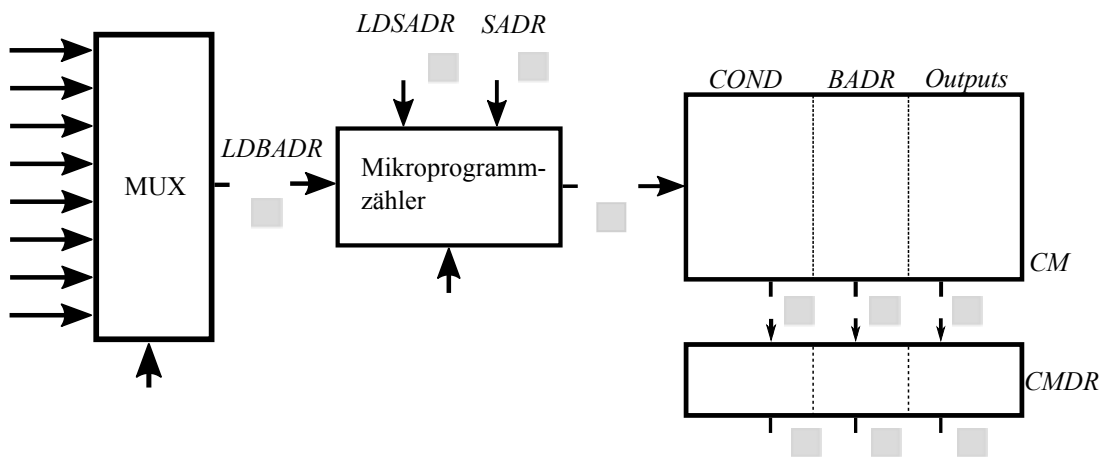


Abbildung 10: Ersatzvorlage für (d), ungültige Lösungen streichen!

NAME:

Matrikelnummer:

(e) Tragen Sie den Inhalt des Control Memory (CM) in die Tabelle 5 ein.

| Adresse | COND | BADR | K | MN | MS | DS | DK | DE |
|---------|------|------|---|----|----|----|----|----|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Tabelle 5: Tabelle für (e), Control Memory

| Adresse | COND | BADR | K | MN | MS | DS | DK | DE |
|---------|------|------|---|----|----|----|----|----|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Tabelle 6: Ersatztabelle für (e), ungültige Lösungen streichen!

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!