

[Schreibtisch](#) / [Kurse](#) / [SS21](#) / [L.079.05200 Digitaltechnik](#) / Online-Klausur 2. Termin (14.03.2022, 10:00 Uhr)
 / [DT Online-Klausur \(14.03.2022\)](#) / [Vorschau](#)

Begonnen am Dienstag, 19. April 2022, 16:39

Status Beendet

Beendet am Dienstag, 19. April 2022, 16:40

Verbrauchte Zeit 14 Sekunden

Bewertung 0,00 von 90,00 (0%)

Frage **1**

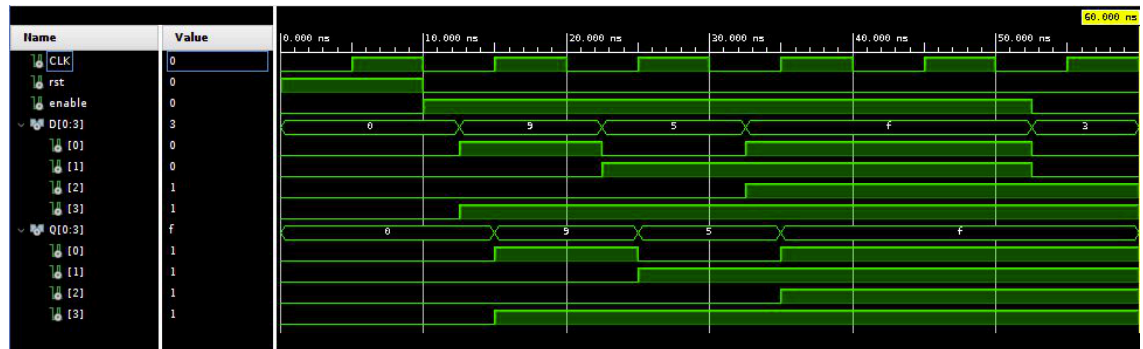
Nicht
beantwortet

Erreichbare
Punkte: 4,00

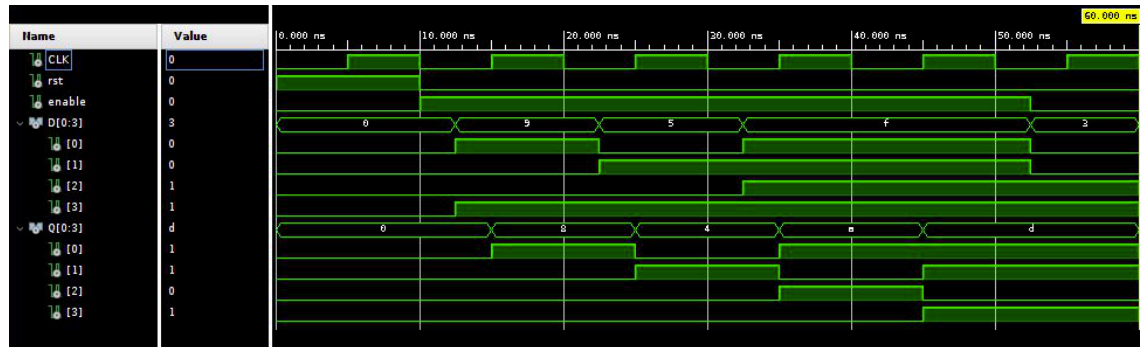
Gegeben seien die folgenden Zeitverhalten. ([Download](#))

Ordnen Sie den unten stehenden Schaltungen das zugehörige Zeitverhalten zu.

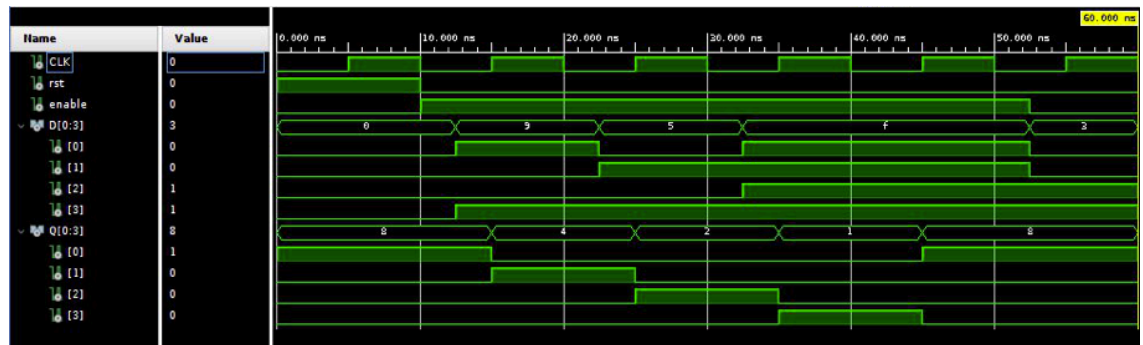
Zeitverhalten 1:

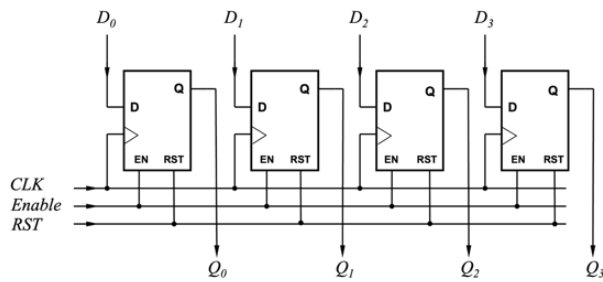


Zeitverhalten 2:

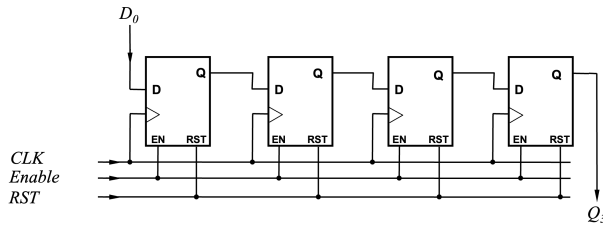


Zeitverhalten 3:

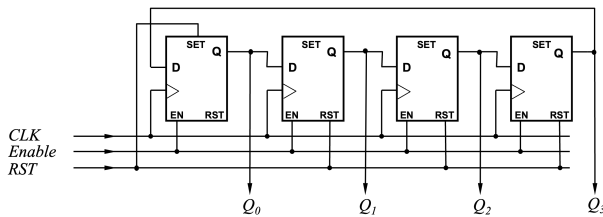




Auswählen ...



Auswählen ...



Auswählen ...

Frage 2

Nicht
beantwortetErreichbare
Punkte: 2,00Gegeben sei die Binärzahl $0011\ 1000\ 1100\ 1110_2$.

Geben Sie die Zahl in Hexadezimaldarstellung an:

Frage 3

Nicht
beantwortetErreichbare
Punkte: 2,00

Welche Dezimalzahl wird von der folgenden 32-Bit Binärzahl im IEEE 754 Gleitkommaformat dargestellt.

0 10000011 011100000000000000000000 =

Frage 4

Nicht
beantwortetErreichbare
Punkte: 2,00Stellen Sie die Dezimalzahl -117_{10} als 8-Bit Binärzahl im 2er-Komplement dar.

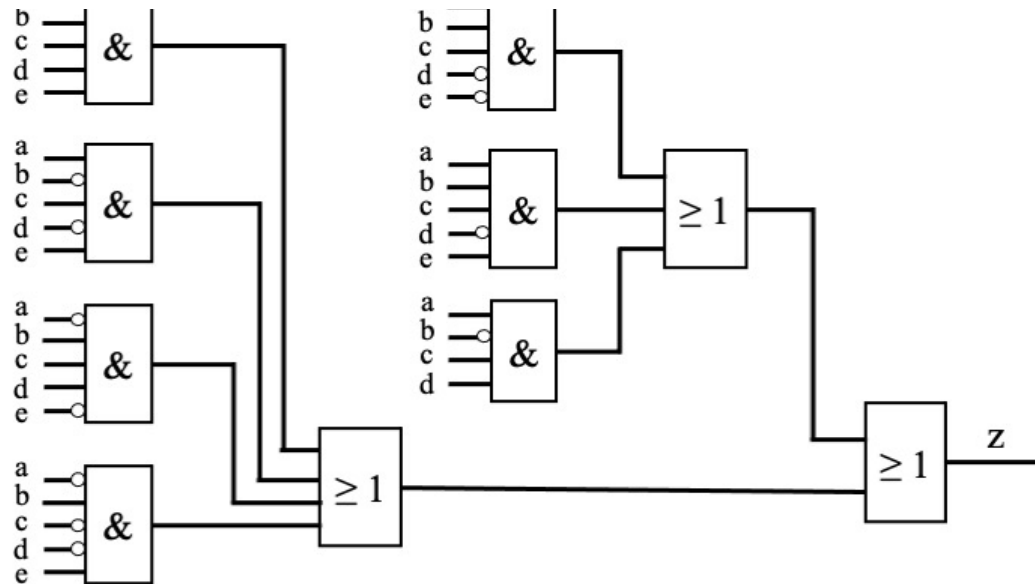
Wählen Sie eine Antwort:

- ☐ 1000 1010
- ☐ Keine Antwort ist richtig.
- ☐ 1000 1011
- ☐ 0111 0101

Frage 5

Nicht
beantwortetBetrachten Sie folgende Logikschaltung der Funktion $z=f(a,b,c,d,e)$.

Erreichbare
Punkte: 3,00



Geben Sie die Einstellenmenge \mathcal{E} der Schaltungsfunktion in der Codewort-Notation an, bei der positive Literale mit einer 1 codiert werden und negative Literale mit einer 0.

Beispiel für die Codewort-Notation: Die Codewort-Darstellung des Implikanten $a'bcd'e$ lautet 01101.

Frage **6**

Nicht
beantwortet

Erreichbare
Punkte: 10,00

Gegeben sei die folgende Einstellenmenge \mathcal{E} in der Codewort-Notation.

$\mathcal{E} = \{00000, 01100, 11000, 01111, 11100, 10000, 01110, 11001, 11110, 10011\}$

Berechnen Sie die Primimplikanten von \mathcal{E} mit Hilfe der ersten Phase des Quine-McCluskey Verfahrens.

Verwenden Sie die untenstehende Vorlage und die Codewort-Notation. Gruppieren Sie dabei die Implikanten nach der Anzahl der Einsen in den Codewörtern.

Markieren Sie jeweils die Implikanten, die zur Erzeugung von allgemeineren Implikanten mit weniger Literalen genutzt wurden (Spalte (✓)).

Panda-Hinweis: Falls Sie nicht alle Textfelder in der Tabelle benötigen, wird Panda die Aufgabe trotzdem als unvollständig markieren. Dies ist der Modellierung von Lückentexten in Panda geschuldet und kann von Ihnen ignoriert werden. Alle Aufgaben werden manuell nachkorrigiert!

L1 (Einstellen)			L2			L3			L4		
Gruppe	Codewort	✓	Gruppe	Codewort	✓	Gruppe	Codewort	✓	Gruppe	Codewort	✓

Grp. 0			Grp. 0			Grp. 0			Grp. 0		
Grp. 1			Grp. 1			Grp. 1			Grp. 1		
Grp. 2			Grp. 2			Grp. 2			Grp. 2		
Grp. 3			Grp. 3			Grp. 3			Grp. 3		
Grp. 4			Grp. 4			Grp. 4			Grp. 4		
Grp. 5			Grp. 5			Grp. 5			Grp. 5		

Frage **7**Nicht
beantwortetErreichbare
Punkte: 2,00Geben Sie alle Primimplikanten von ε an.

Frage 8

Nicht
beantwortetErreichbare
Punkte: 1,00

Eine Logikfunktion, welche ausschließlich mit AND und OR Gattern spezifiziert ist, soll in eine äquivalente Logikfunktion umgeformt werden, welche ausschließlich NAND Gatter benötigt. Welche Umformungsregel kann dazu angewandt werden?

Frage 9

Nicht
beantwortetErreichbare
Punkte: 4,00

Gegeben ist die untenstehende Logikfunktion z.

$$z = ade + bcd + cde + acd$$

Welche der folgenden Ausdrücke ist **äquivalent zu z** und verwendet **ausschließlich 3-NAND und 2-NAND Gatter**?

Hinweis: Eine einzelne Negation kann durch ein 2-NAND Gatter implementiert werden.

- ☐ a.
$$z = \overline{\overline{ad} \cdot \overline{bd} \cdot \overline{cd} \cdot \overline{ad} \cdot \overline{ab} \cdot \overline{de}}$$
- ☐ b.
$$z = \overline{\overline{ade} \cdot \overline{bcd} \cdot \overline{cde} \cdot \overline{acd}}$$
- ☐ c.
$$z = \overline{\overline{ade} \cdot \overline{bcd} \cdot \overline{cde} \cdot \overline{acd}}$$
- ☐ d.
$$z = \overline{\overline{ade} \cdot \overline{bcd} \cdot \overline{cde} \cdot \overline{acd}}$$

Information

Sind die folgenden Funktionen mit einem Mealy- bzw. Moore-Automaten realisierbar?

Wählen Sie jeweils ja/nein aus (1 P bei Korrektheit) **und** geben Sie eine kurze Begründung Ihrer Antwort (1 P bei Korrektheit).

Frage 10

Nicht
beantwortet

Erreichbare
Punkte: 2,00

Feststellen, wie viele Münzen in einen Kaffeeautomaten eingeworfen wurden?

Sie können hier annehmen, dass prinzipiell eine unendliche Anzahl an Münzen vorhanden ist.

Realisierbar?

Begründung

Frage 11

Nicht
beantwortet

Erreichbare
Punkte: 2,00

Sequentielles Addieren zweier 32-Bit Zahlen in Binärdarstellung?

Realisierbar?

Begründung

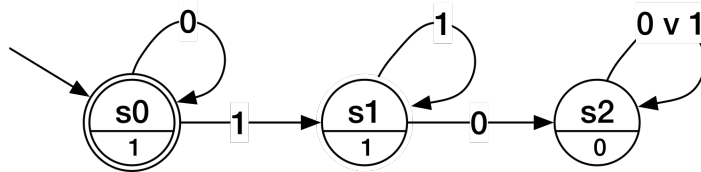
Frage 12

Nicht
beantwortet

Erreichbare
Punkte: 2,00

Gegeben sei der folgende Moore-Automat. Als Eingabe dient ein beliebiger Bitstream, der vom niederwertigsten Bit aus abgearbeitet wird.

Welche Eigenschaft hat der Bitstream, wenn nach der bitweisen Bearbeitung des gesamten Bitstreams die Ausgabe des Moore-Automaten 1 ist?



- ☐ a. Im Bitstream kommen alle 0en vor allen 1en. (z.B.: 1111 1000)
- ☐ b. Der Bitstream beginnt mit 10. (z.B.: 01 0100 0001)
- ☐ c. Der Bitstream beginnt mit 01. (z.B.: 1100 1111 0010)
- ☐ d. Im Bitstream kommen alle 1en vor allen 0en. (z.B.: 0011 1111 1111)
- ☐ e. Keine Antwort ist richtig.

Information

Beim Aufstellen eines Automatengraphen oder einer Automatentafel werden häufig mehr Zustände als notwendig verwendet. Dadurch wird die Hardware-Implementierung unnötig kompliziert und die Überprüfung der Implementierung auf korrektes Funktionieren kann schwierig werden. Eine wichtige Vereinfachung bietet die **Zustandsreduktion**.

Frage 13

Nicht
beantwortetErreichbare
Punkte: 14,00

Führen Sie für den folgenden Mealy-Automaten die Reduktion nach dem Verfahren von Ginsburg/Huffmann durch.

Hier, sowie im PANDA-Kurs, können Sie eine Vorlage zur Durchführung des Verfahrens herunterladen.

Hinweis: Nicht genutzte Tabellen können freigelassen werden.

δ / λ	a	b
S1	S3 / 1	S6 / 0
S2	S3 / 1	S7 / 0
S3	S1 / 0	S2 / 0
S4	S1 / 0	S8 / 0
S5	S6 / 0	S7 / 0
S6	S7 / 1	S1 / 1
S7	S6 / 1	S1 / 1
S8	S2 / 1	S6 / 1

1-Äquivalenz

1. Schritt: Bestimmen Sie zunächst die 1-Äquivalenzklassen

1-Äquivalenzklasse	δ / λ	a	b
	S1	S3 / 1	S6 / 0
	S2	S3 / 1	S7 / 0
	S3	S1 / 0	S2 / 0
	S4	S1 / 0	S8 / 0
	S5	S6 / 0	S7 / 0
	S6	S7 / 1	S1 / 1
	S7	S6 / 1	S1 / 1
	S8	S2 / 1	S6 / 1

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 1-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind.

1-Äquivalenzklasse	Zustand	δ		Übergang eindeutig?
		a	b	
(wie angegeben)	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

3. Schritt: Falls die Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

2-Äquivalenz

1. Schritt: Bestimmen Sie die 2-Äquivalenzklassen

2-Äquivalenzklasse	δ / λ	a	b
<input type="text"/>	S1	(wie angegeben)	
<input type="text"/>	S2		
<input type="text"/>	S3		
<input type="text"/>	S4		
<input type="text"/>	S5		
<input type="text"/>	S6		
<input type="text"/>	S7		
<input type="text"/>	S8		

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 2-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

2-Äquivalenzklasse	Zustand	δ		Übergang eindeutig?
		a	b	
(wie angegeben)	S1	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S2	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S3	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S4	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S5	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S6	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S7	<input type="text"/>	<input type="text"/>	<input type="text"/>
	S8	<input type="text"/>	<input type="text"/>	<input type="text"/>

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

3-Äquivalenz

1. Schritt: Bestimmen Sie die 3-Äquivalenzklassen

3-Äquivalenzklasse	δ / λ	a	b
<input type="text"/>	S1	(wie angegeben)	
<input type="text"/>	S2		
<input type="text"/>	S3		
<input type="text"/>	S4		

		S5	
		S6	
		S7	
		S8	

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 3-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

3-Äquivalenzklasse (wie angegeben)	Zustand	δ		Übergang eindeutig?
		a	b	
	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

4-Äquivalenz

1. Schritt: Bestimmen Sie die 4-Äquivalenzklassen

4-Äquivalenzklasse	δ / λ	a	b
	S1	(wie angegeben)	
	S2		
	S3		
	S4		
	S5		
	S6		
	S7		
	S8		

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 4-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

4-Äquivalenzklasse	Zustand	δ		Übergang
		a	b	

Äquivalenzklasse	Zustand	a	b	eindeutig?
(wie angegeben)	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

5-Äquivalenz

1. Schritt: Bestimmen Sie die 5-Äquivalenzklassen

5-Äquivalenzklasse	δ / λ	a	b
	S1	(wie angegeben)	
	S2		
	S3		
	S4		
	S5		
	S6		
	S7		
	S8		

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 5-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

5-Äquivalenzklasse	Zustand	δ		Übergang eindeutig?
		a	b	
(wie angegeben)	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

30

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

6-Äquivalenz

1. Schritt: Bestimmen Sie die 6-Äquivalenzklassen

6-Äquivalenzklasse	δ / λ	a	b
	S1	(wie angegeben)	
	S2		
	S3		
	S4		
	S5		
	S6		
	S7		
	S8		

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 6-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

6-Äquivalenzklasse	Zustand	δ		Übergang eindeutig?
		a	b	
(wie angegeben)	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

7-Äquivalenz

1. Schritt: Bestimmen Sie die 7-Äquivalenzklassen

7-Äquivalenzklasse	δ / λ	a	b
	S1	(wie angegeben)	
	S2		

		S3	
		S4	
		S5	
		S6	
		S7	
		S8	

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 7-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

7-Äquivalenzklasse (wie angegeben)	Zustand	δ		Übergang eindeutig?
		a	b	
	1			
	2			
	3			
	4			
	5			
	6			
	7			
	8			

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

8-Äquivalenz

1. Schritt: Bestimmen Sie die 8-Äquivalenzklassen

8-Äquivalenzklasse	δ / λ	a	b
	S1	(wie angegeben)	
	S2		
	S3		
	S4		
	S5		
	S6		
	S7		
	S8		

2. Schritt: Bestimmen Sie nun die Übergangsfunktion in die 8-Äquivalenzklassen und geben Sie an, ob die entstandenen Übergänge innerhalb einer Äquivalenzklasse eindeutig sind. Falls ein Übergang nicht eindeutig ist, wählen Sie in der entsprechenden Zeile "nein" aus.

8- Äquivalenzklasse (wie angegeben)	Zustand	δ		Übergang eindeutig?
		a	b	
	S1			
	S2			
	S3			
	S4			
	S5			
	S6			
	S7			
	S8			

3. Schritt: Falls alle Übergänge eindeutig sind, endet das Verfahren. Falls nicht, muss die nächste Äquivalenzklasse gebildet werden. Führen Sie das Verfahren so lange wie notwendig fort.

Information

Der in der folgenden Abbildung 4.1 gezeigte Automatengraph soll mithilfe eines Mikroprogramm-Steuerwerks (siehe Abbildung 4.2) realisiert werden.

Hier können Sie die Abbildungen zusätzlich herunterladen:

- [Abbildung 4.1](#)
- [Abbildung 4.2](#)

Abbildung 4.1

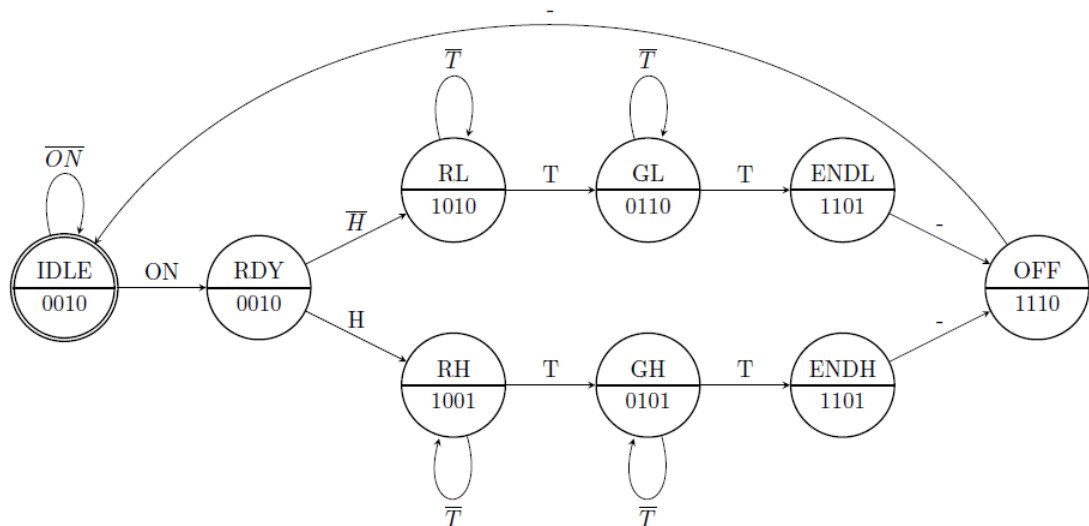
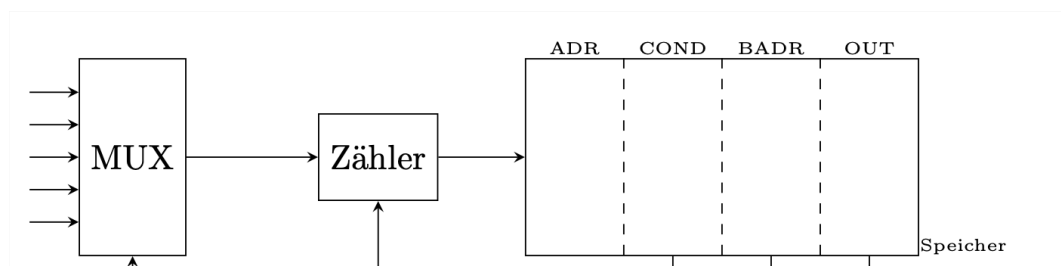
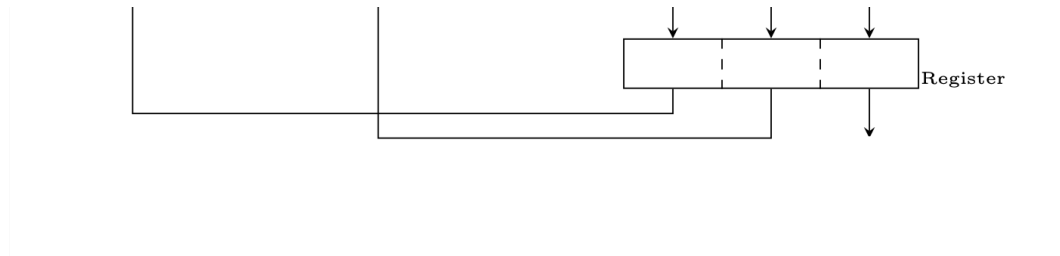


Abbildung 4.2





Die Abkürzung "ADR" steht hier für Adresse, "COND" heißt Condition bzw. Bedingung, "BADR" steht für branch address bzw. Sprungadresse und "OUT" steht für Output.

Frage 14

Nicht
beantwortet

Erreichbare
Punkte: 3,00

Geben Sie eine geeignete Kodierung für die verwendeten Zustände des Automatengraphen an.

Tragen Sie hierzu die Codeworte in Binärdarstellung zu den entsprechenden Zustandsbezeichnungen in die folgende Tabelle ein.

Hinweis: Notieren Sie sich Ihre Lösung auf einem Blatt Papier (o.ä.), um sie ggf. später weiter verwenden zu können.

Zustandsbezeichnung	Kodierung
IDLE	<input type="text"/>
READY	<input type="text"/>
RL	<input type="text"/>
GL	<input type="text"/>
ENDL	<input type="text"/>
RH	<input type="text"/>
GH	<input type="text"/>
ENDH	<input type="text"/>
OFF	<input type="text"/>

Frage 15

Nicht
beantwortet

Erreichbare
Punkte: 3,00

Für die Implementierung des gezeigten Automaten als Mikroprogramm-Steuerwerk werden geeignete Sprungbedingungen benötigt.

Spezifizieren Sie hierzu geeignete Eingangssignale für den gezeigten Input-Multiplexer und weisen Sie den Eingangssignalen geeignete Adressen zu.

Tragen Sie die Signalbezeichnungen und die zugehörigen Codeworte in die folgende Tabelle ein.

Hinweis: Notieren Sie sich Ihre Lösung auf einem Blatt Papier (o.ä.), um sie ggf. später weiter verwenden zu können.

Panda-Hinweis: Falls Sie nicht alle Textfelder in der Tabelle benötigen, wird Panda die Aufgabe trotzdem als unvollständig markieren. Dies ist der Modellierung von Lückentexten in Panda geschuldet und kann von Ihnen ignoriert werden. Alle Aufgaben werden manuell nachkorrigiert!

Sprungbedingung	Invertiert?	Kodierung
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Information

Zur besseren Übersicht finden Sie hier noch einmal die Abbildungen.

Abbildung 4.1

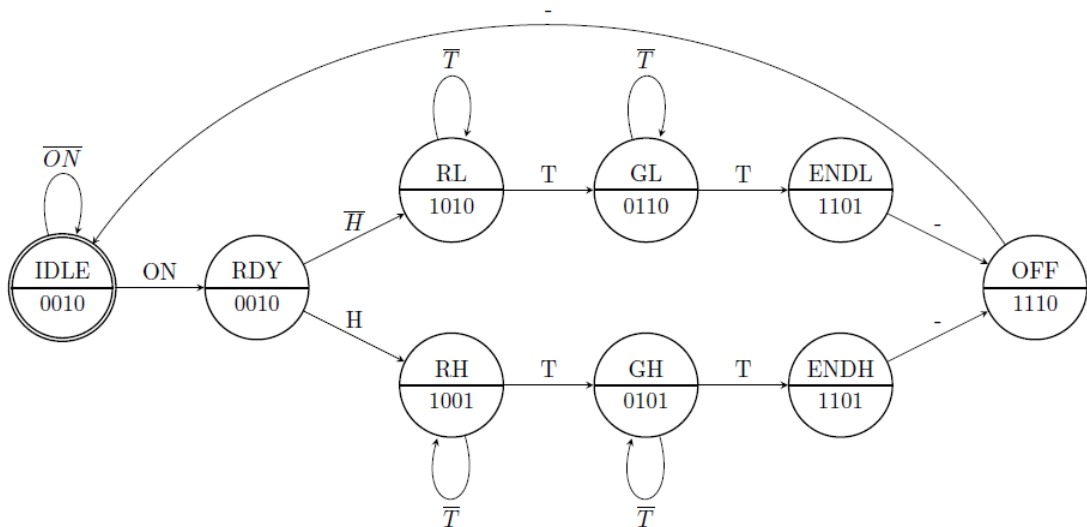
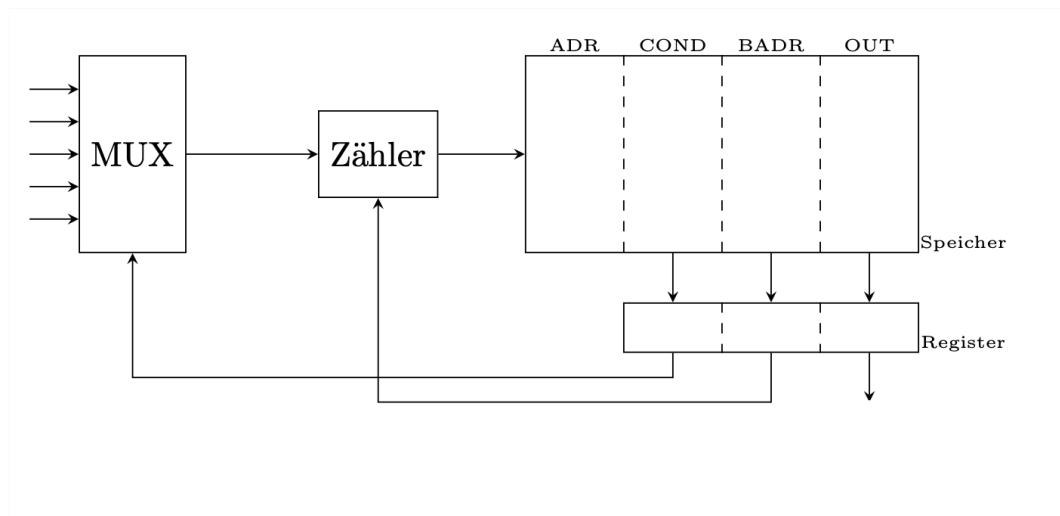


Abbildung 4.2



Frage 16

Nicht
beantwortetErreichbare
Punkte: 10,00

Vervollständigen Sie das gezeigte Mikroprogramm-Steuerwerk, indem Sie den Inhalt des Speichers in die untenstehende Tabelle eintragen.

Die Abkürzung "ADR" steht hier für Adresse, "COND" heißt Condition bzw. Bedingung, "BADR" steht für branch address bzw. Sprungadresse und "OUT" steht für Output.

Hinweis: Falls Sie ein don't-care-Symbol benötigen, benutzen sie "----" (ohne die "").

Hinweis: Notieren Sie sich Ihre Lösung auf einem Blatt Papier (o.ä.), um sie ggf. später weiter verwenden zu können.

ADR	COND	BADR	OUT

Information

Zur besseren Übersicht finden Sie hier noch einmal die Abbildungen.

Abbildung 4.1

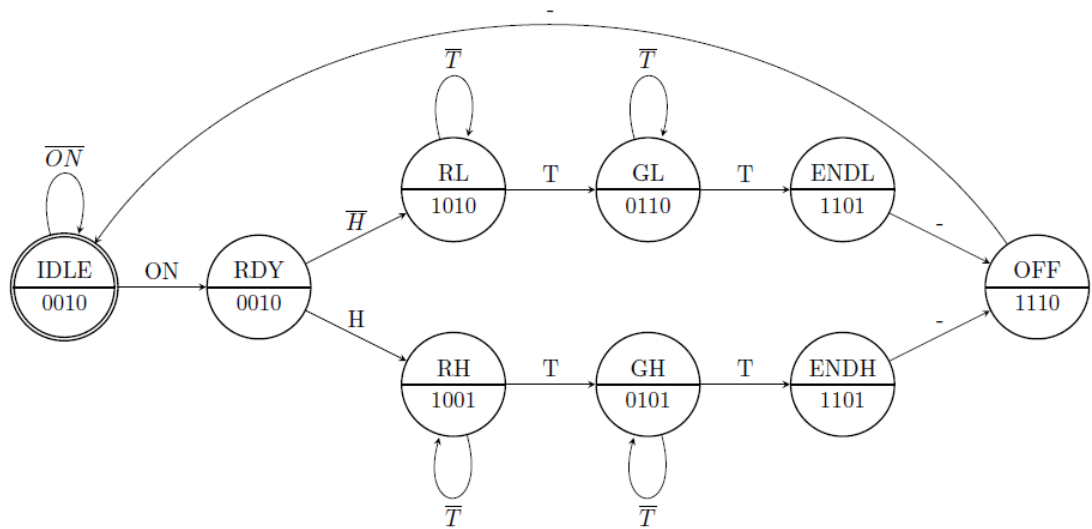
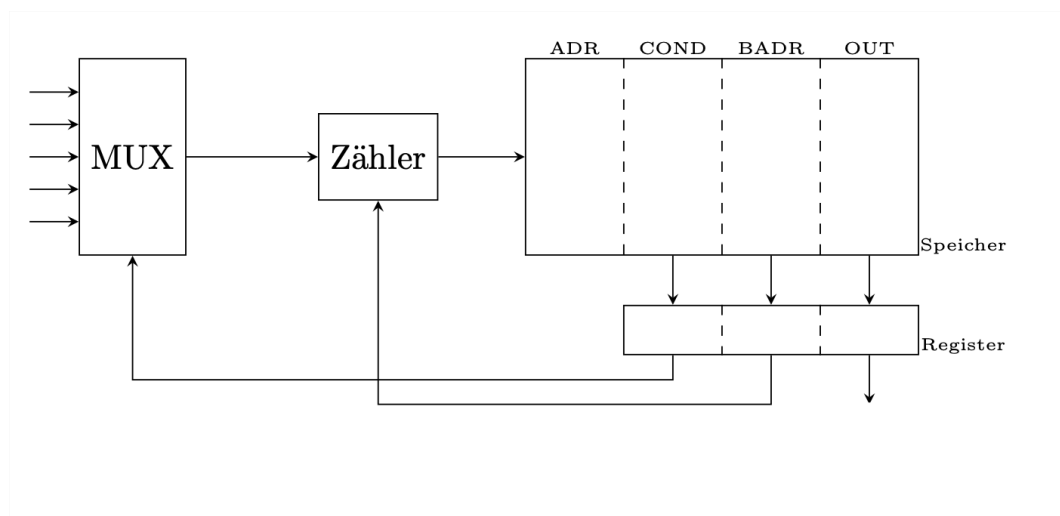


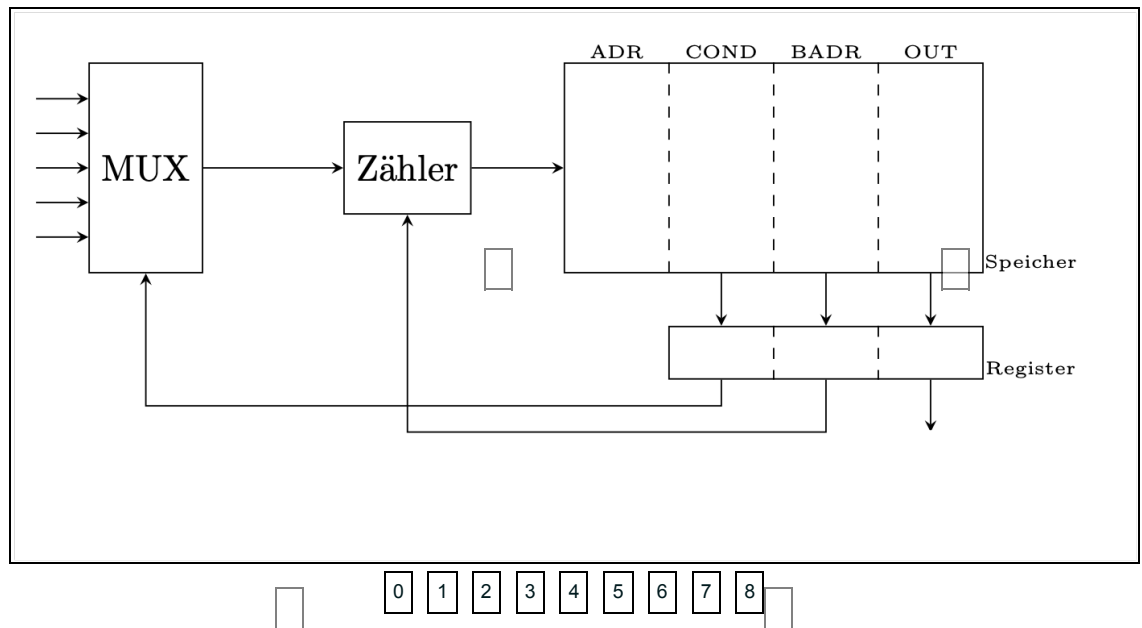
Abbildung 4.2



Frage 17

Nicht
beantwortetErreichbare
Punkte: 2,00

Vervollständigen Sie das in der folgenden Abbildung gezeigte Mikroprogramm-Steuerwerk, indem Sie die benötigten Busbreiten neben den jeweiligen Signalen platzieren.



Information

Zur besseren Übersicht finden Sie hier noch einmal die Abbildungen.

Abbildung 4.1

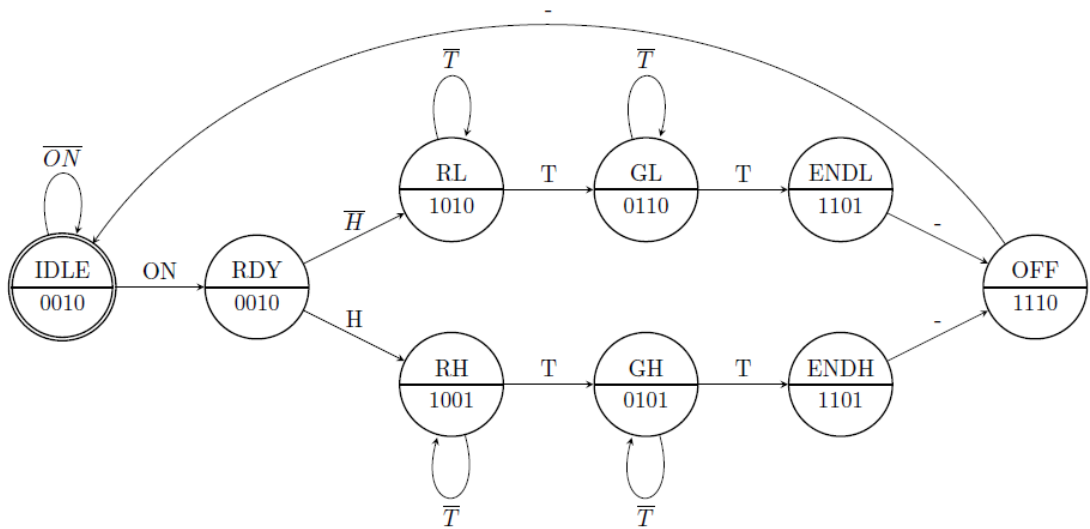
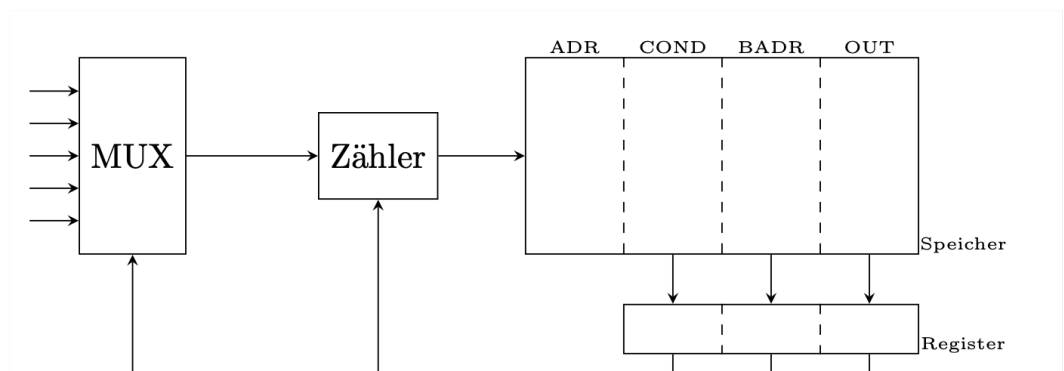


Abbildung 4.2



Frage 18

Nicht
beantwortetErreichbare
Punkte: 2,00

Um Speicherplatz zu sparen, sollen die Outputs im Speicher anders abgelegt werden. Füllen Sie folgenden Lückentext, der beschreibt, wie Platz gespart werden kann und wie der Dekodierer funktioniert.

Anstatt die vollen Bits im Speicher zu speichern, reichen Bits aus, um die Outputs zu kodieren.

Wenn ein Output im Speicher gespeichert werden soll, werden Bits in einem Bit zusammengefasst.

Der Dekodierer gibt dann Bits einfach weiter. Um das zusammengefasste Bit zu dekodieren, kann der Dekodierer ein einzelnes -Gate benutzen, um Bits wieder herzustellen.

Information

Zur besseren Übersicht finden Sie hier noch einmal die Abbildungen.

Abbildung 4.1

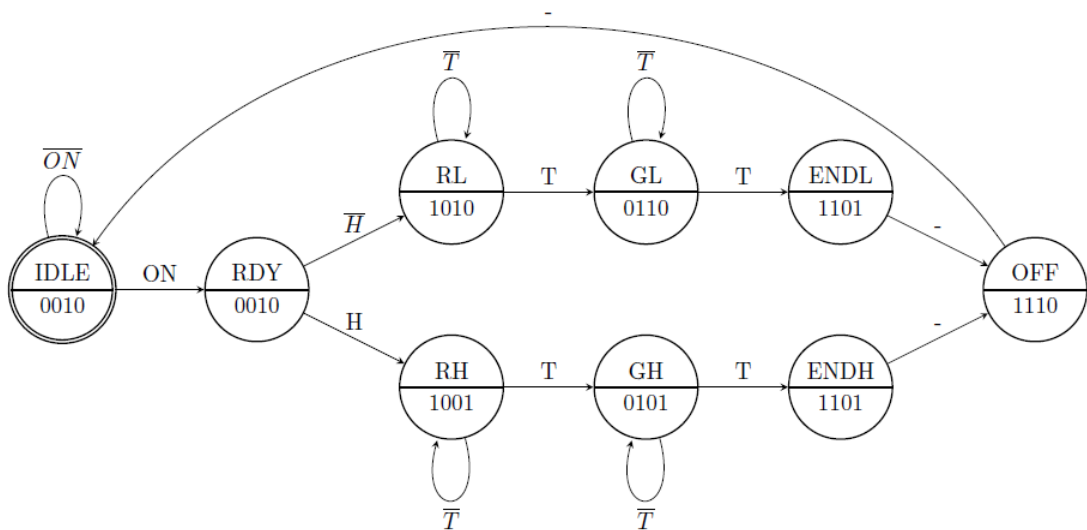
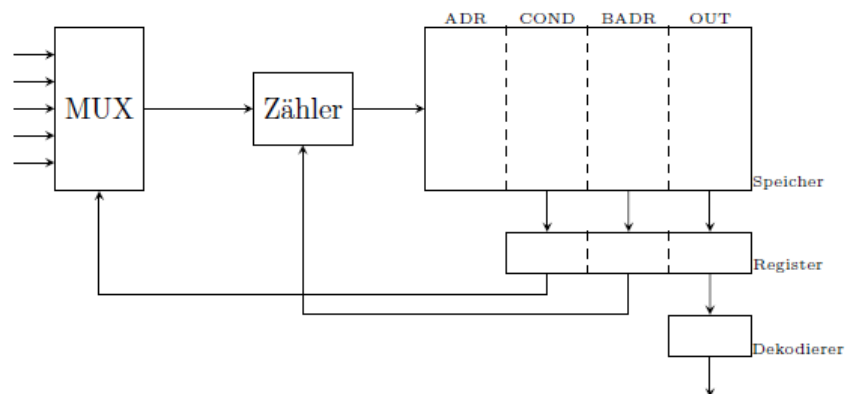


Abbildung 4.3



Frage 19

Nicht
beantwortet
Erreichbare
Punkte: 1,00

Benennen Sie die Bereiche einer VHDL Datei!

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity <modulname> is
port (
  clk          : in  std_logic;
  enable       : in  std_logic;
  result       : out std_logic_vector (6 downto 0));
end <modulname>;
```

```
architecture behave of <modulname> is
```

```
signal r_lfsr          : std_logic_vector (7 downto 1);
```

```
begin
```

```
result <= r_lfsr(7 downto 1);
my_proc: process(clk, enable) is
begin
end prcoess;
```

```
end <modulname>;
```

Ziehen Sie dafür nachfolgende Begriffe auf die freien Felder.

Deklarationsteil

Anweisungsteil

Schnittstellenbeschreibung

Frage 20

Nicht
beantwortet
Erreichbare
Punkte: 1,00

Wie lautet in VHDL der Zuweisungsoperator, mit dem ein Wert einem Signal zugewiesen wird?

Beispiel:

```
signal data : std_logic;
```

...

data [?] wert;

Wählen Sie eine Antwort:

- ☐ a. +=
- ☐ b. :=
- ☐ c. <=
- ☐ d. ==>
- ☐ e. =

Frage 21

Nicht
beantwortet
Erreichbare
Punkte: 1,00

Was versteht man in VHDL bei einem Prozess unter einer Sensitivitätsliste?

Wählen Sie eine Antwort:

- ☐ a. Sie beinhaltet die Ausgaben der Signale der Testbench Datei des zu testenden Moduls.
- ☐ b. Sie beinhaltet die Reihenfolge der Zahlen bei einem Zähler.
- ☐ c. Es ist die Liste von Signalen, die einen Prozess aktivieren.
- ☐ d. Dem Prozess können durch die Sensitivitätsliste Parameter übergeben werden.

u. Dem Prozess können durch die Sensitivitätsliste Parameter übergeben werden.

Frage 22

Nicht
beantwortet

Erreichbare
Punkte: 1,00

Der VHDL Datentyp **std_logic** kann zusätzlich zu den Werten '1' und '0' noch welche folgenden Werte annehmen?

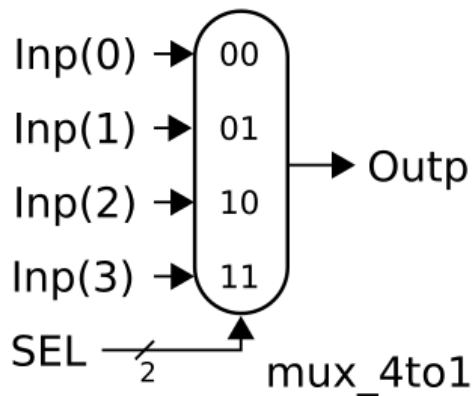
- ☐ 'Z' : hochohmig
- ☐ 'W' : unbekannt (schwacher Treiber)
- ☐ 'U' : undefiniert - wird für nichtinitialisierte Signale in der Simulation verwendet
- ☐ '-' : unwichtig (don't care)
- ☐ 'H' : logische Eins (schwacher Treiber)
- ☐ 'X' : unbekannt (starker Treiber)
- ☐ 'L' : logische Null (schwacher Treiber)

Frage 23

Nicht
beantwortet

Erreichbare
Punkte: 8,00

Nachfolgende Abbildung zeigt den Multiplexer **mux_4to1** mit den Eingängen *Inp* und *SEL* und dem Ausgang *Outp*. ([Download](#))



Vervollständigen Sie den nachfolgenden VHDL Code, so dass die Funktionalität eines Multiplexers für das Modul *mux_4to1* gegeben ist.

Sollte der Eingang *SEL* nicht eine der Belegungen 00, 01, 10 oder 11 haben, so muß der Ausgang *Outp* ein undefinierten Pegel ausgeben!

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mux_4to1 is
  Port ( [ ] : [ ] [ ] ( [ ] [ ] 0 );
         [ ] : out std_logic;
         [ ] : [ ] [ ] (1 [ ] 0) [ ]
end mux_4to1;

architecture Behavioral of [ ] is
begin
  mux_proc: process(SEL, [ ]) is
  begin
    if(SEL = [ ]) then
      Outp <= Inp([ ]);
    end if;
  end process;
end mux_4to1;
  
```

```

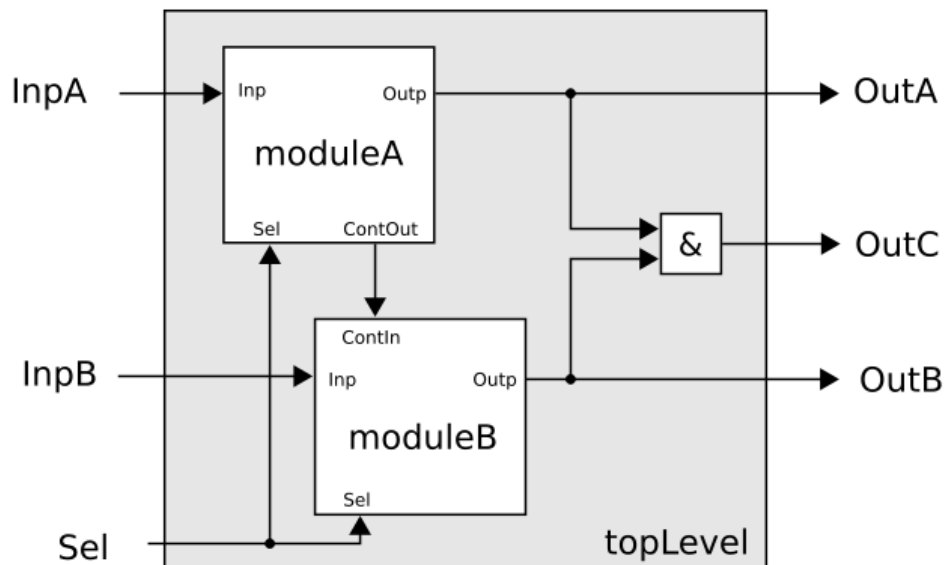
        (SEL = ) then
            Outp <= Inp( );
        (SEL = ) then
            Outp <= Inp( );
        (SEL = ) then
            Outp <= Inp( );
        Outp <= ;
    end process;
end ;

```

Frage 24

Nicht
beantwortetErreichbare
Punkte: 8,00

Gegeben sei die nachfolgende Strukturbeschreibung des Moduls **topLevel**, welches wiederum aus einer Verschaltung der beiden Module **moduleA** und **moduleB** besteht. ([Download](#))



Ergänzen Sie nachfolgenden VHDL Code, so dass dieser die Funktionalität der oben gezeigten Schaltung abbildet.

```

entity topLevel is
    port (
        InpA: in std_logic;
        InpB: in std_logic;
        Sel : in std_logic;
        OutA : out std_logic;
        OutB : out std_logic;
        OutC : out std_logic
    );
end topLevel;

architecture Structural of topLevel is
    -- Deklaration moduleA
    moduleA

```

```

    (
        Inp      : in  std_logic;
        Sel      : in  std_logic;
        ContOut  : out std_logic;
        Outp     : out std_logic
    );
end moduleA;

-- Deklaration moduleB
moduleB
(
    Inp      : in  std_logic;
    Sel      : in  std_logic;
    ContIn   : in  std_logic;
    Outp     : out std_logic
);
end moduleB;

-- zusätzliche Signaldeklaration
signal sig_cont : std_logic;
signal sig_outpA : std_logic;
signal sig_outpB : std_logic;

begin

inst_A: moduleA --Instanz moduleA
(
    Inp      <input> ,
    Sel      <input> ,
    ContOut  <input> ,
    Outp     <input>
);

inst_B: moduleB --Instanz moduleB
(
    Inp      <input> ,
    Sel      <input> ,
    ContIn   <input> ,
    Outp     <input>
);

-- Signalzuweisungen
<input> <input> sig_outpA;
<input> <input> <input>;
<input> <input> sig_outpA and <input>;

end Structural;

```

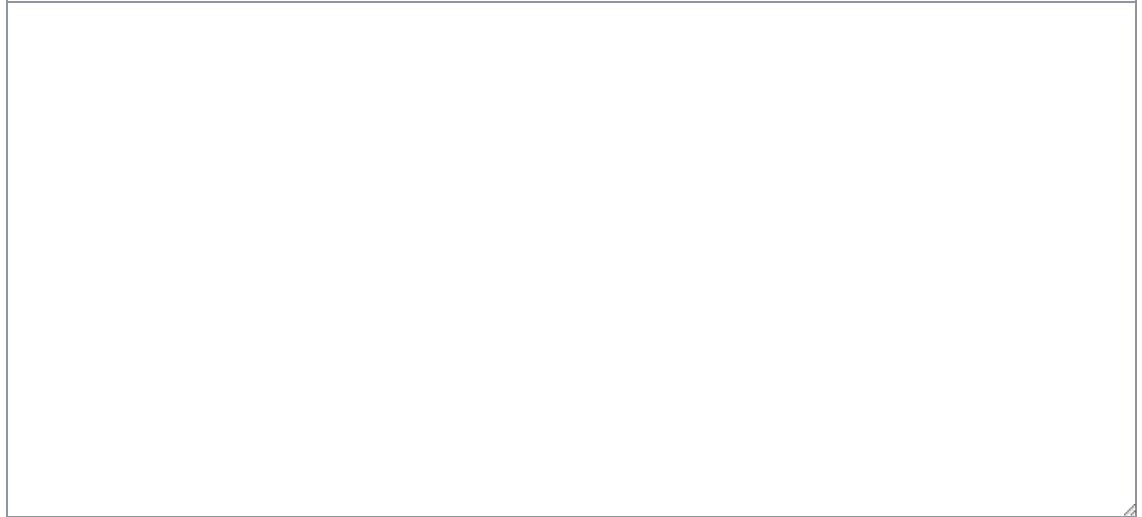
Frage **25**Nicht
beantwortet

Nicht bewertet

Hier können Sie getroffenen Annahmen vermerken. Geben Sie eindeutig an zu welcher Frage Sie eine Annahme getroffen haben.

Bei der Bewertung können nur Annahmen berücksichtigt werden, die hier angegeben wurden.

Zusätzlich zur Eingabe von Freitext ist auch der Upload von gescannten oder abfotografierten Notizen möglich.



[◀ Gesundheits- und
Selbständigkeitserklärung 2. Termin](#)

Direkt zu:

[Datei-Download Online-Klausur ▶](#)