



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik

Individuelle Laboraufgaben

Seite		erledigt
2	Ampel : Variante 1, Variante 2, Variante 3, Variante 4	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
11	Codegeneratoren : 51 bis 70	<input type="checkbox"/>
71	Digitaluhr	<input type="checkbox"/>
76	Elektronischer Würfel	<input type="checkbox"/>
83	Kreuzung : Variante 1, Variante 2, Variante 3, Variante 4	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
94	LED-Mux_Demux	<input type="checkbox"/>
104	PWM	<input type="checkbox"/>
111	RGB-Würfel : Variante 1, Variante 2	<input type="checkbox"/> <input type="checkbox"/>
119	Siebensegment-Dekoder-BCD	<input type="checkbox"/>
130	Siebensegment-Dekoder-Hex	<input type="checkbox"/>

Allgemeine Laboraufgaben siehe obige Homepage



Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631 – 3724 - 5357

<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Ampel Variante 1
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der Signallampen sind nur NOT, OR, AND, NOR, NAND und XOR - Gatter zugelassen.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631 – 3724 - 5357

<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Ampel Variante 2
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der Signallampen verwenden Sie u.a. Bausteine vom Typ „SN74LS138“ oder „SN74LS139“.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631 – 3724 - 5357

<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Ampel Variante 3
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Bausteine des Typs „SN74LS161“. Für die Schaltnetze zur Ansteuerung der Signallampen sind nur NOT, OR, AND, NOR, NAND und XOR - Gatter zugelassen.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631 – 3724 - 5357

<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Ampel Variante 4
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Bausteine des Typs „SN74LS161“. Für die Schaltnetze zur Ansteuerung der Signallampen verwenden Sie u.a. Bausteine vom Typ „SN74LS138“ oder „SN74LS139“.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

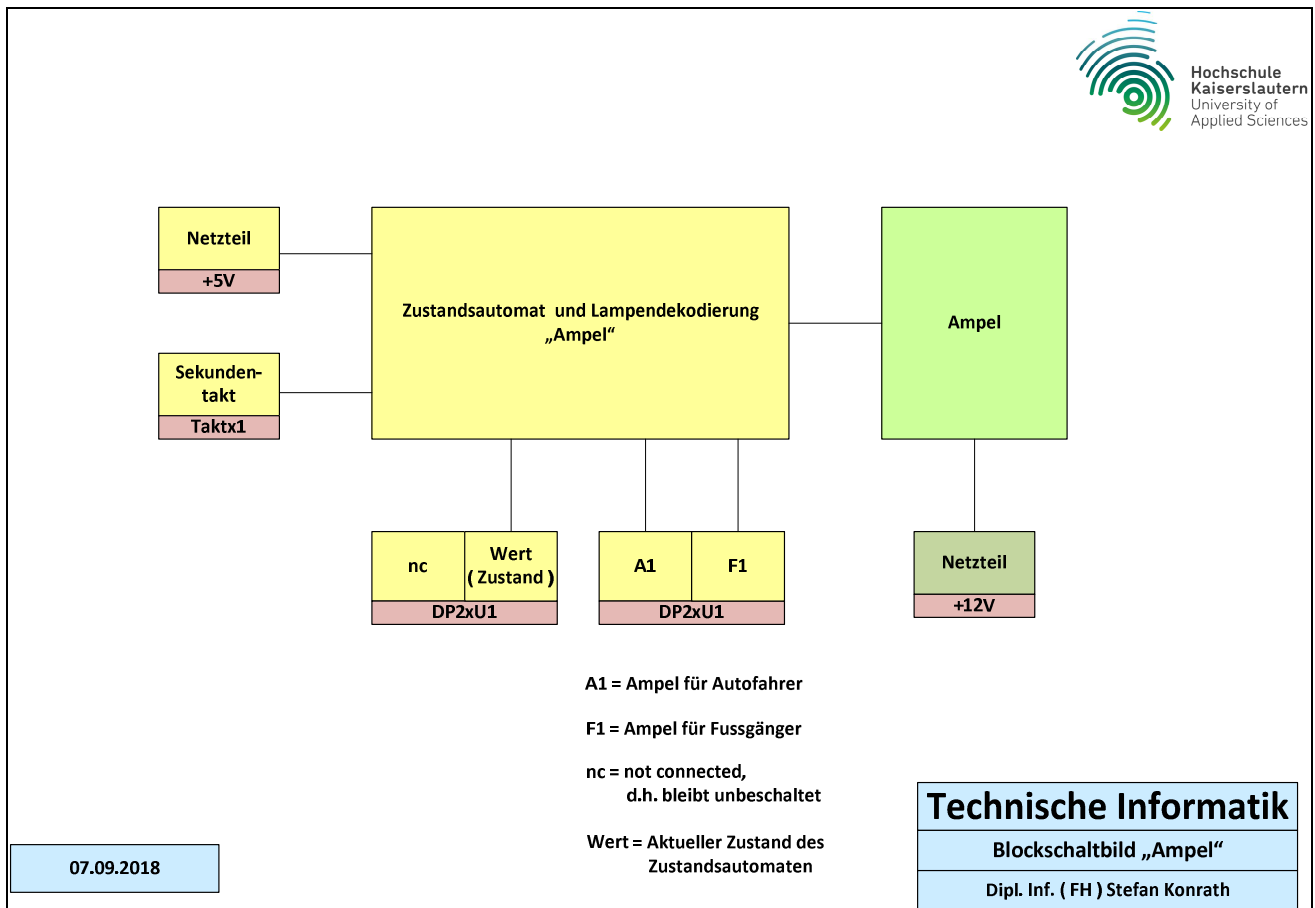
<http://www.hs-kl.de/~stefan.konrath>

Laborversuch „Ampel“



Aufgabenstellung ist hier die Realisierung der digitalen Steuerung einer elektronischen Ampelanlage .

Die komplette Aufgabenstellung ist im Blockschaltbild auf dieser Seite prinzipiell aufgeführt.
Es dient als Grundlage für Ihre Applikation.



Zur Realisierung Ihrer Applikation verwenden Sie bitte nur die digitalen Bausteine, welche in der unten aufgeführten Liste „Verfügbare Digital-Bausteine“ aufgelistet sind.

Hinweis : Einschränkungen zur Bausteinauswahl siehe Deckblatt !

Der Zustandszähler des Zustandsautomaten soll in **synchroner Bauweise** realisiert werden !

Verfügbare Digital-Bausteine							
Bezeichnung	Funktion		Bezeichnung	Funktion		Bezeichnung	Funktion
SN74LS00	4 x NAND mit je 2 Eingängen		SN74LS32	4 x OR mit je 2 Eingängen		SN74LS139	2 Bit Bin.Dekoder, DEMUX 2 zu 4
SN74LS02	4 x NOR mit je 2 Eingängen		SN74LS51	AND / NOR - Kombination		SN74L151	8 zu 1 Multiplexer
SN74LS04	6 x Inverter		SN74LS55	AND / NOR - Kombination		SN74L161	Synchr.prog. 4-Bit Bin.Zähler,Clr
SN74LS08	4 x AND mit je 2 Eingängen		SN74LS74	Dual-D-Flip-Flop		SN74LS175	Quad, synchr. D-Flip-Flop
SN74LS10	3 x NAND mit je 3 Eingängen		SN74LS85	4 Bit Vergleicher		SN74LS260	2 x NOR mit je 5 Eingängen
SN74LS11	3 x AND mit je 3 Eingängen		SN74LS86	4 x EXOR mit je 2 Eingängen		SN74LS393	Asynchr. 4 Bit Binärzähler
SN74LS14	6 x Inverter, Schmitt-Trigger		SN74LS125	4-Bus-Leitungs-treiber, tristate		SN74LS573	8-Bit Latch
SN74LS20	2 x NAND mit je 4 Eingängen		SN74LS126	4-Bus-Leitungs-treiber, tristate		GAL16V8	Siebensegment-Dekoder
SN74LS21	2 x AND mit je 4 Eingängen		SN74LS133	1 x NAND mit je 13 Eingängen			
SN74LS27	3 x NOR mit je 3 Eingängen		SN74LS136	4 x EXOR mit je 2 Eingängen			
SN74LS30	1 x NAND mit je 8 Eingängen		SN74LS138	3 Bit Bin.Dekoder, DEMUX 3 zu 8			

In der Tabelle „Zustandsfolge der Ampelschaltung mit TTL-Logikbausteinen“ ist die geforderte Ablauffolge der Ansteuerung der entsprechenden Signalleuchten dargestellt. Die Taktfrequenz der Zustandsabfolge beträgt 1Hz. Dieser Takt wird von der Taktgeberplatine „Taktx1“ geliefert. Hierzu ist dort die entsprechende Frequenz am Drehschalter einzustellen. Die detaillierte Beschreibung der Taktgeberplatine „Taktx1“ sowie der Siebensegmentanzeigeplatine „DP2xU1“ finden Sie im Anhang dieses Dokumentes.

Zustandsfolge der Ampelschaltung mit TTL-Logikbausteinen									
Wert	A1	2 ⁰	2 ¹	2 ²	F1	2 ⁰	2 ¹		
Schritt		A1-Rot	A1-Gelb	A1-Grün		F1-Rot	F1-Grün		Gelbe Signalleuchte
0	1	X			1	X			
1	3	X	X		1	X			
2	4			X	1	X			
3	4			X	1	X			
4	4			X	1	X			
5	4			X	1	X			
6	4			X	1	X			
7	4			X	1	X			
8	4			X	1	X			
9	2		X		1	X			
10	1	X			1	X			
11	1	X			2		X		X (blinkt, 1 Hz)
12	1	X			2		X		X (blinkt, 1 Hz)
13	1	X			2		X		X (blinkt, 1 Hz)
14	1	X			2		X		X (blinkt, 1 Hz)
15	1	X			2		X		X (blinkt, 1 Hz)
Stefan Konrath					Version V1.0				

Die Verknüpfung zwischen den Signalleuchten und Ihrer Applikation lässt sich über die u.a. Tabelle ersehen.

Zuordnung Kabelfarbe <--> Signalleuchten	
Kabelfarbe	Zuordnung
Schwarz	A1-rot
Violett	A1-Gelb
Grau	A1-Grün
Rosa	F1-Rot
Gelb	F1-Grün
Braun	Gelbe Signalleuchte
Weiss	Reserve-Ausgang-7
Grün	Reserve-Ausgang-8
Gelb-Weiss	Eingang-1
Braun-Grün	Eingang-2
Grau-Rosa	Eingang-3
Weiss-Grün	Eingang-4
Gelb-Braun	Eingang-5
Weiss-Grau	Eingang-6
Grau-Braun	Eingang-7
Rot-Blau	Eingang-8
Blau	Gnd (Masse, Minus)

Stefan Konrath	Version 1.0
----------------	-------------



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 51
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

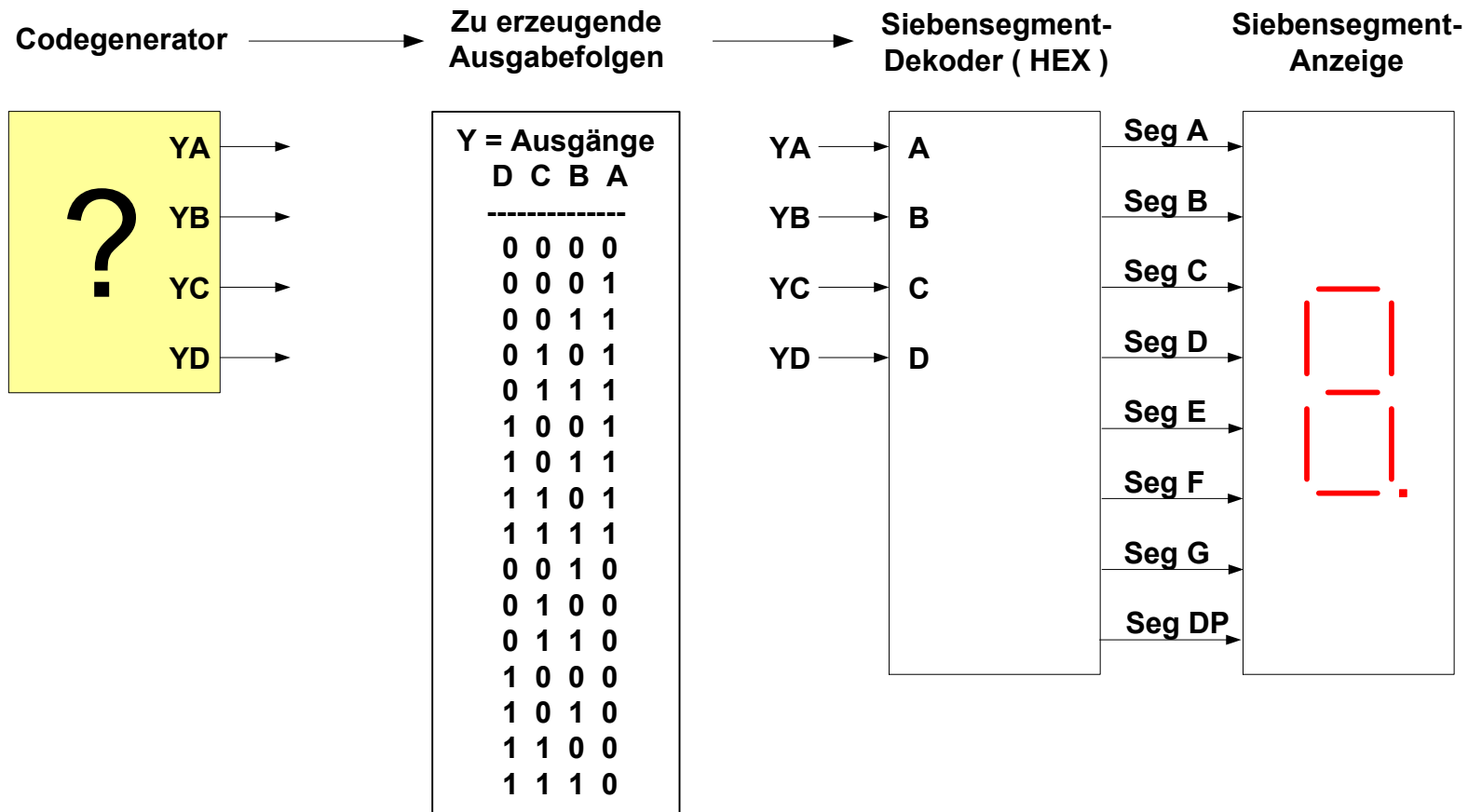
Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (51)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 52
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

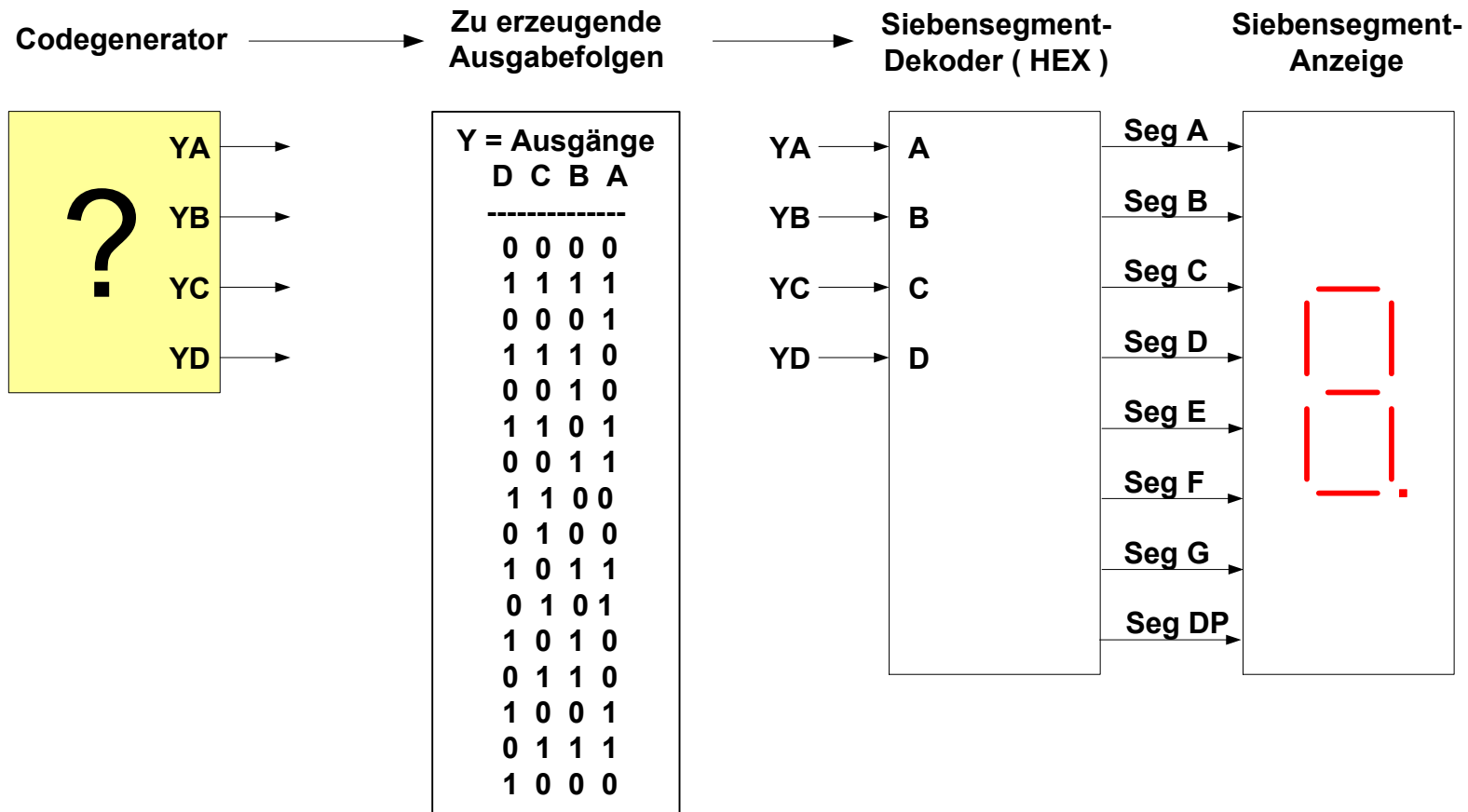
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (52)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 53
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

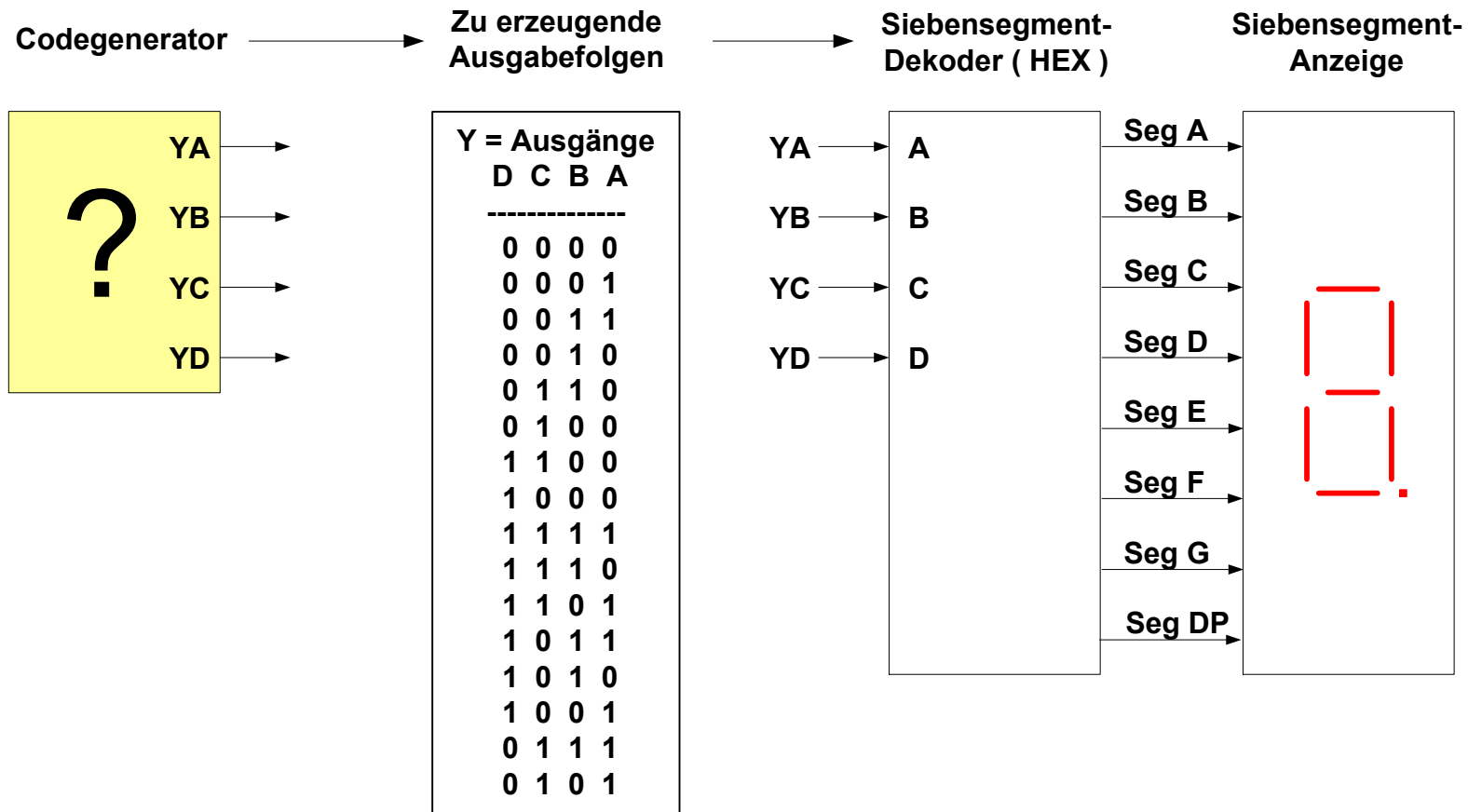
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (53)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 54
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

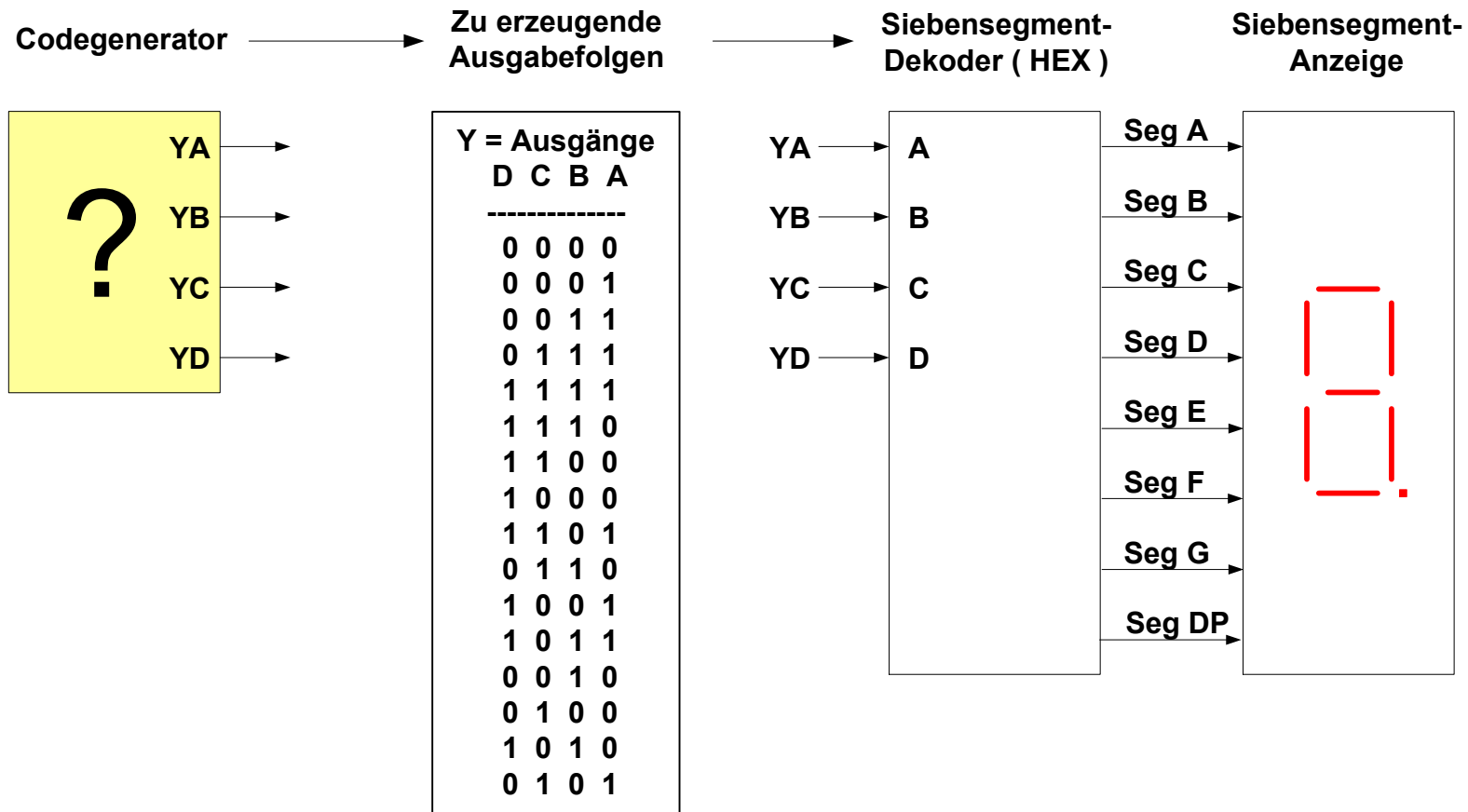
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit **?** hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (54)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 55
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

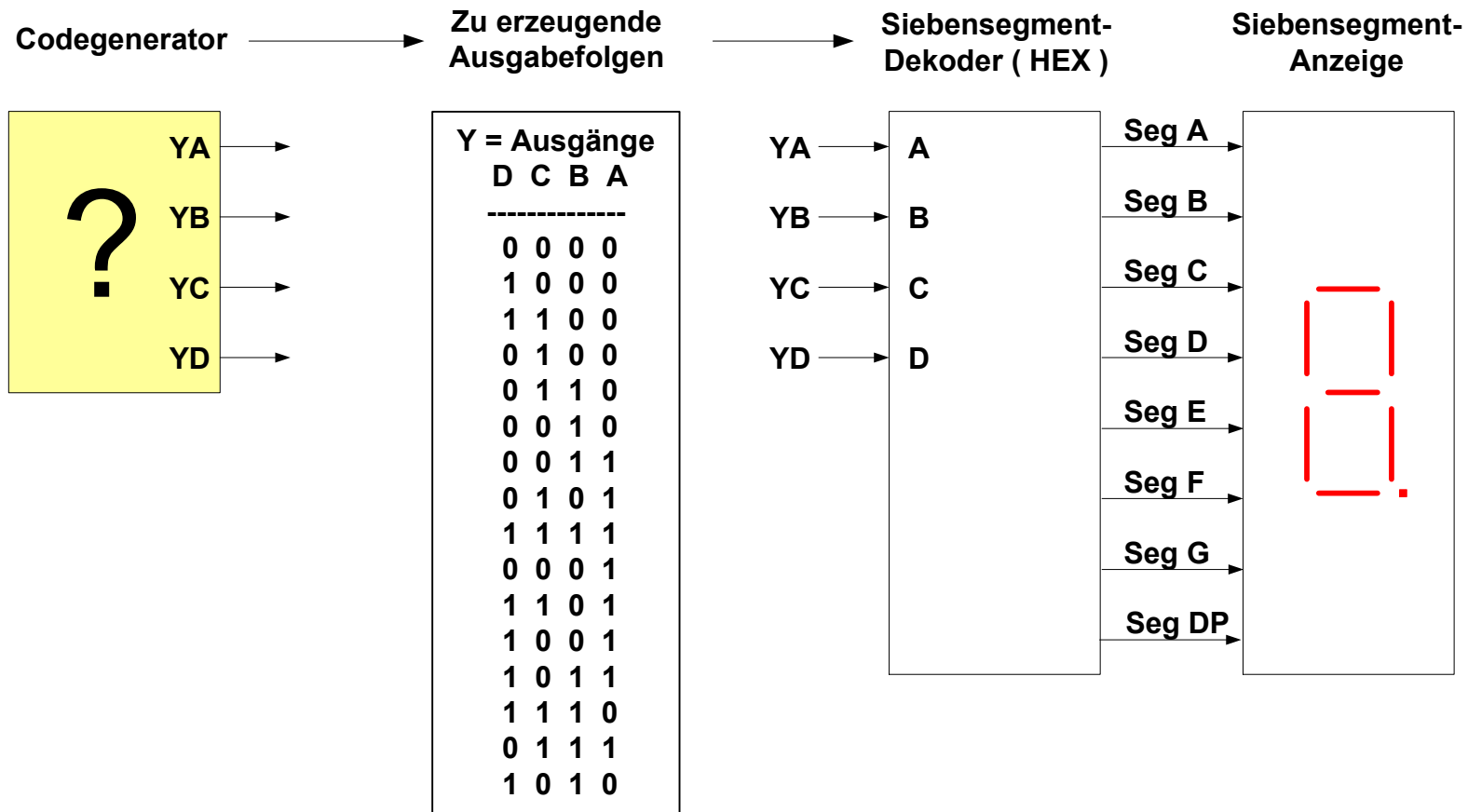
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (55)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 56
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

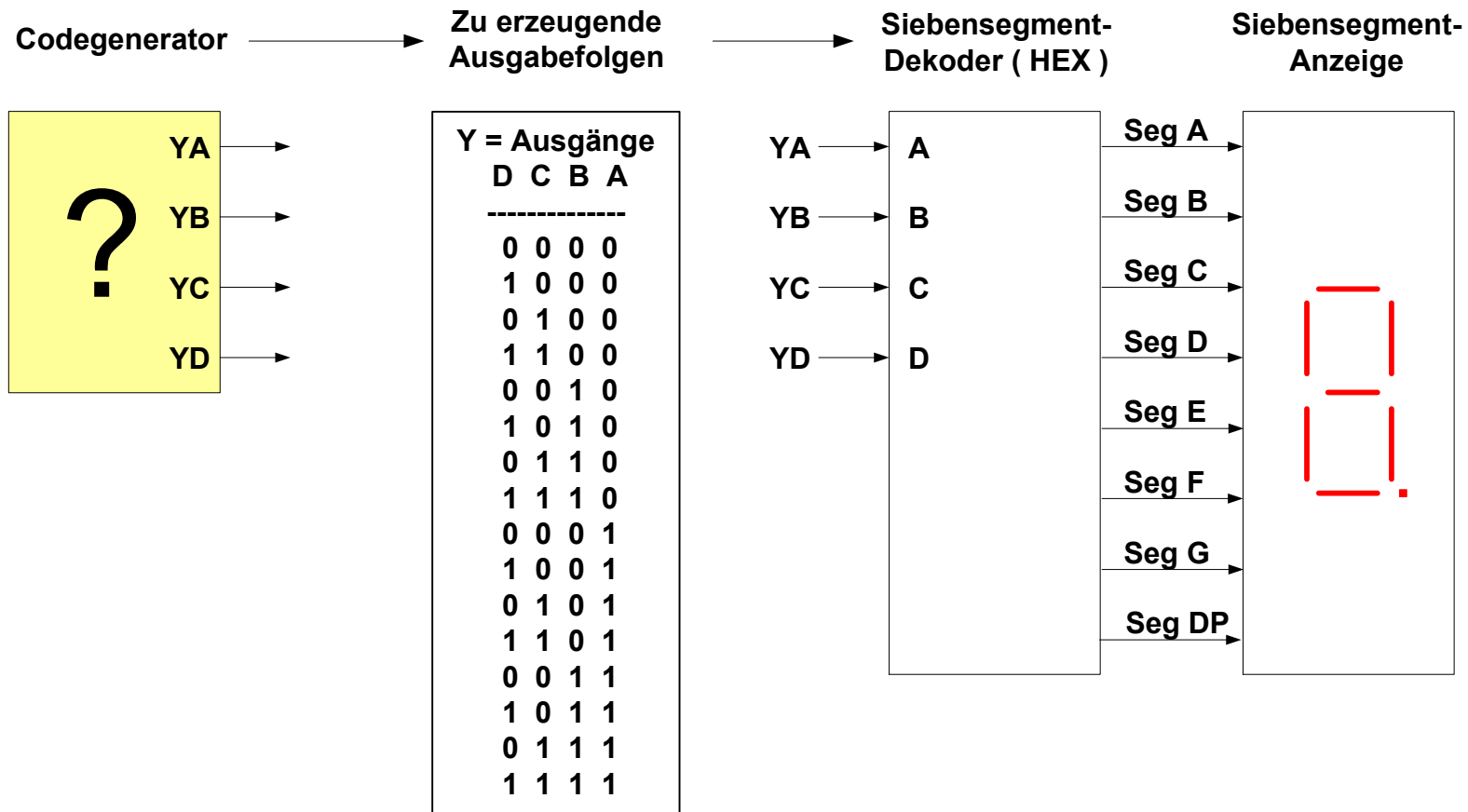
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (56)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 57
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

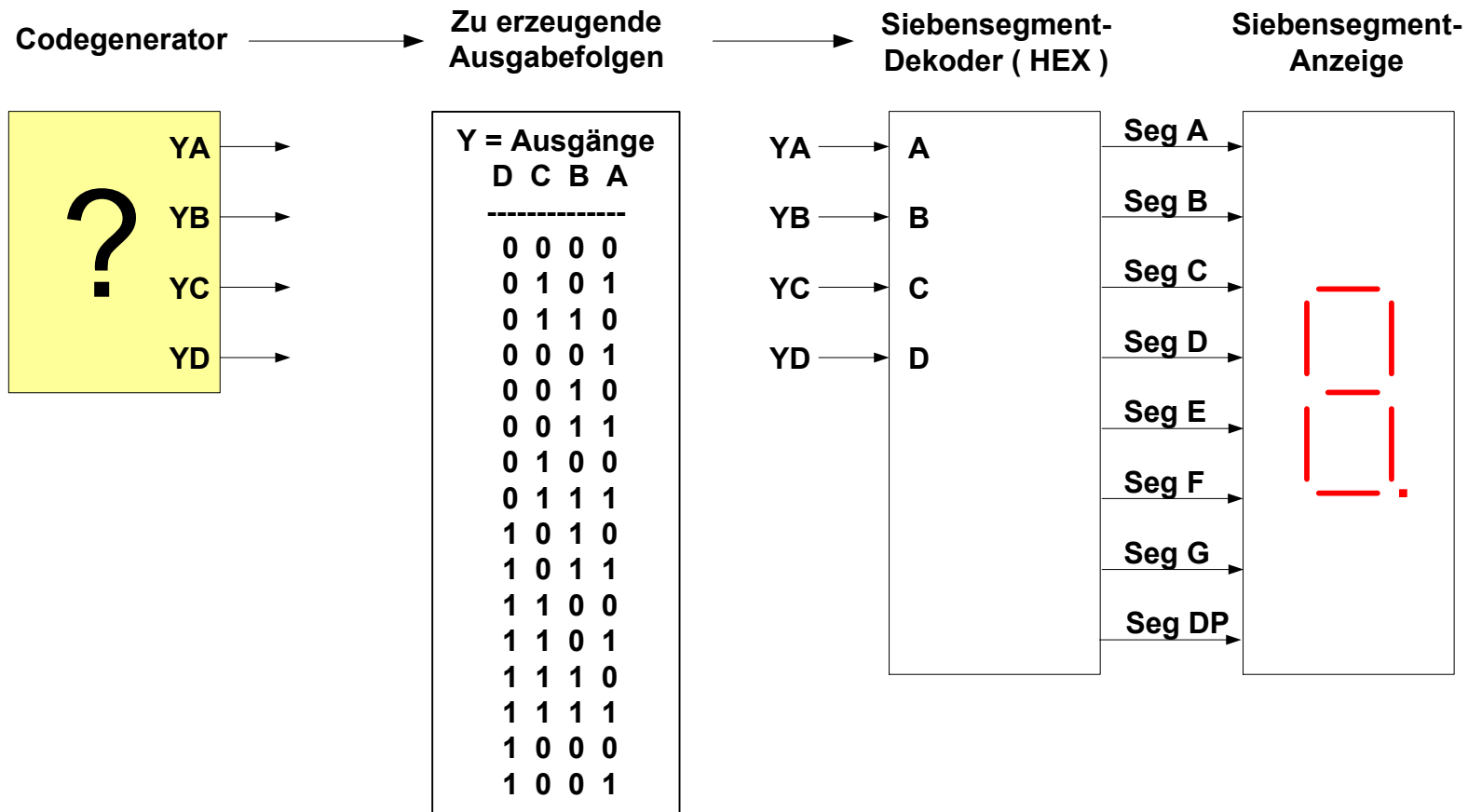
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

12.09.2018

Technische Informatik

Praktische Ausarbeitung TI (57)

Dipl. Inf. (FH) Stefan Konrath



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 58
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

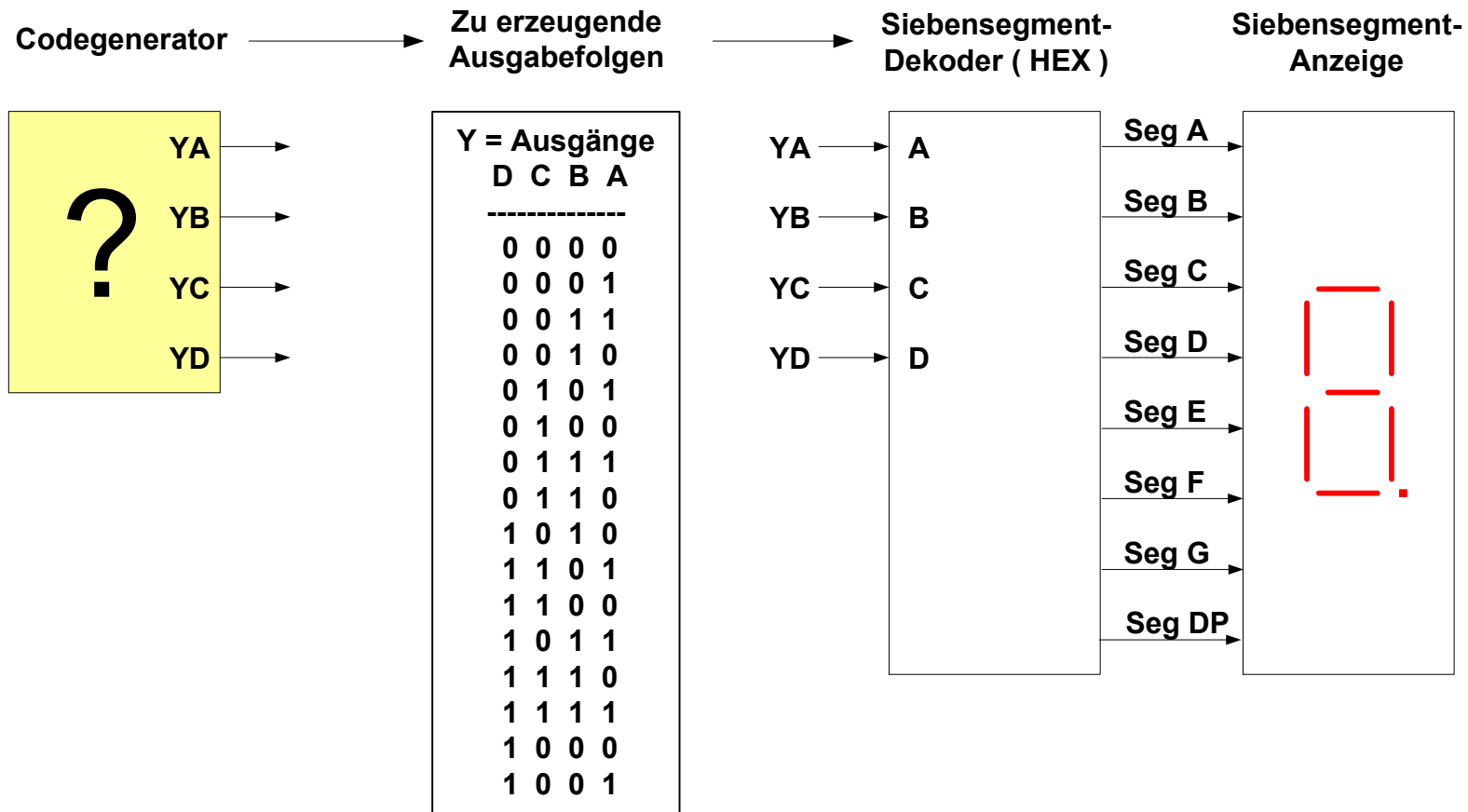
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 59
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

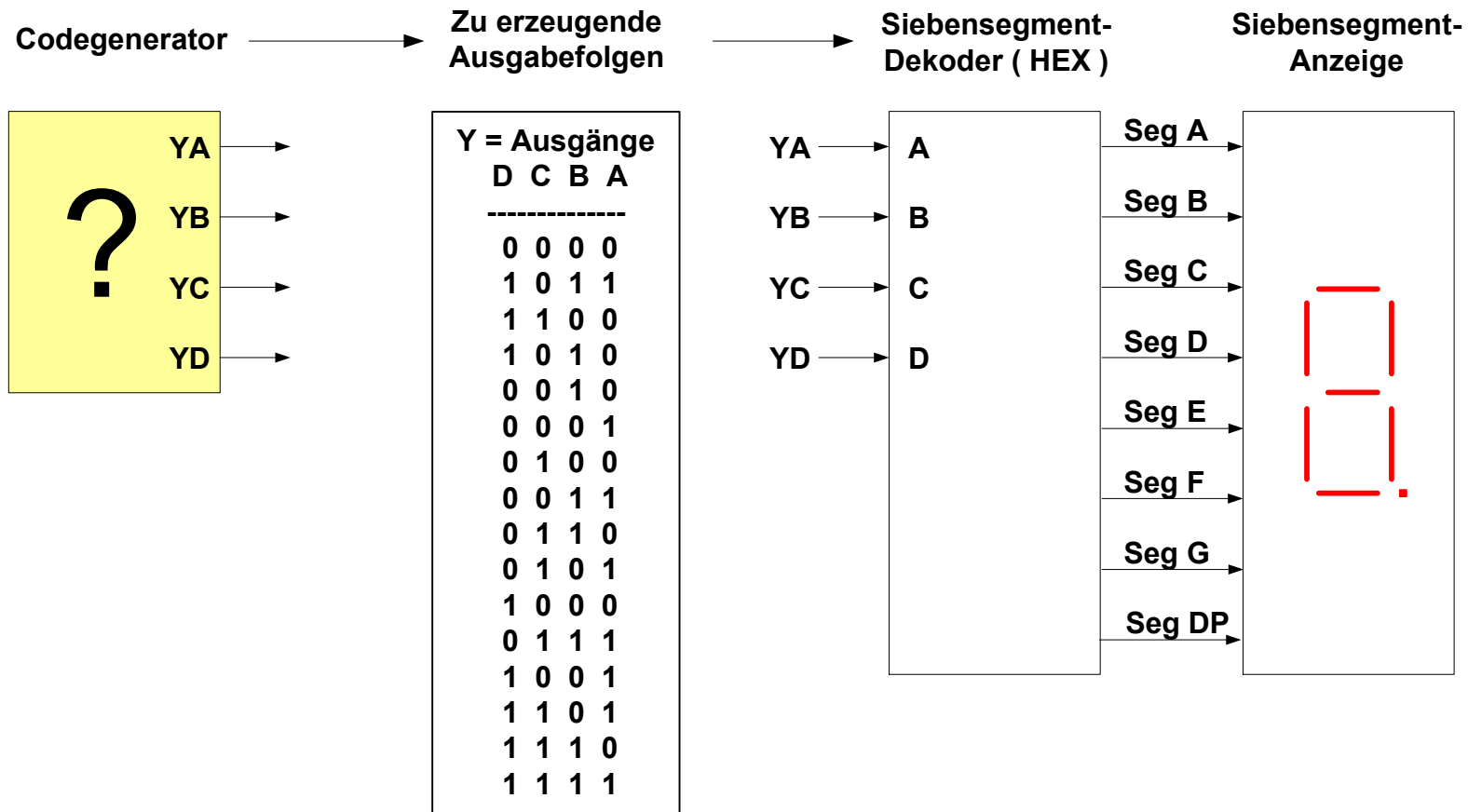
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (59)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 60
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

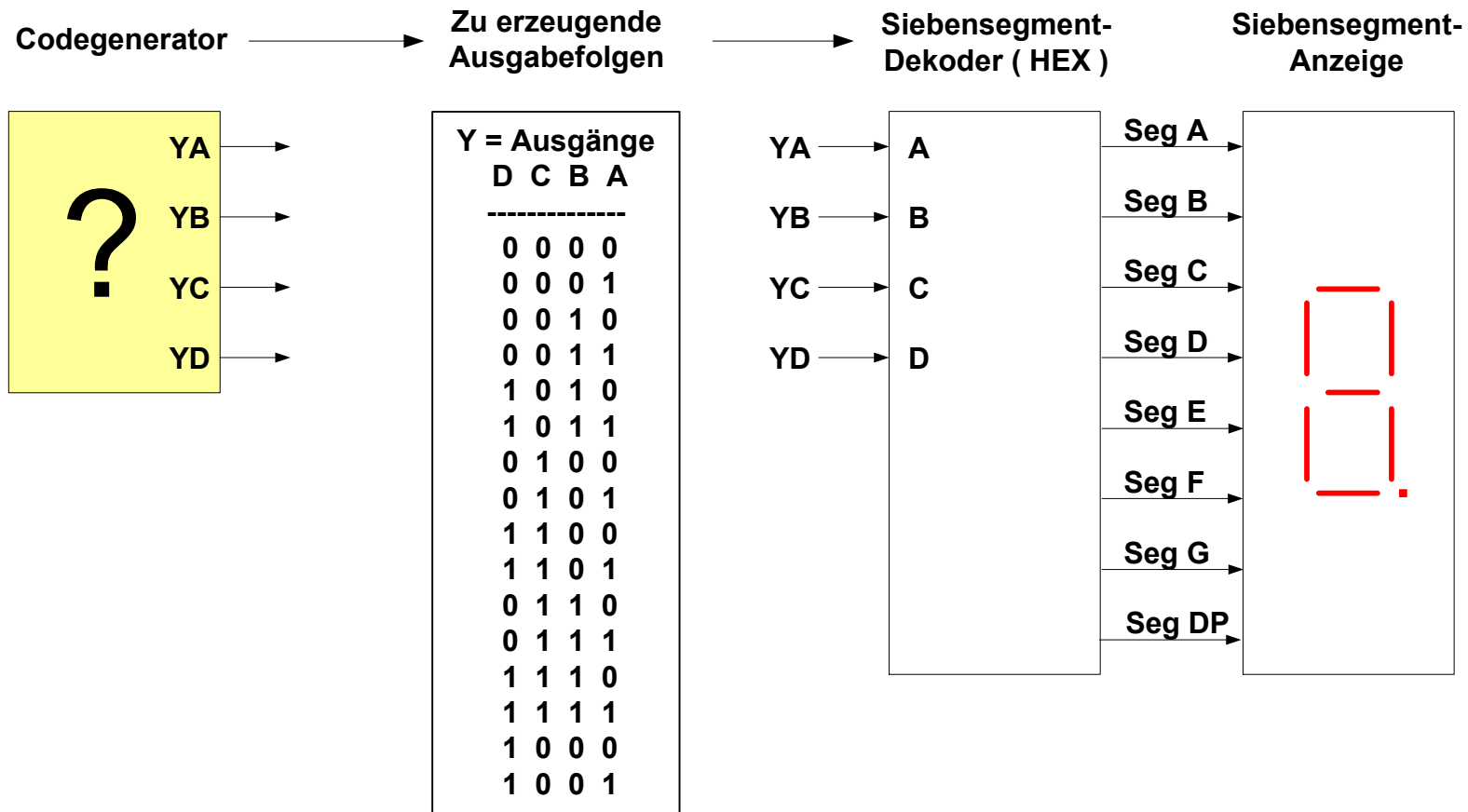
Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (60)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 61
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

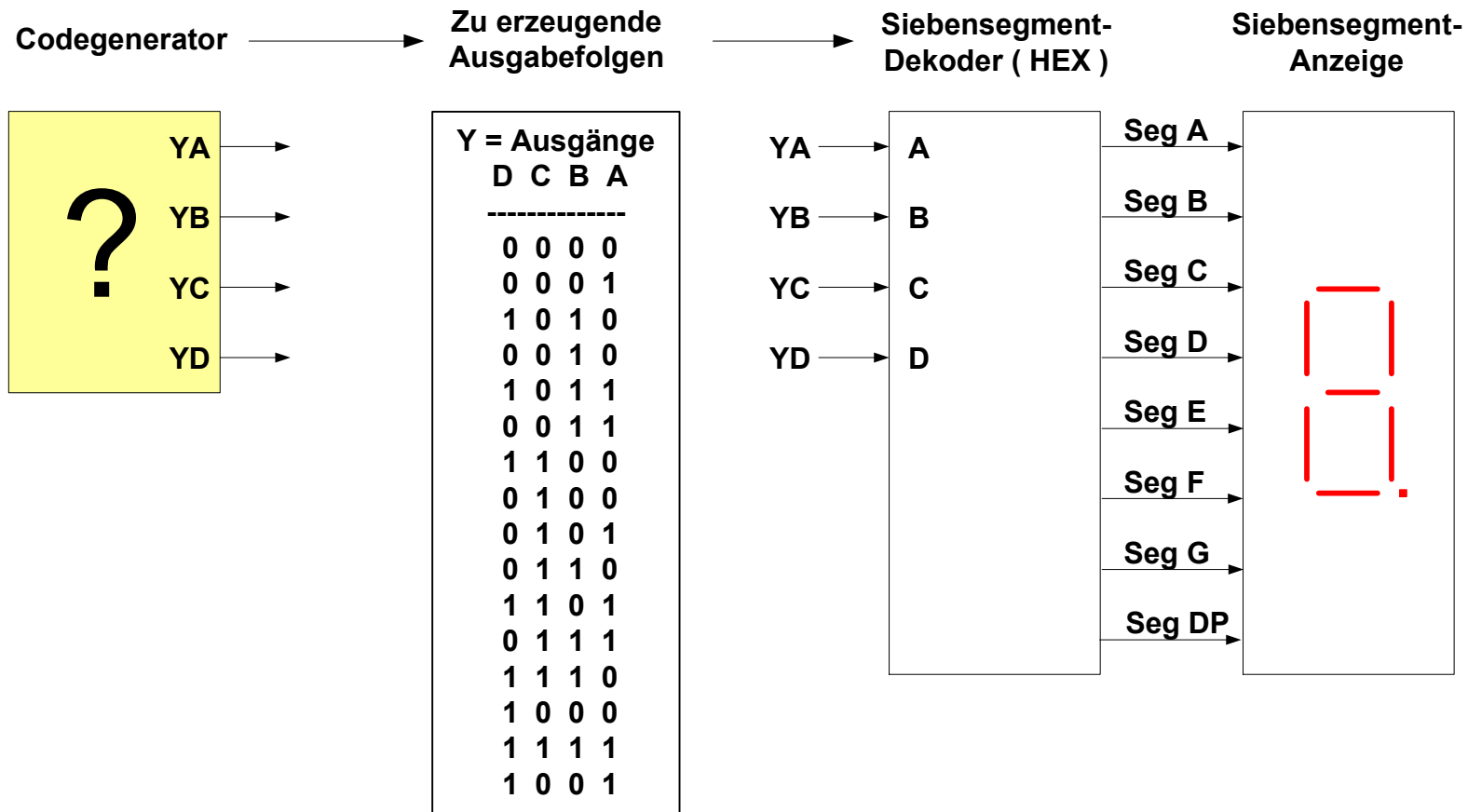
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (61)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 62
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

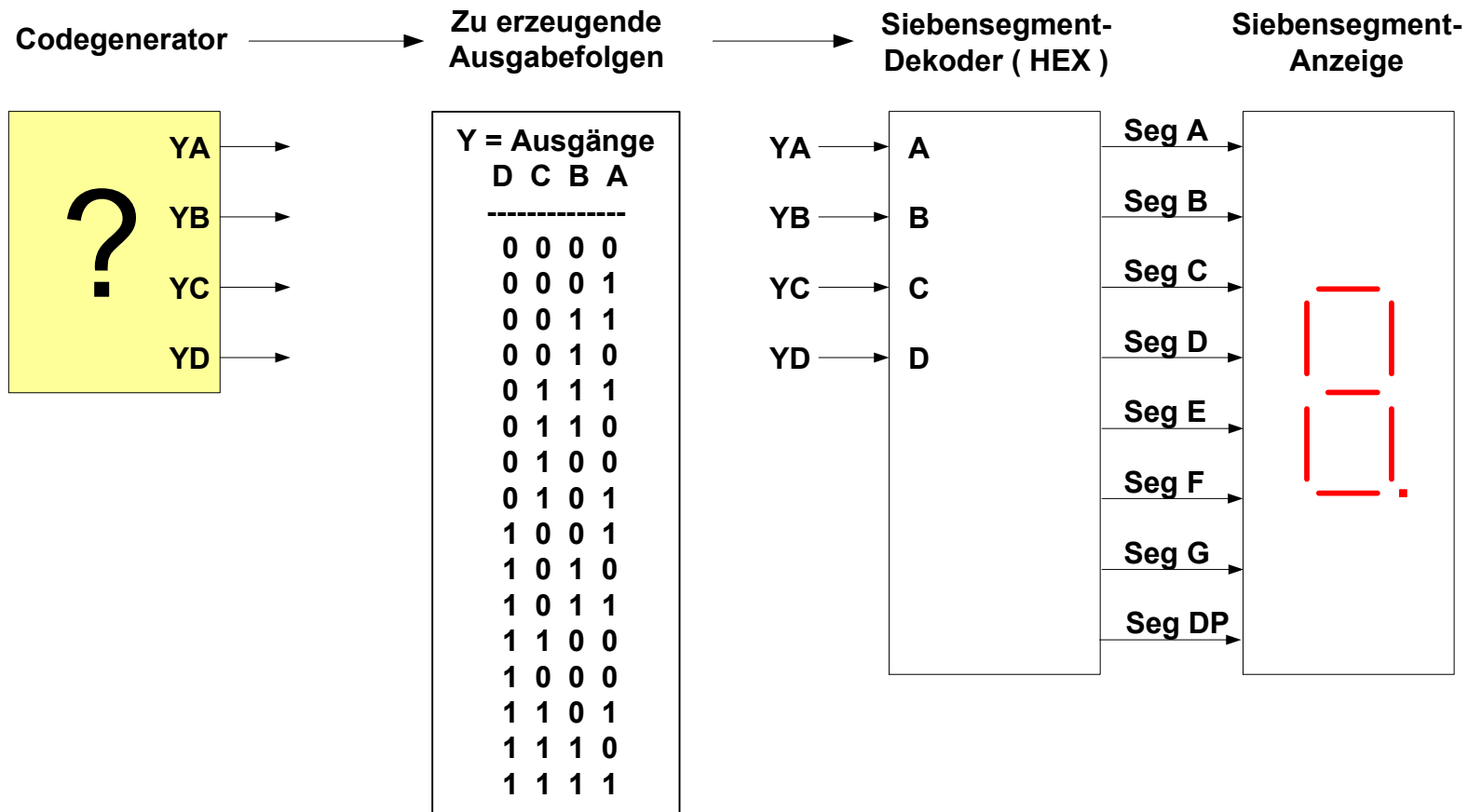
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

12.09.2018

Technische Informatik

Praktische Ausarbeitung TI (62)

Dipl. Inf. (FH) Stefan Konrath



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 63
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

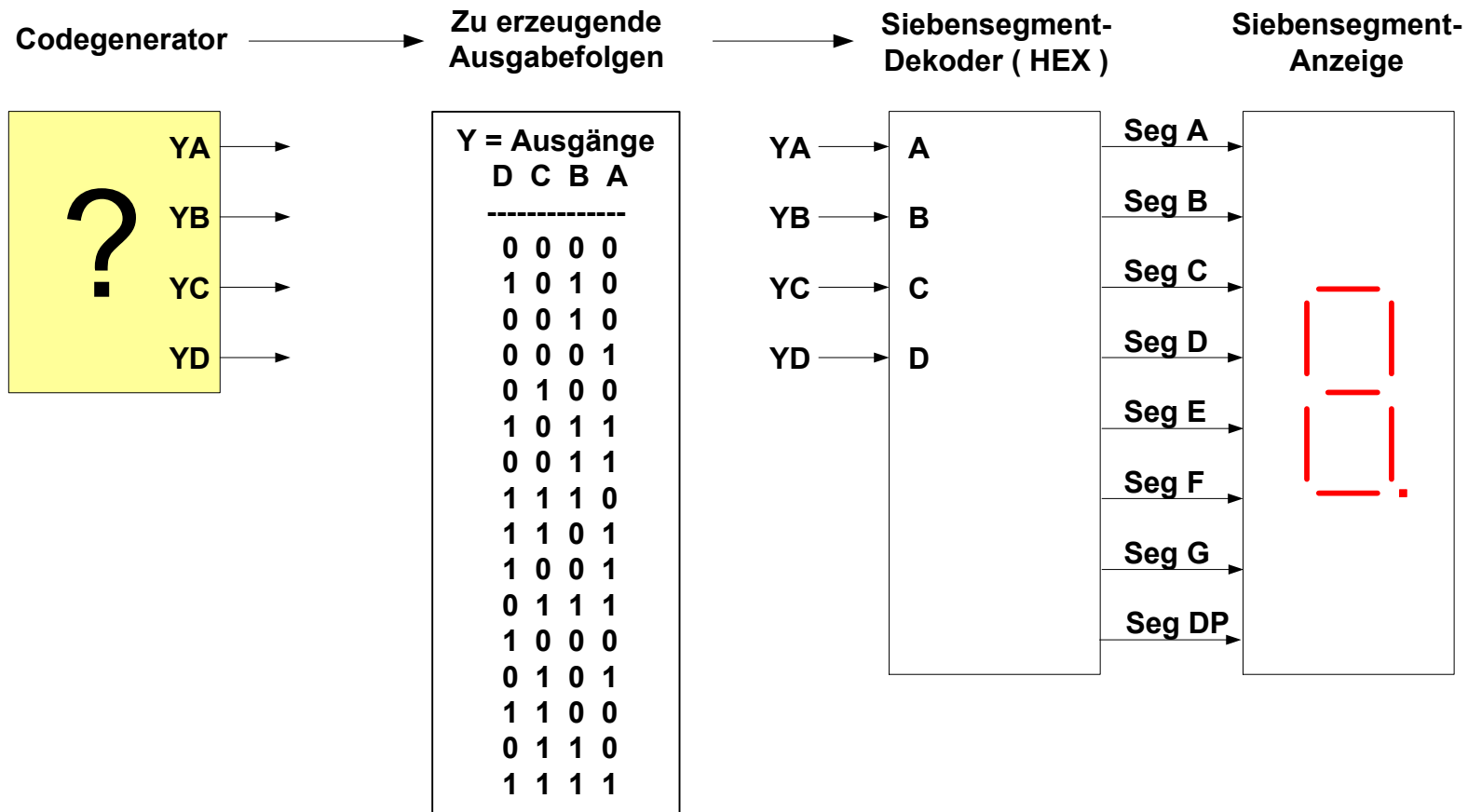
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

12.09.2018

Technische Informatik

Praktische Ausarbeitung TI (63)

Dipl. Inf. (FH) Stefan Konrath



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 64
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

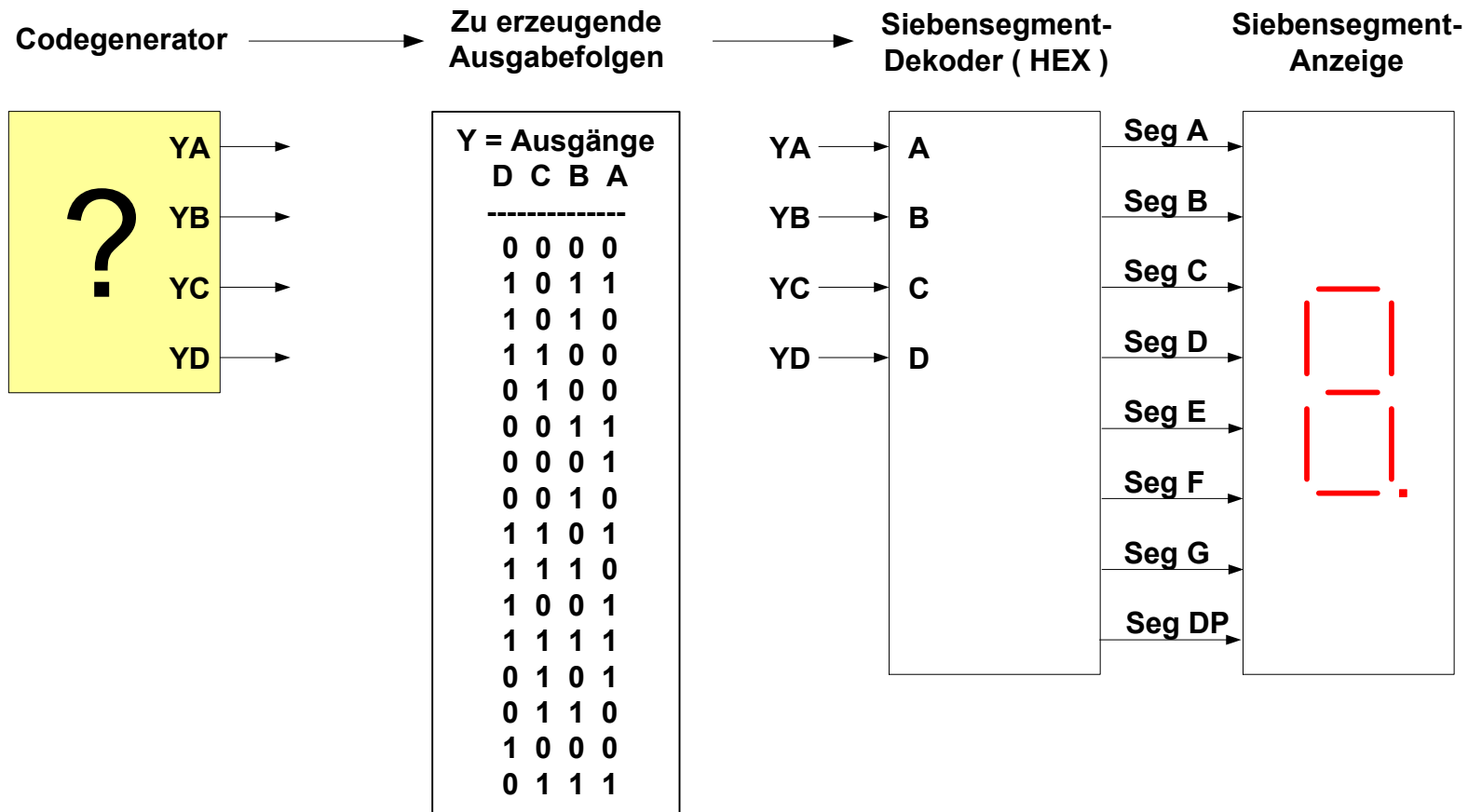
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (64)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 65
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

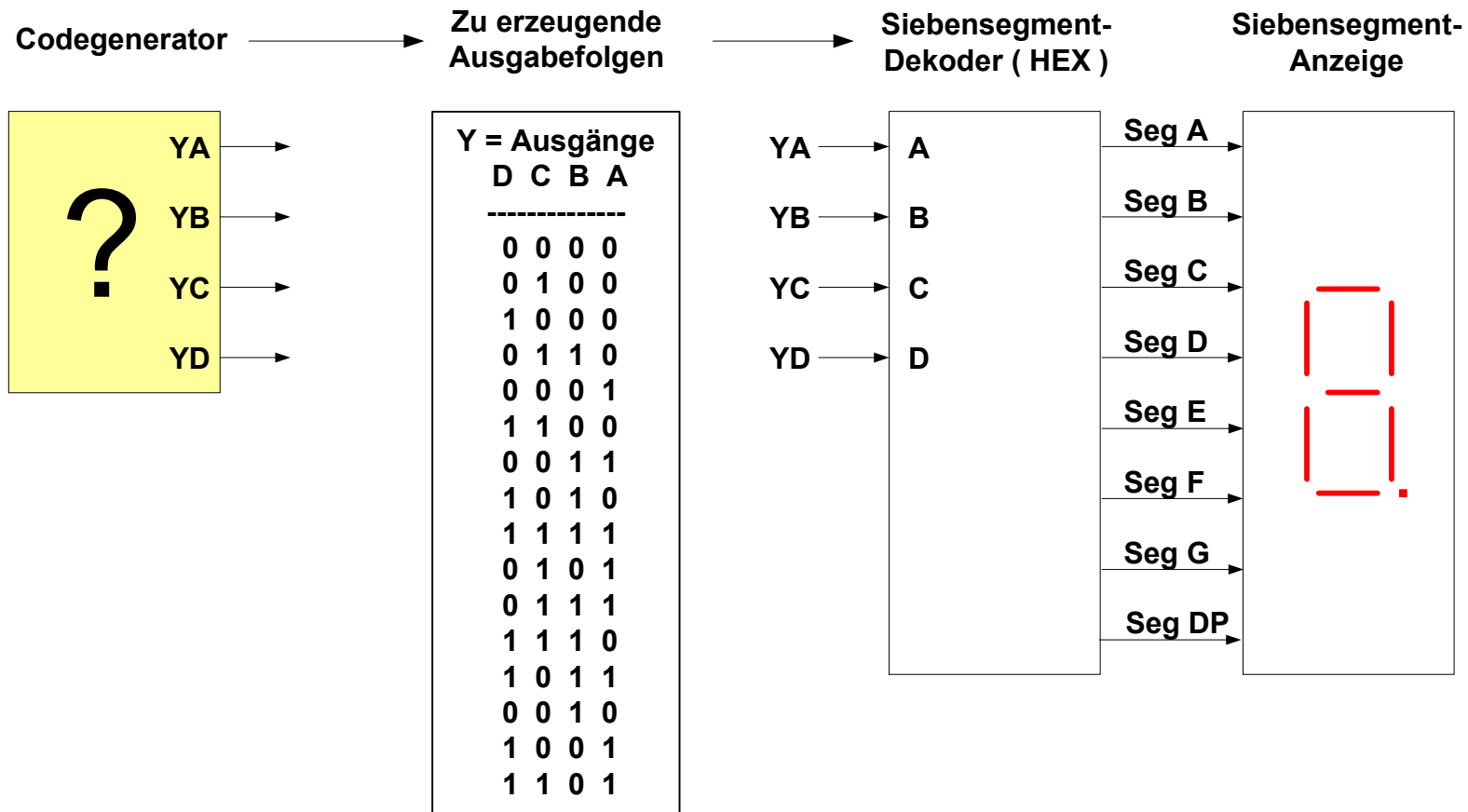
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (65)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 66
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

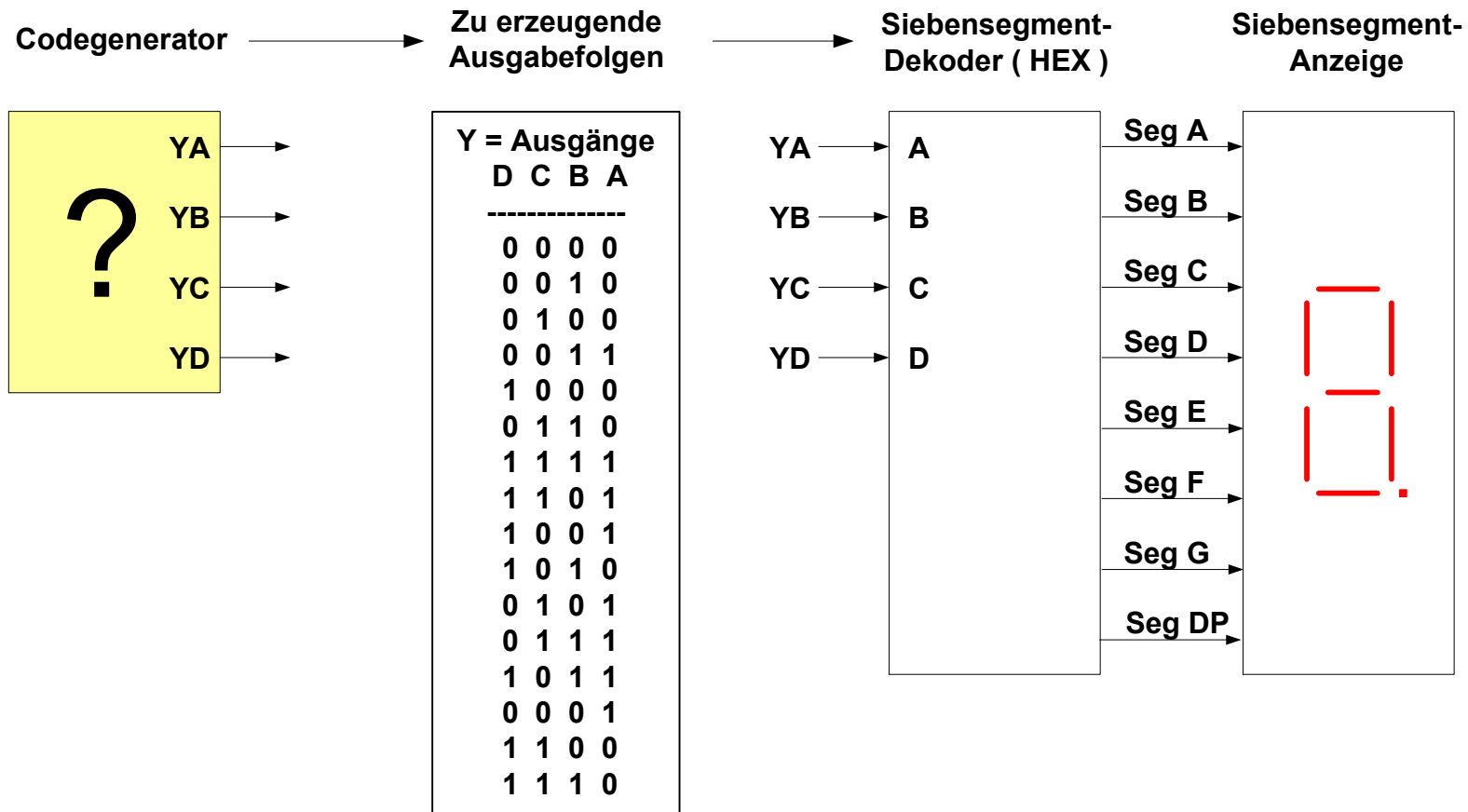
Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

23.10.2018

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 67
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

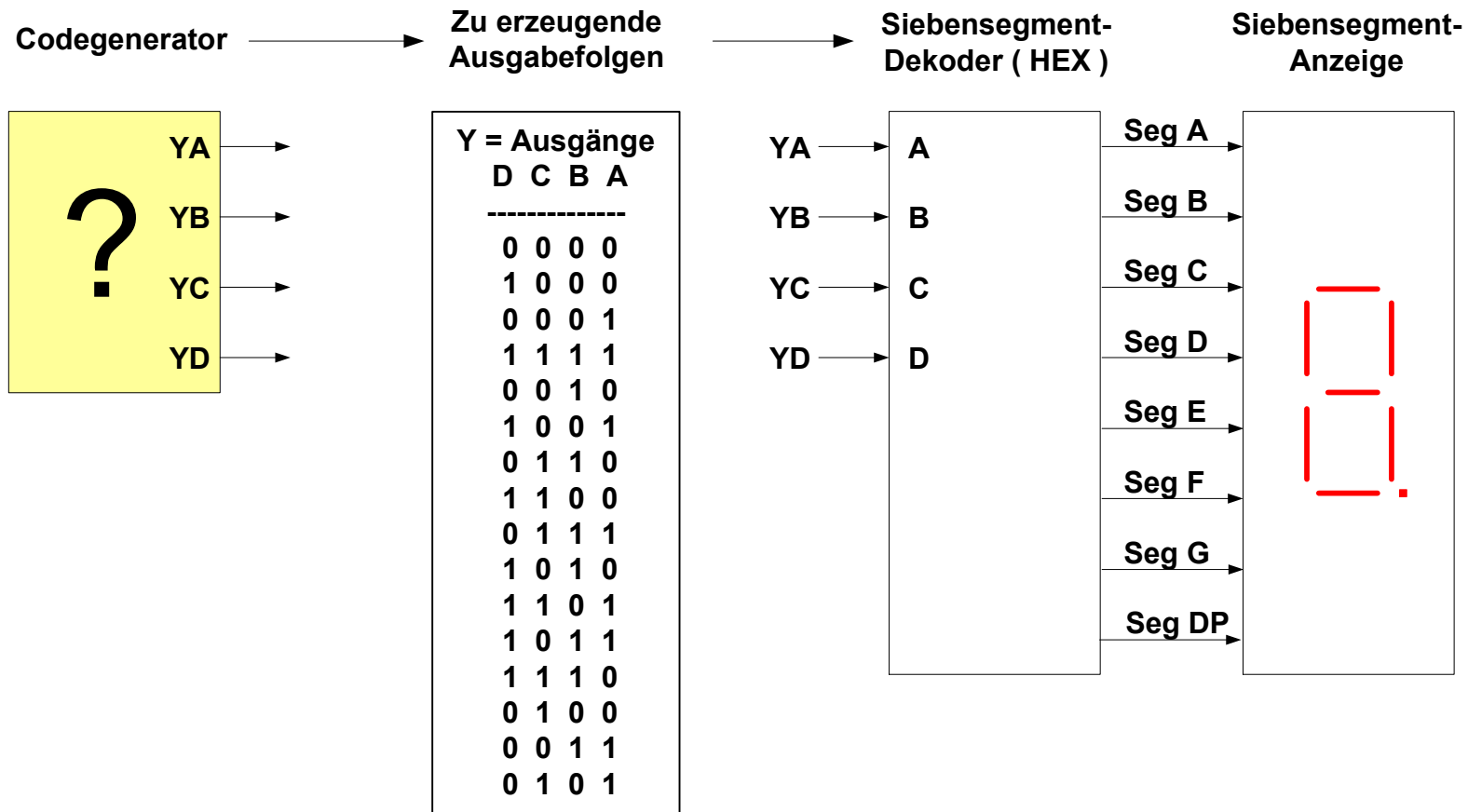
Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (67)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 68
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

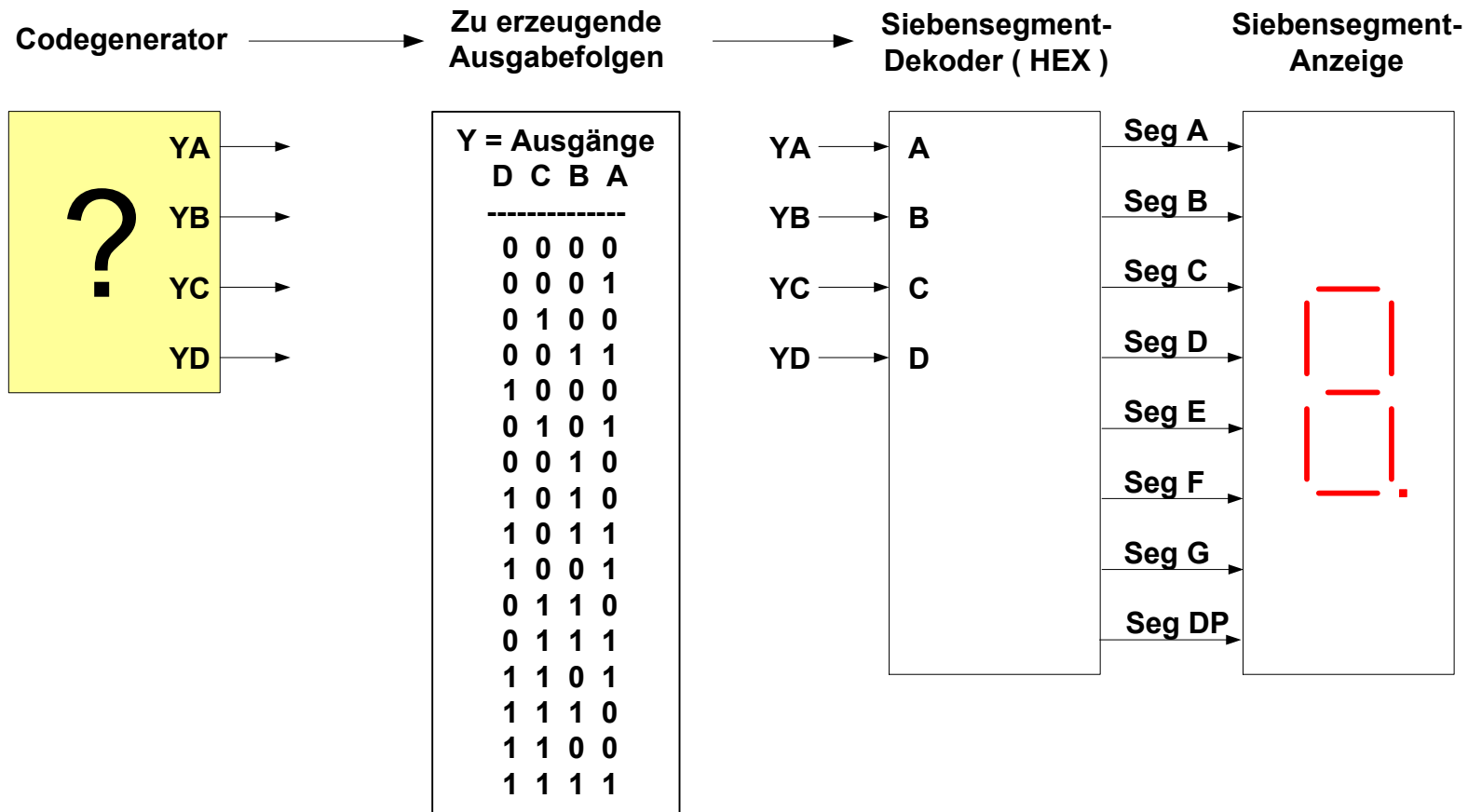
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (68)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 69
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

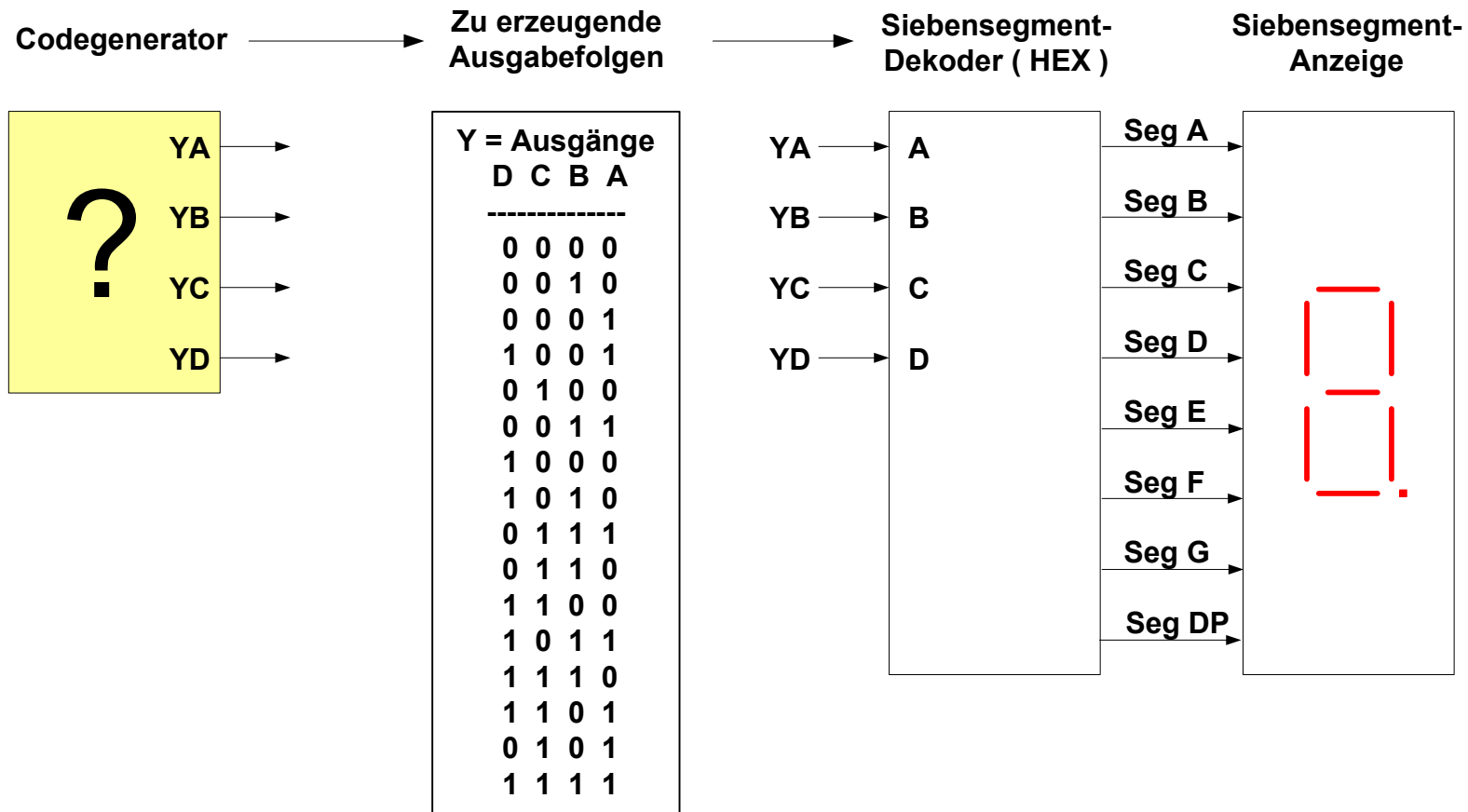
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (69)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Codegenerator 70
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine synchrone Zählerschaltung, welche die auf der folgenden Seite gegebene Zahlenfolge fortlaufend generiert. Die Zählerschaltung soll asynchron rücksetzbar sein. Verwenden Sie zur kompakten Realisierung den Baustein SN74LS175 !

Die D-FF-Ausgänge (YA ... YD) im Folgeblatt, sind direkt über eine Siebensegment-Anzeige zu visualisieren. D.h. zwischen den D-FF-Ausgängen und den Eingängen des Siebensegment-Decoders (A ... D) darf sich kein Schaltnetz befinden !

Die Zählfolge lt. der Tabelle „Zu erzeugende Ausgabefolgen“ ist durch ein geeignetes Schaltnetz an den D-FF-Eingängen des SN74LS175 zu realisieren. D.h. durch Rückführung der aktuellen Zustände der D-FF-Ausgänge wird über ein Schaltnetz an den D-FF-Eingängen der Folgezustand des Zählers lt. Tabelle festgelegt.

Zum Test der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Zählerstands das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

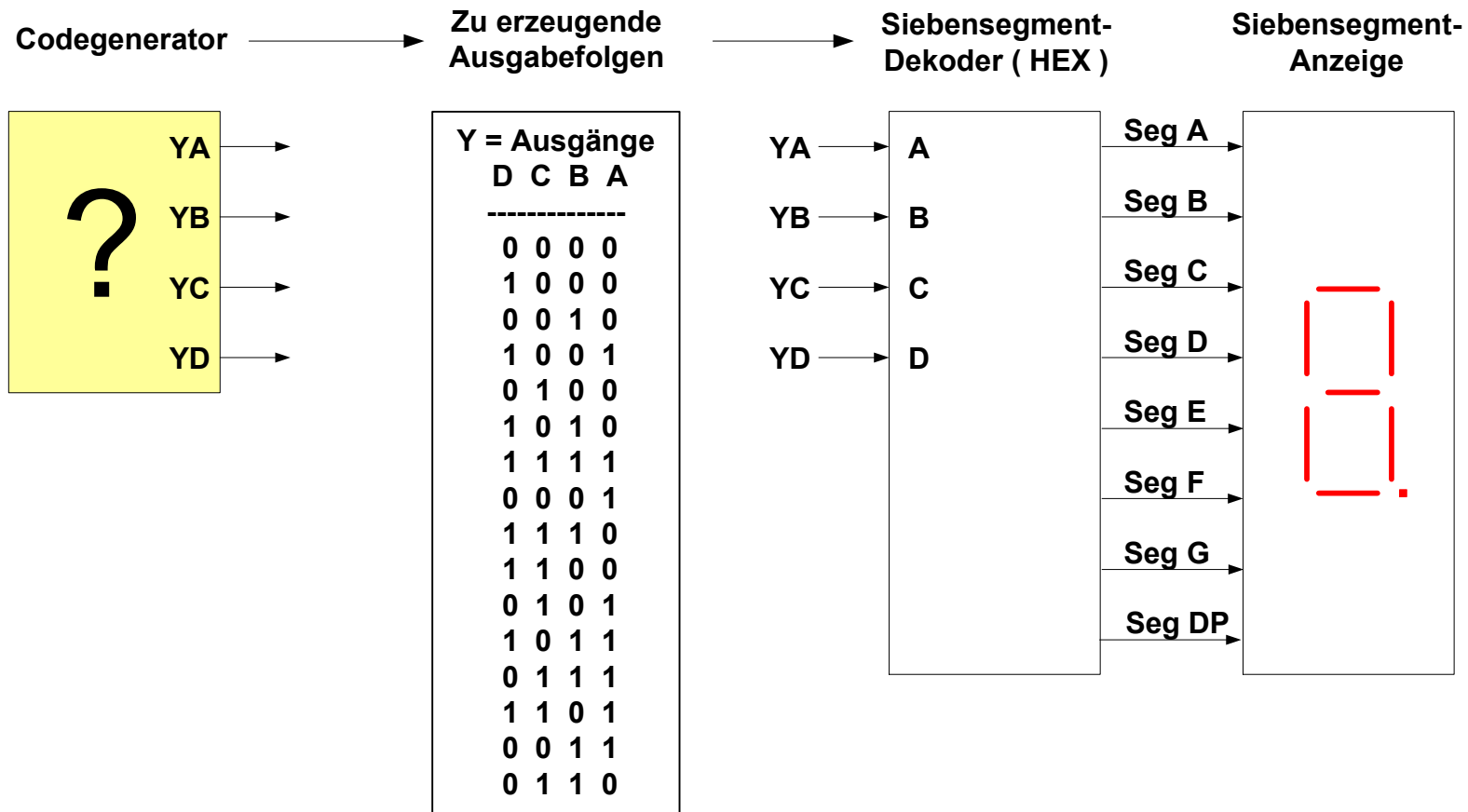
23.10.2018

Technische Informatik

Aufgabenstellung Codegenerator

Dipl. Inf. (FH) Stefan Konrath

4-Bit-Codegenerator



Hinweis : Der mit ? hinterlegte Funktionsblock ist von Ihnen zu realisieren !

Technische Informatik

Praktische Ausarbeitung TI (70)

Dipl. Inf. (FH) Stefan Konrath

12.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Digitaluhr
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Praktische Aufgabe
Technische Informatik
Studiengang *Angewandte Informatik*

Praktische Aufgabe „Digitaluhr“

Zu entwerfen, durch Simulation zu validieren und im Labor physikalisch zu realisieren ist eine Digitaluhr mit Stunden-, Minuten- und Sekundenanzeige gemäß Bild-1.

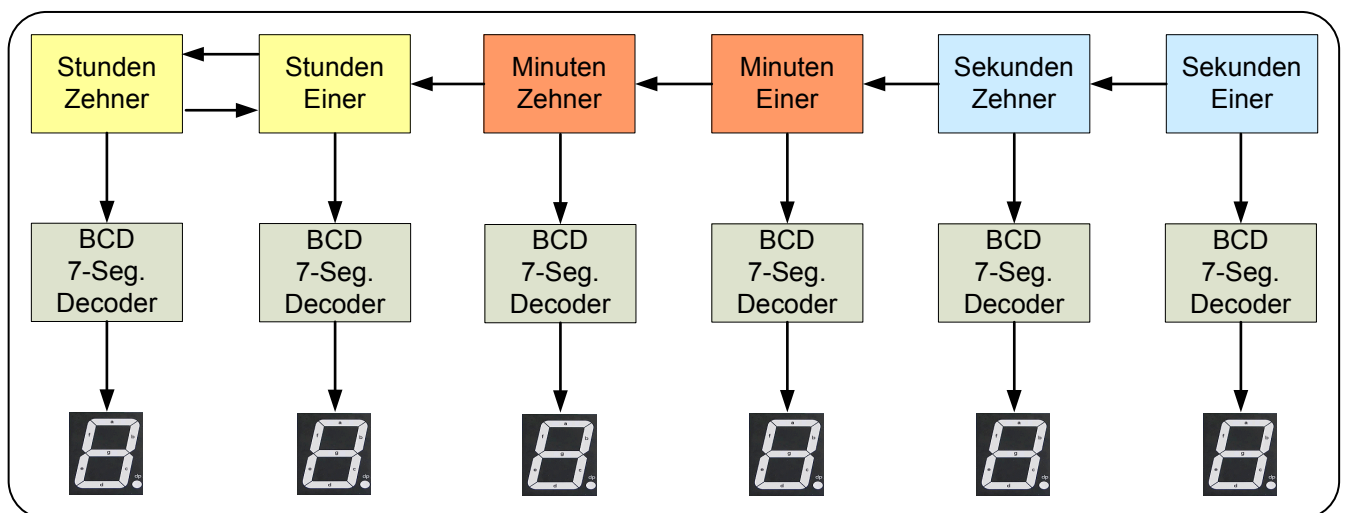


Bild-1

Die Ziffern werden im BCD-Code getrennt verwaltet, so dass ein modularer Aufbau möglich ist. Die Darstellung der einzelnen Ziffern erfolgt in 7-Segment-Anzeigen, nachdem das jeweilige BCD-Codewort in den 7-Segment-Code umgewandelt wurde. Die BCD-zu-7-Segment-Decoder stehen im Labor als fertige Bausteine zur Verfügung.

Zu beachten ist, dass der Wertebereich der Stunden-Einer-Stelle vom aktuellen Wert der Stunden-Zehner-Stelle abhängt.

Die Uhr wird mit einem Taktsignal der Frequenz 1 Hz getaktet. Zur Speicherung der aktuellen Uhrzeit werden D-Flip-Flops verwendet, die als fertige Bausteine zur Verfügung stehen. Der prinzipielle Aufbau des Blocks einer Stelle ist im folgenden Bild-2 dargestellt.

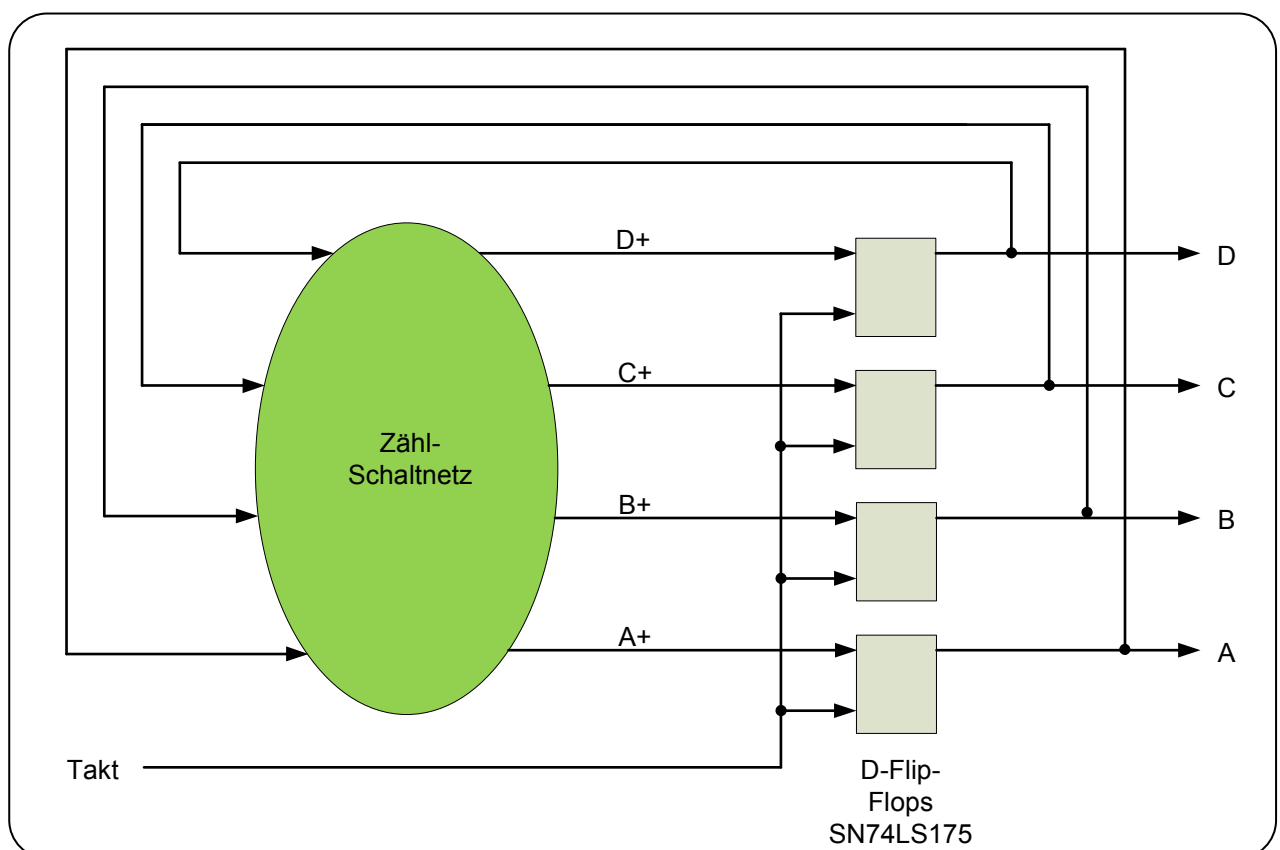


Bild-2



Der Entwurf beschränkt sich auf die Zählschaltnetze gemäß Bild-2.

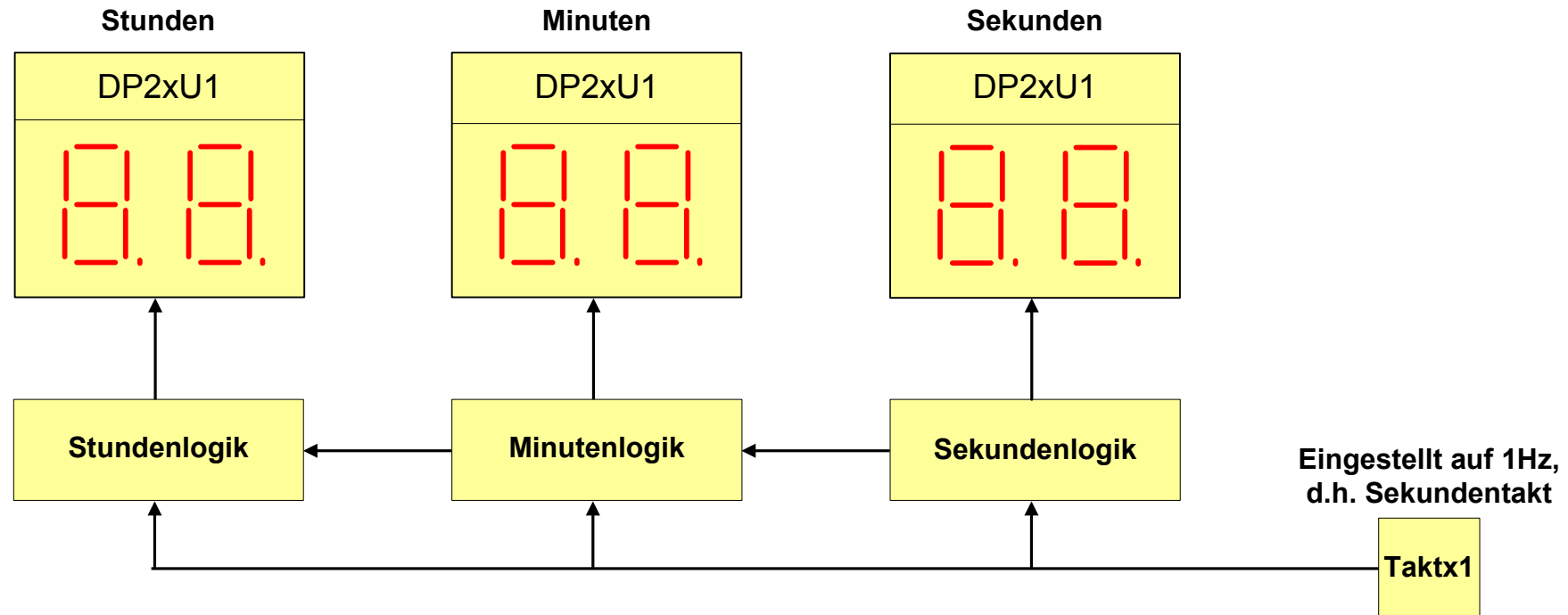
Vorgehensweise:

1. Erstellen Sie je eine Wahrheitstabelle für jeden der zu entwerfenden Blöcke. Für identische Blöcke muss nur eine Tabelle erstellt werden.
2. Übertragen Sie die Funktionswerte aus den Wahrheitstabellen in KV-Diagramme und ermitteln Sie die Schaltfunktionen der Ausgangsvariablen.
3. Validieren Sie Ihren Entwurf zunächst blockweise, anschließend auch die Gesamtschaltung in einem Digital-Simulator.
4. Realisieren Sie die Digitaluhr im Labor.

Hinweis : Verwenden Sie zur Simulation einen Simulator ihrer Wahl, z. B. den kostenlos erhältlichen Simulator „Logisim“. Für die D-FF's sollten Sie Bausteine vom Typ SN74LS175 verwenden.

Digitaluhr mit 24-Stundenanzeige

(Realisiert mit synchronen Zählern für Stunden, Minuten und Sekunden)



DP2xU1 = Displayplatine
IC2x16 = IC-Bausteinplatine
Taktx1 = Taktgeberplatine

Technische Informatik

Praktische Ausarbeitung Digitaluhr

Dipl. Inf. (FH) Stefan Konrath

10.09.2018



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

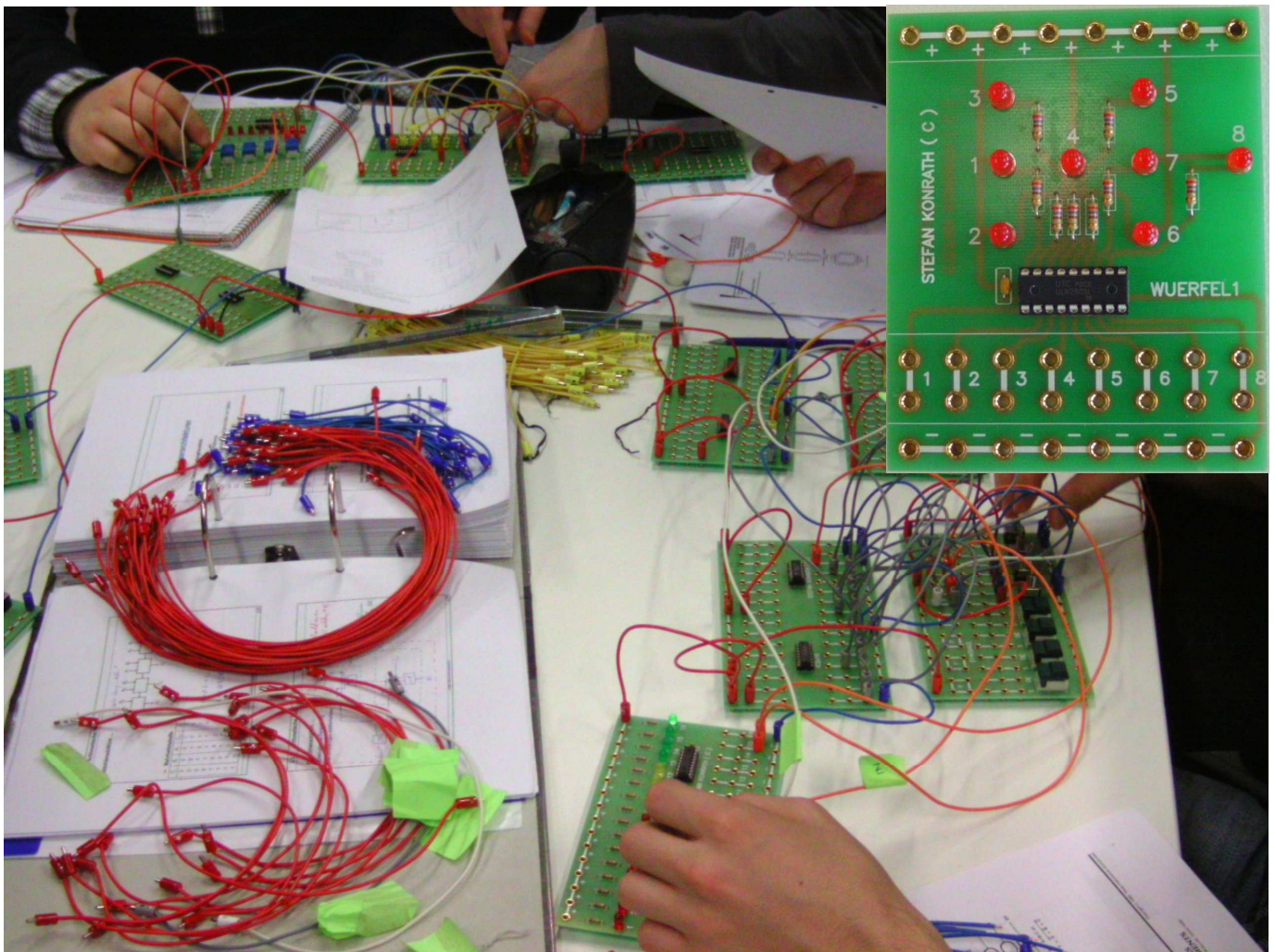
Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Elektronischer Würfel
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

<http://www.hs-kl.de/~stefan.konrath>

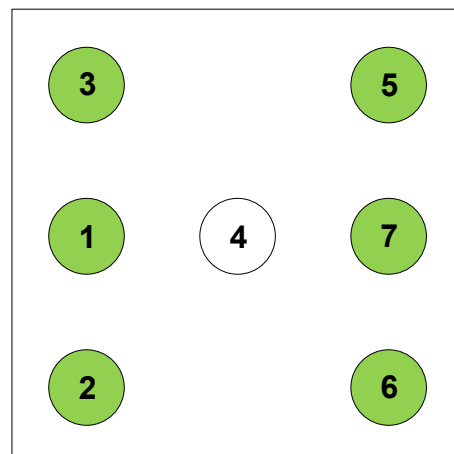
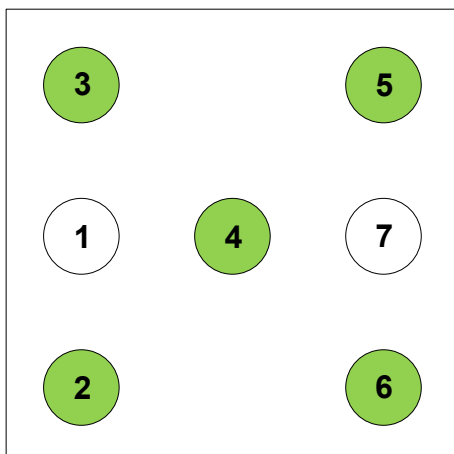
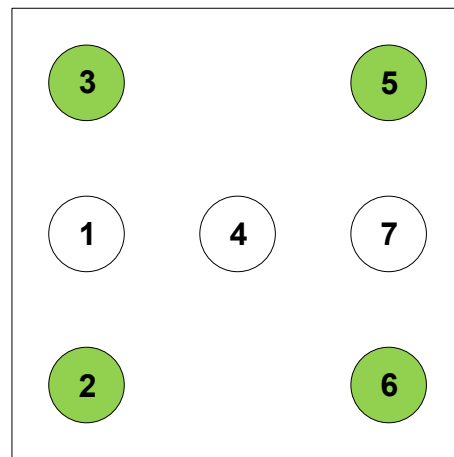
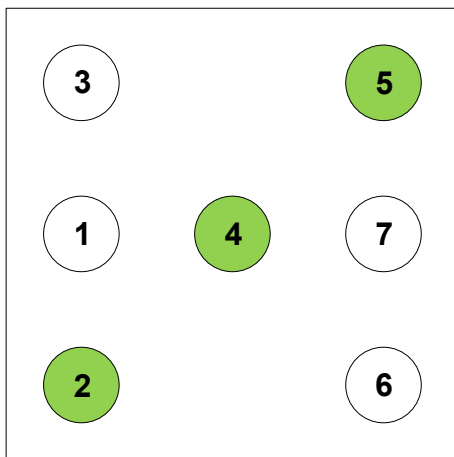
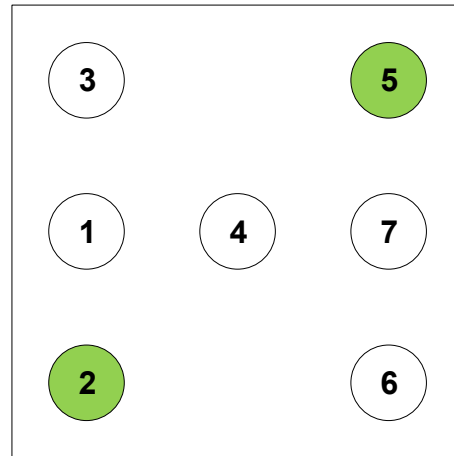
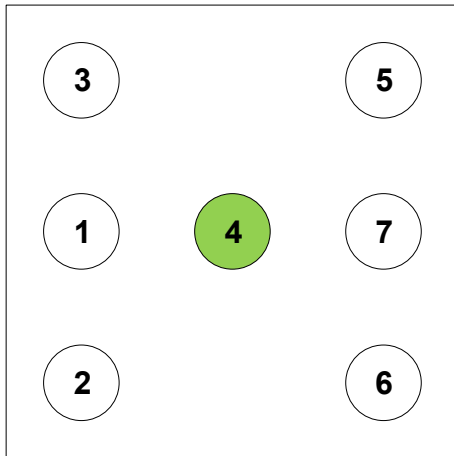
Laborversuch „Elektronischer Würfel“





Realisierung eines elektronischen Würfels

Zu realisierende Anzeigeformen



Übung zur Lehrveranstaltung „Technische Informatik“



**Informatik und
Mikrosystemtechnik
Zweibrücken**

NetLab : Raum O227

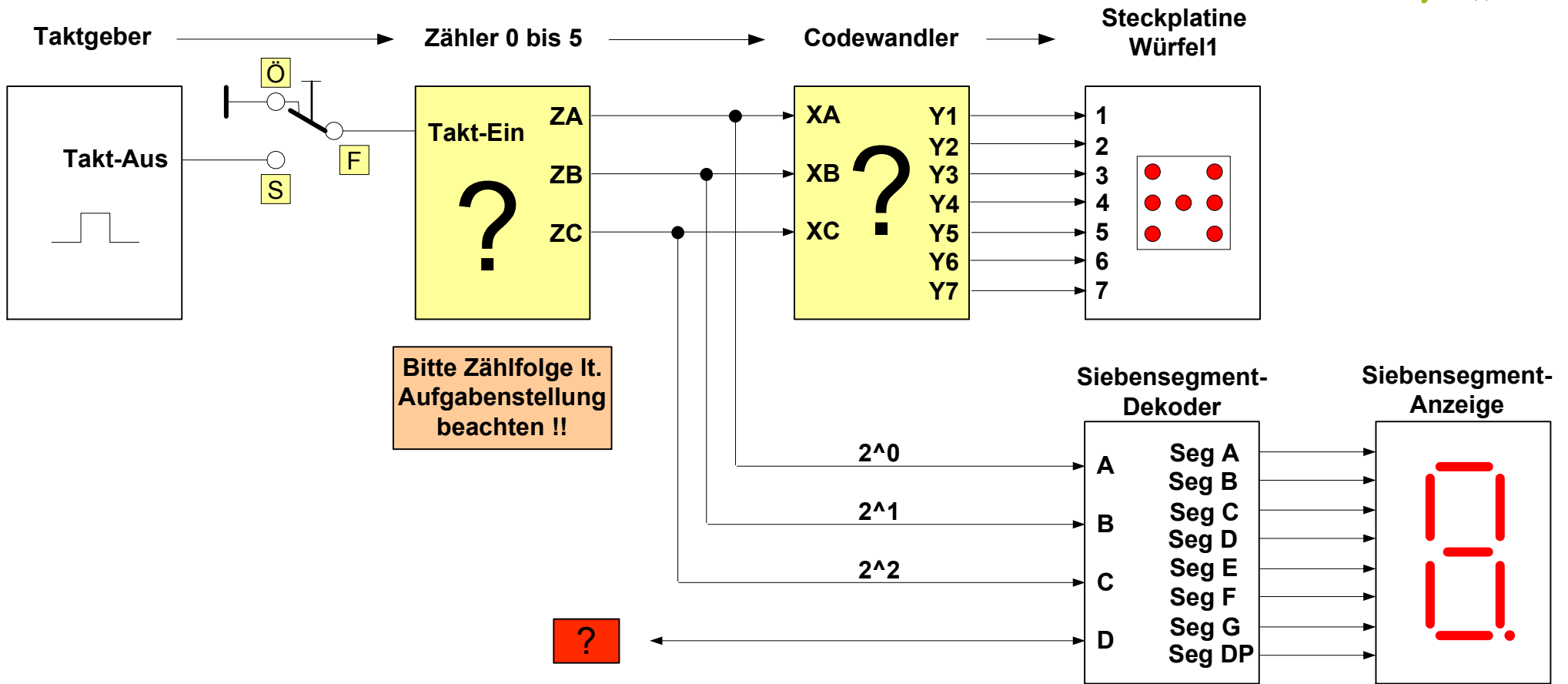
Die Ziffern in den o.a. Leuchtdioden entsprechen den Anschlußbezeichnungen auf der Labor-Steckplatine „Würfel1“.

Realisieren Sie einen synchronen Zähler, aufgebaut mit **D-Flip Flops (SN74LS175)**, mit der sich wiederholenden Zählfolge von 0 bis 5 sowie den nachgeschalteten Codewandler zur Ansteuerung der Leuchtdioden, die wie die Augen eines Würfels angeordnet sind und so angezeigt werden.

Dezimal	ZC	ZB	ZA	LED 1	LED 2	LED 3	LED 4	LED 5	LED 6	LED 7
0										
1										
2										
3										
4										
5										
6										
7										

Tabelle : Zuordnung Zählerstand zu Würfelergebnisanzeige über die LED-Anzeige, ZC, ZB und ZA entsprechen den Ausgängen des synchronen Zählers.

Blockschaltbild „Elektronischer Würfel“



Hinweis: Alle ? hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !

10.09.2018

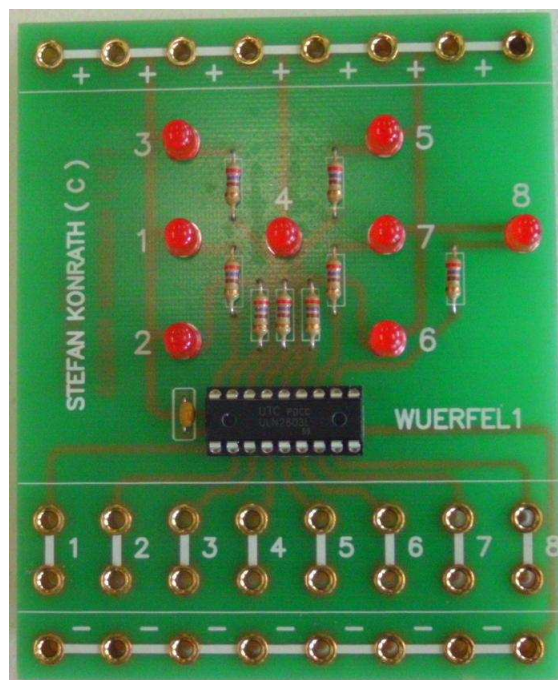
Technische Informatik

Blockschaltbild „Würfel“

Dipl. Inf. (FH) Stefan Konrath

Elektronischer Würfel (WUERFEL1)

+5V Spannungsversorgung



LED- Ansteuerung

Gnd (Minus)

Erläuterung der Steckfelder

+	Hier wird der Pluspol der +5V-Spannungsversorgung angeschlossen.
-	Hier wird der Minuspol der +5V-Spannungsversorgung angeschlossen.
LED-Ansteuerung 1...8	Bei Anlegen einer logischen „1“ leuchtet die entsprechende LED auf.



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Kreuzung Variante 1
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der Signallampen sind nur NOT, OR, AND, NOR, NAND und XOR - Gatter zugelassen.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Kreuzung Variante 2
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der Signallampen verwenden Sie u.a. Bausteine vom Typ „SN74LS138“ oder „SN74LS139“.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Kreuzung Variante 3
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Bausteine des Typs „SN74LS161“. Für die Schaltnetze zur Ansteuerung der Signallampen sind nur NOT, OR, AND, NOR, NAND und XOR - Gatter zugelassen.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborversuch

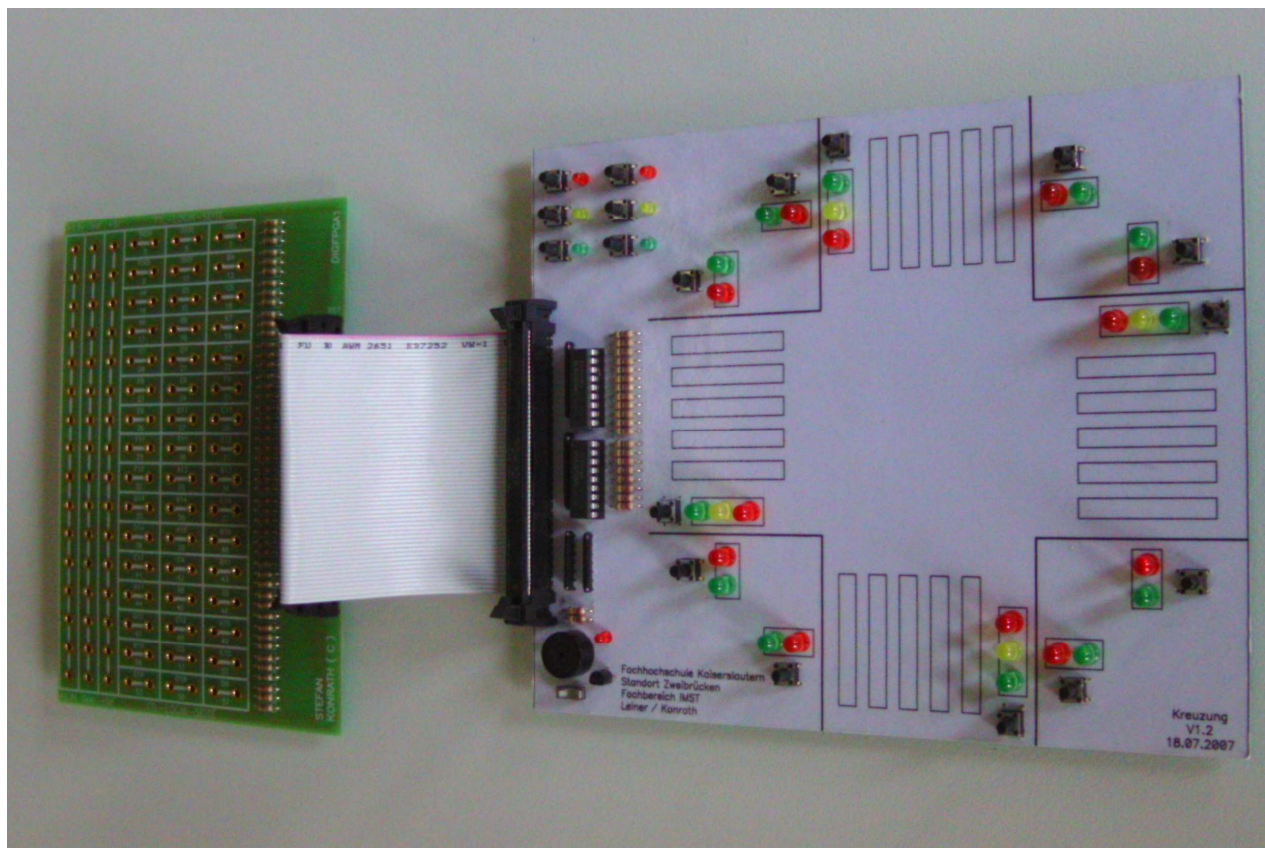
Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Kreuzung Variante 4
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Bausteine des Typs „SN74LS161“. Für die Schaltnetze zur Ansteuerung der Signallampen verwenden Sie u.a. Bausteine vom Typ „SN74LS138“ oder „SN74LS139“.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

<http://www.hs-kl.de/~stefan.konrath>

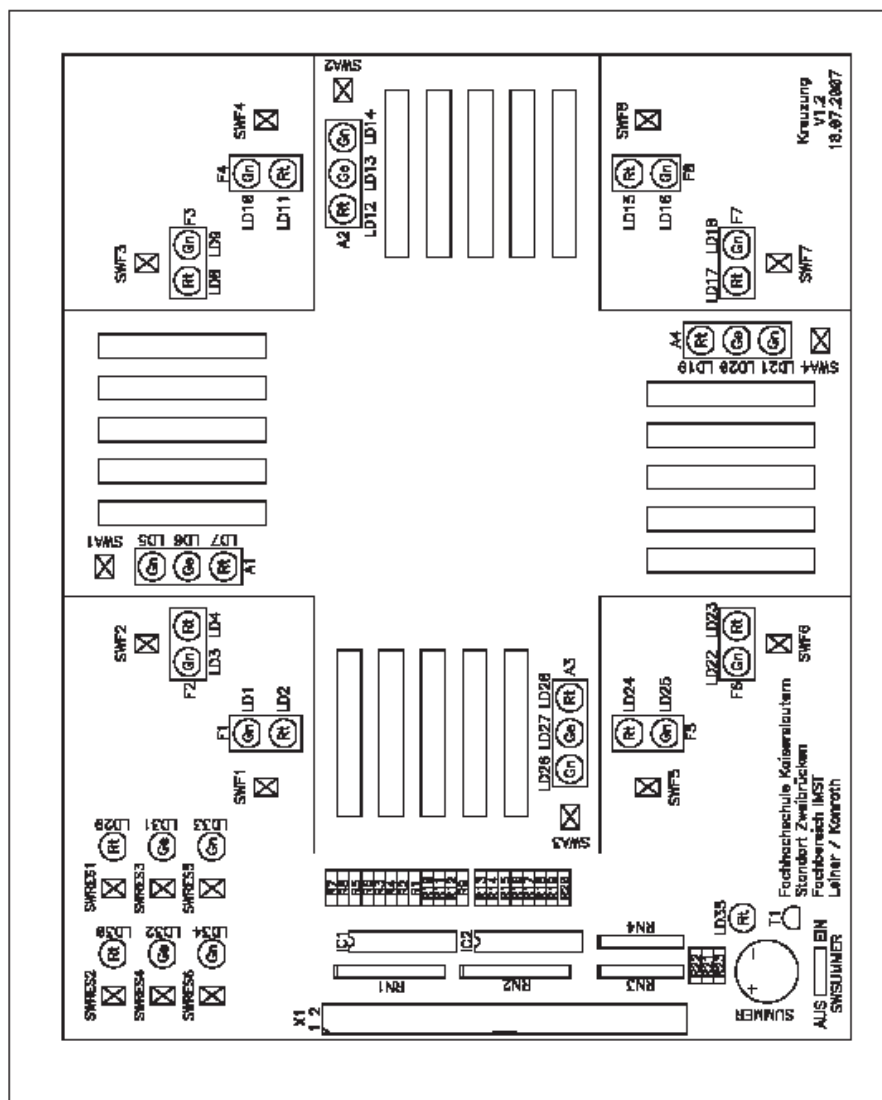
Laborversuch „Kreuzung“



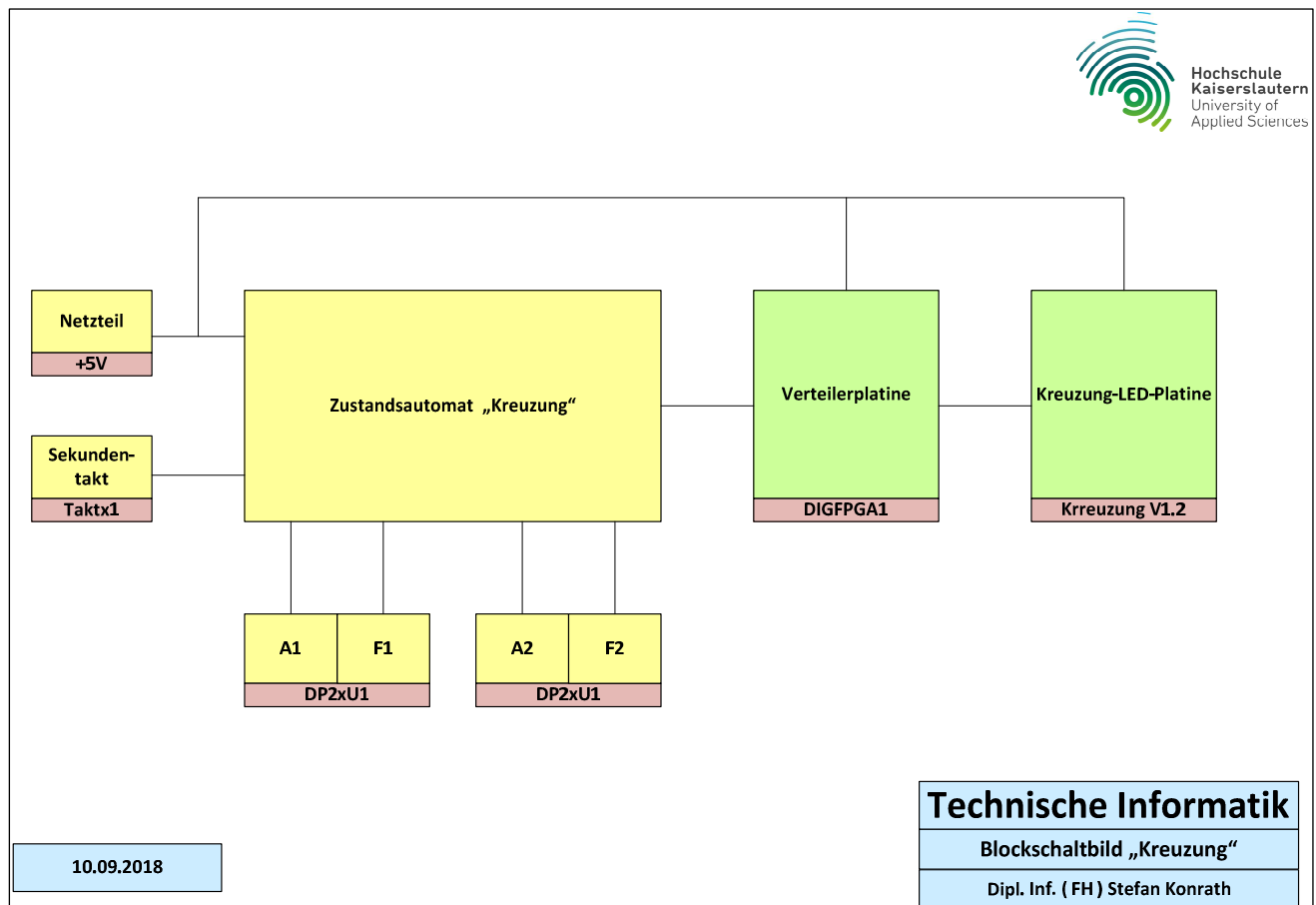
Realisierung einer digital gesteuerten Ampelanlage

Aufgabenstellung ist hier die Realisierung der digitalen Steuerung einer elektronischen Ampelanlage für eine typische Straßenkreuzung.

Auf der u.a. Abbildung ist eine solche Kreuzung schematisch aufgeführt. Hier finden sich auch die Kurzbezeichnungen der entsprechenden Signalleuchten.



Die komplette Aufgabenstellung ist im Blockschaltbild auf dieser Seite prinzipiell aufgeführt.
Es dient als Grundlage für Ihre Applikation.



Zur Realisierung Ihrer Applikation verwenden Sie bitte nur die digitalen Bausteine, welche in der unten aufgeführten Liste „Verfügbare Digital-Bausteine“ aufgelistet sind.

Hinweis : Einschränkungen zur Bausteinauswahl siehe Deckblatt !

Der Zustandszähler des Zustandsautomaten soll in **synchroner Bauweise** realisiert werden !

Verfügbare Digital-Bausteine							
Bezeichnung	Funktion		Bezeichnung	Funktion		Bezeichnung	Funktion
SN74LS00	4 x NAND mit je 2 Eingängen		SN74LS32	4 x OR mit je 2 Eingängen		SN74LS139	2 Bit Bin.Dekoder, DEMUX 2 zu 4
SN74LS02	4 x NOR mit je 2 Eingängen		SN74LS51	AND / NOR - Kombination		SN74L151	8 zu 1 Multiplexer
SN74LS04	6 x Inverter		SN74LS55	AND / NOR - Kombination		SN74L161	Synchr.prog. 4-Bit Bin.Zähler,Clr
SN74LS08	4 x AND mit je 2 Eingängen		SN74LS74	Dual-D-Flip-Flop		SN74LS175	Quad, synchr. D-Flip-Flop
SN74LS10	3 x NAND mit je 3 Eingängen		SN74LS85	4 Bit Vergleicher		SN74LS260	2 x NOR mit je 5 Eingängen
SN74LS11	3 x AND mit je 3 Eingängen		SN74LS86	4 x EXOR mit je 2 Eingängen		SN74LS393	Asynchr. 4 Bit Binärzähler
SN74LS14	6 x Inverter, Schmitt-Trigger		SN74LS125	4-Bus-Leitungs-treiber, tristate		SN74LS573	8-Bit Latch
SN74LS20	2 x NAND mit je 4 Eingängen		SN74LS126	4-Bus-Leitungs-treiber, tristate		GAL16V8	Siebensegment-Dekoder
SN74LS21	2 x AND mit je 4 Eingängen		SN74LS133	1 x NAND mit je 13 Eingängen			
SN74LS27	3 x NOR mit je 3 Eingängen		SN74LS136	4 x EXOR mit je 2 Eingängen			
SN74LS30	1 x NAND mit je 8 Eingängen		SN74LS138	3 Bit Bin.Dekoder, DEMUX 3 zu 8			

Übung zur Lehrveranstaltung „Technische Informatik“




Informatik und Mikrosystemtechnik Zweibrücken

NetLab : Raum O227

In der Tabelle „Zustandsfolge der Ampelschaltung mit TTL-Logikbausteinen“ ist die geforderte Ablauffolge der Ansteuerung der entsprechenden Signalleuchten dargelegt. Die Taktfrequenz der Zustandsabfolge beträgt 1Hz. Dieser Takt wird von der Taktgeberplatine „Taktx1“ geliefert. Hierzu ist dort die entsprechende Frequenz am Drehschalter einzustellen. Die detaillierte Beschreibung der Taktgeberplatine „Taktx1“ sowie der Siebensegmentanzeigeplatine „DP2xU1“ finden Sie im Anhang dieses Dokumentes.

Zustandsfolge der Ampelschaltung mit TTL-Logikbausteinen														
Wert	A1	2^0	2^1	2^2	F1	2^0	2^1	A2	2^0	2^1	2^2	F2	2^0	2^1
Schritt		A1-Rot	A1-Gelb	A1-Grün		F1-Rot	F1-Grün		A2-Rot	A2-Gelb	A2-Grün		F2-Rot	F2-Grün
0	1	X			1	X		1	X			1	X	
1	3	X	X		1	X		1	X			2		X
2	4			X	1	X		1	X			2		X
3	4			X	1	X		1	X			2		X
4	4			X	1	X		1	X			2		X
5	4			X	1	X		1	X			2		X
6	4			X	1	X		1	X			2		X
7	4			X	1	X		1	X			2		X
8	4			X	1	X		1	X			2		X
9	4			X	1	X		1	X			2		X
10	4			X	1	X		1	X			2		X
11	4			X	1	X		1	X			2		X
12	4			X	1	X		1	X			2		X
13	2		X		1	X		1	X			1	X	
14	1	X			1	X		1	X			1	X	
15	1	X			1	X		1	X			1	X	
16	1	X			1	X		1	X			1	X	
17	1	X			2		X	3	X	X		1	X	
18	1	X			2		X	4			X	1	X	
19	1	X			2		X	4			X	1	X	
20	1	X			2		X	4			X	1	X	
21	1	X			2		X	4			X	1	X	
22	1	X			2		X	4			X	1	X	
23	1	X			2		X	4			X	1	X	
24	1	X			2		X	4			X	1	X	
25	1	X			2		X	4			X	1	X	
26	1	X			2		X	4			X	1	X	
27	1	X			2		X	4			X	1	X	
28	1	X			2		X	4			X	1	X	
29	1	X			1	X		2		X		1	X	
30	1	X			1	X		1	X			1	X	
31	1	X			1	X		1	X			1	X	

Die Verknüpfung zwischen den Komponenten der Kreuzungsplatine und Ihrer Applikation erfolgt über die Verteilerplatine „DIGFPGA“. In der nachfolgenden Tabelle ist die Zuordnung zwischen diesen Komponenten aufgeführt.

		Informatik und Mikrosystemtechnik Zweibrücken			
<i>Kreuzung</i>					
Pin-Nr. 60-pol. Verbinder	Pin- Bezeichnung	Funktion (von der Applikation aus gesehen)	Ziel auf der Kreuzungsplatine V1.2		Bemerkung
1	VCCO (+3.3V)	+3.3V Spg-Versorgung	+3.3V Spg-Versorgung		
2	VCCO (+3.3V)	+3.3V Spg-Versorgung	+3.3V Spg-Versorgung		
3	nc	nc	nc		
4	Gnd	Gnd	Gnd		
5	nc	nc	nc		
6	Gnd	Gnd	Gnd		
7	nc	nc	nc		
8	Gnd	Gnd	Gnd		
9	nc	nc	nc		
10	Gnd	Gnd	Gnd		
11	nc	nc	nc		
12	Gnd	Gnd	Gnd		
13	B4	Ausgang	F2_F3_F6_F7_grün		high active
14	A4	Ausgang	F2_F3_F6_F7_rot		high active
15	D5	Ausgang	F1_F5_F4_F8_grün		high active
16	C5	Ausgang	F1_F5_F4_F8_rot		high active
17	A6	Ausgang	A1_A4_rot		high active
18	B6	Ausgang	A1_A4_gelb		high active
19	E7	Ausgang	A2_A3_rot		high active
20	F7	Ausgang	A1_A4_grün		high active
21	D7	Ausgang	A2_A3_gelb		high active
22	C7	Ausgang	A2_A3_grün		high active
23	F8	nc	nc		
24	E8	nc	nc		

Technische Informatik

Kreuzung


Dipl. Inf. (FH) Stefan Konrath

Übung zur Lehrveranstaltung „Technische Informatik“



Informatik und Mikrosystemtechnik Zweibrücken

NetLab : Raum O227

 Hochschule Kaiserslautern University of Applied Sciences		Informatik und Mikrosystemtechnik Zweibrücken			
<i>Kreuzung</i>					
Pin-Nr. 60-pol. Verbinder	FPGA-Pin- Bezeichnung	Funktion (von der Applikation aus gesehen)	Ziel auf der Kreuzungsplatine V1.2		Bemerkung
25	F9	nc	nc		
26	E9	nc	nc		
27	D11	nc	nc		
28	C11	nc	nc		
29	F11	nc	nc		
30	E11	nc	nc		
31	E12	nc	nc		
32	F12	nc	nc		
33	A13	Ausgang	Reserve-1		high active
34	B13	Ausgang	Reserve-2		high active
35	A14	Ausgang	Reserve-4		high active
36	B14	Ausgang	Reserve-3		high active
37	C14	Ausgang	Reserve-6		high active
38	D14	Ausgang	Reserve-5		high active
39	A16	Eingang	SWF1_SWF5_SWF4_SWF8		low active
40	B16	Eingang	SWF2_SWF3_SWF6_SWF7		low active
41	E13	Eingang	SWA1		low active
42	C4	Eingang	SWA4		low active
43	B11	Eingang	SWA2		low active
44	A11	Eingang	SWA3		low active
45	A8	Eingang	SWRES1		low active
46	G9	Eingang	SWRES2		low active
47	D12	Eingang	SWRES3		low active
48	C12	Eingang	SWRES4		low active
49	A15	Eingang	SWRES5		low active
50	B15	Eingang	SWRES6		low active
51	C3	Ausgang	SUMMER		high active
52	C15	nc	nc		
53	E10	nc	nc		
54	Gnd	Gnd	Gnd		
55	D10	nc	nc		
56	Gnd	Gnd	Gnd		
57	D9	nc	nc		
58	Gnd	Gnd	Gnd		
59	+5V	+5.0V Spg-Versorgung	+5.0V Spg-Versorgung		
60	+5V	+5.0V Spg-Versorgung	+5.0V Spg-Versorgung		
		Dipl. Inf. Stefan Konrath			



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

RA-Laborbesuch

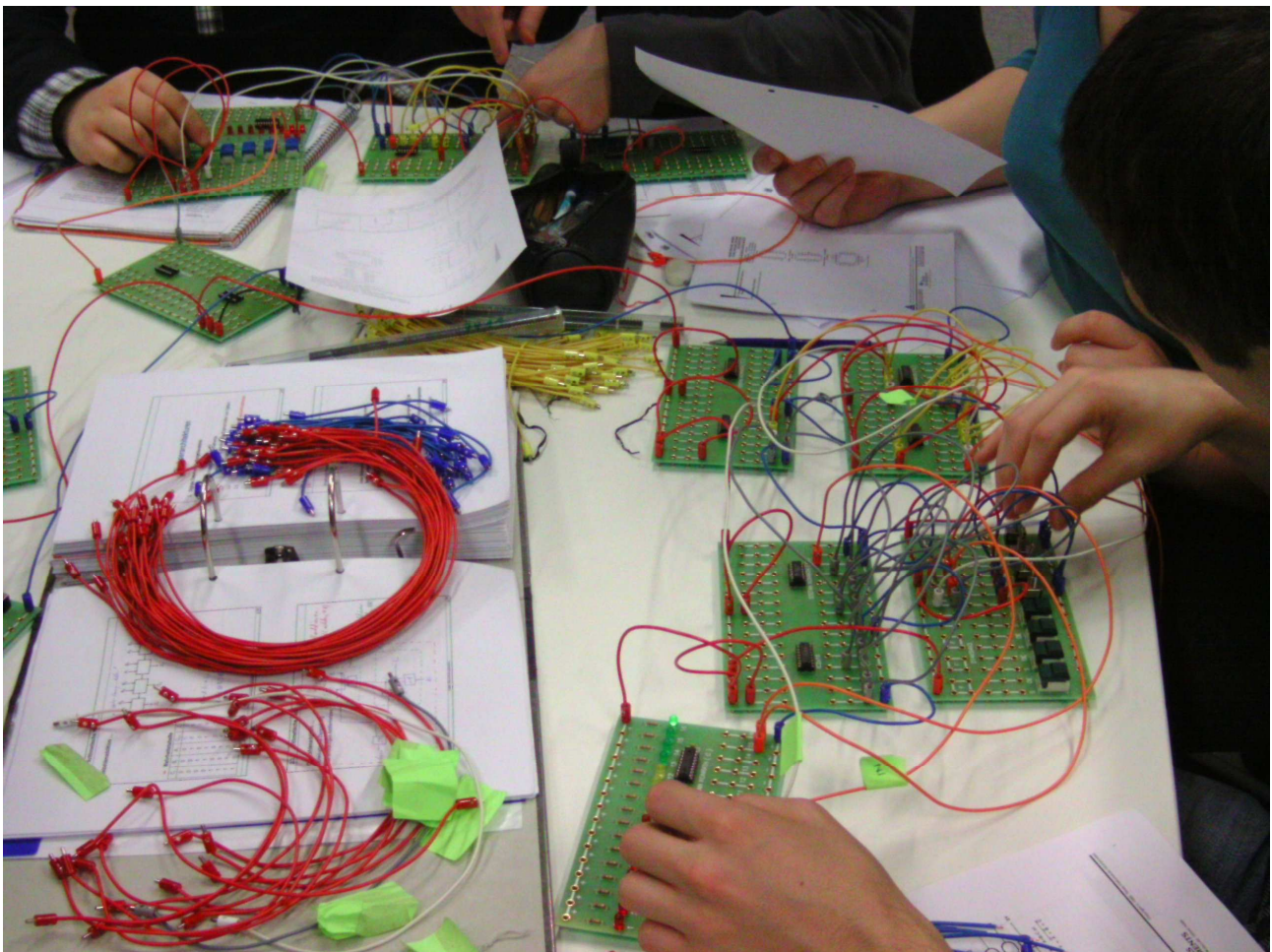
Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	LED - Multiplexer / Demultiplexer
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung der synchronen Zähler nur Flip-Flops des Typs „ SN74LS175 “. Beachten Sie die Zählfolge der Zähler.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

<http://www.hs-kl.de/~stefan.konrath>

Laborversuch „LED - Multiplexer / Demultiplexer“



Aufgabenstellung :

Zu entwerfen und im Labor physikalisch zu realisieren ist eine Multiplexer / Demultiplexer-Einheit zur Übertragung von Zählerständen an eine entfernte LED-Anzeigeeinheit (LED2x8) mit geringem Leitungsaufwand.

Sämtliche Zähler sind in **synchroner Arbeitsweise** und mit **D-Flip Flops vom Typ SN74LS175** aufzubauen.

Vorgehen :

Die Aufgabe gliedert sich in zwei Bereiche. Die synchronen Zähler mit den vorgegebenen Zählfolgen (0...15) und der Multiplexer / Demultiplexer-Einheit. Realisieren Sie zunächst die Zähler und anschließend die Multiplexer / Demultiplexer-Einheit

Visualisieren Sie den aktuellen Zählerstand des 2 x 4 Bit Zählers auch auf der Senderseite über eine LED-Anzeigeplatine (LED2x8) und eine Siebensegment-Anzeigeplatine „DP2xU1“. Auf der Empfängerseite erfolgt die Visualisierung über die LED-Anzeigeplatine „LED2x8“. Die Taktversorgung übernehmen die beiden Platinen „Taktx1“.

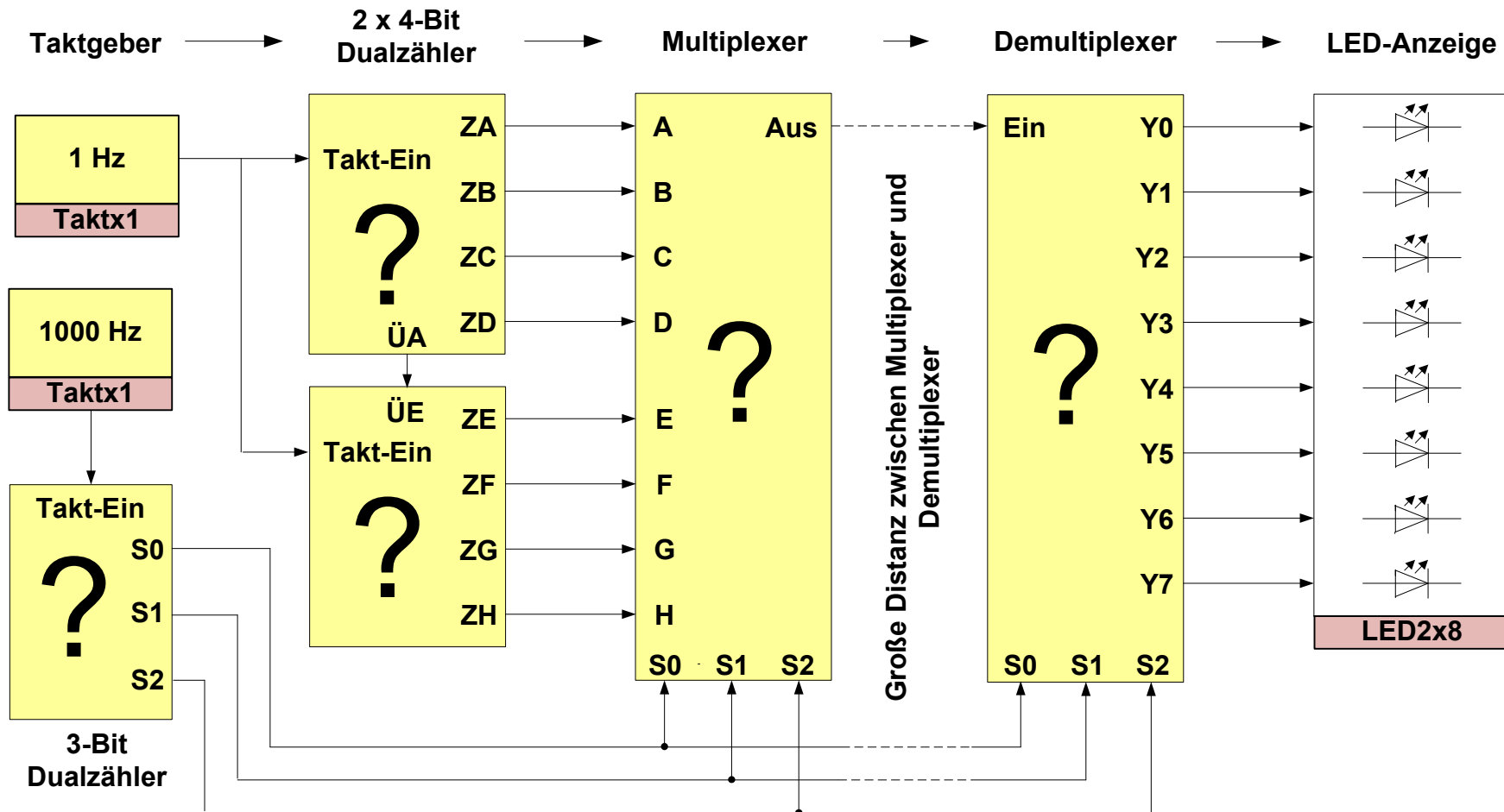
Treffen Sie für Ihr Design unbedingt eine optimale Bausteinauswahl aus der Liste „Verfügbare Digital-Bausteine“.

.

Weitere Details entnehmen Sie bitte dem folgenden Blockschaltbild.

Die Dokumentationen zu den Platinen „DP2xU1“ und „Taktx1“ finden Sie im Anhang dieses Dokumentes.

Multiplexer / Demultiplexer



Hinweis: Alle ? hinterlegten Funktionsblöcke sind von Ihnen zu realisieren!

Technische Informatik

Multiplexer / Demultiplexer

Dipl. Inf. (FH) Stefan Konrath

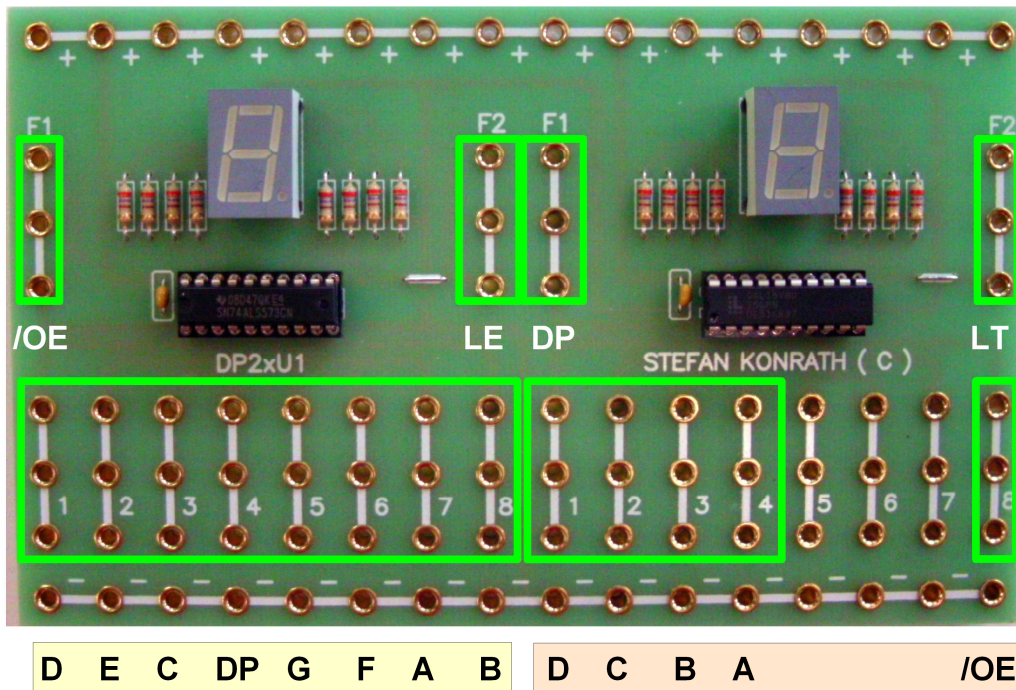
10.09.2018

Verfügbare Digital-Bausteine

Bezeichnung	Funktion		Bezeichnung	Funktion		Bezeichnung	Funktion
SN74LS00	4 x NAND mit je 2 Eingängen		SN74LS32	4 x OR mit je 2 Eingängen		SN74LS139	2 Bit Bin.Dekoder, DEMUX 2 zu 4
SN74LS02	4 x NOR mit je 2 Eingängen		SN74LS51	AND / NOR - Kombination		SN74L151	8 zu 1 Multiplexer
SN74LS04	6 x Inverter		SN74LS55	AND / NOR - Kombination		SN74L161	Synchr.prog. 4-Bit Bin.Zähler,Clr
SN74LS08	4 x AND mit je 2 Eingängen		SN74LS74	Dual-D-Flip-Flop		SN74LS175	Quad, synchr. D-Flip-Flop
SN74LS10	3 x NAND mit je 3 Eingängen		SN74LS85	4 Bit Vergleicher		SN74LS260	2 x NOR mit je 5 Eingängen
SN74LS11	3 x AND mit je 3 Eingängen		SN74LS86	4 x EXOR mit je 2 Eingängen		SN74LS393	Asynchr. 4 Bit Binärzähler
SN74LS14	6 x Inverter, Schmitt-Trigger		SN74LS125	4-Bus-Leitungstreiber, tristate		SN74LS573	8-Bit Latch
SN74LS20	2 x NAND mit je 4 Eingängen		SN74LS126	4-Bus-Leitungstreiber, tristate		GAL16V8	Siebensegment-Dekoder
SN74LS21	2 x AND mit je 4 Eingängen		SN74LS133	1 x NAND mit je 13 Eingängen			
SN74LS27	3 x NOR mit je 3 Eingängen		SN74LS136	4 x EXOR mit je 2 Eingängen			
SN74LS30	1 x NAND mit je 8 Eingängen		SN74LS138	3 Bit Bin.Dekoder, DEMUX 3 zu 8			

Siebensegmentanzeige (DP2xU1)

+5V Spannungsversorgung



Gnd (Minus)

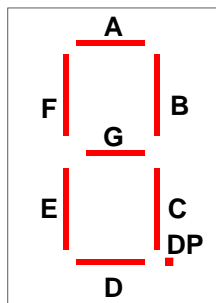
Bei dieser Steckplatine handelt es sich um eine universelle Anzeigeplatine. Je nach Bestückung der beiden IC's stehen verschiedene Funktionalitäten (4 Varianten) zur Verfügung.

Links bestückt mit SN74ALS573	Rechts bestückt mit SN74ALS573
Links bestückt mit SN74ALS573 (Standard)	Rechts bestückt mit GAL16V8D (Standard)
Links bestückt mit GAL16V8D	Rechts bestückt mit SN74ALS573
Links bestückt mit GAL16V8D	Rechts bestückt mit GAL16V8D

Erläuterung der Steckfelder

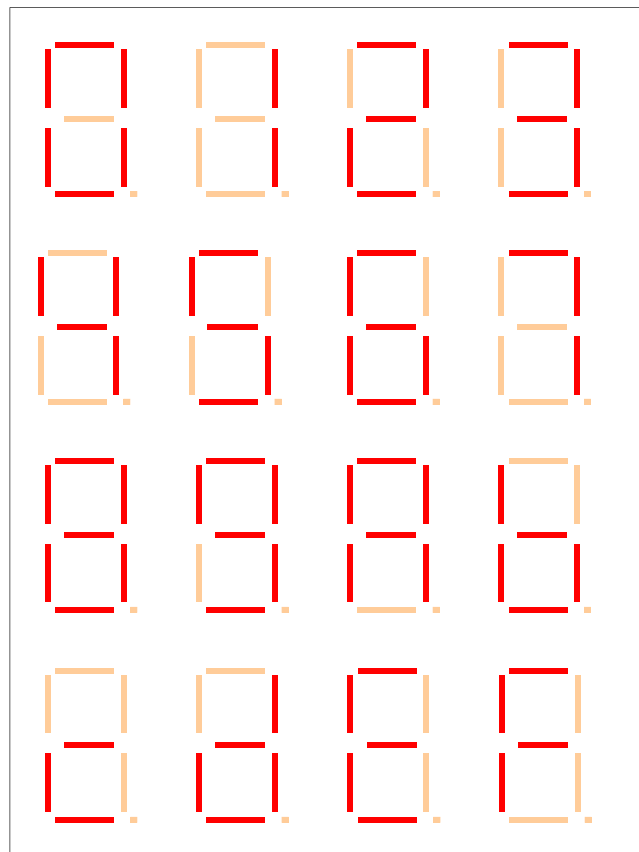
Bestückt mit SN74ALS573	Dieser Baustein ist ein sog. Latch mit der Funktion als Segmenttreiber. Die diskrete Ansteuerung der Anzeigesegmente geschieht wie auf der linken Seite der Platinenabbildung exemplarisch dargestellt über die Segmenteingänge A..DP.
Segmenteingänge A..DP	Logisch-„1“- Pegel lässt das entsprechende Anzeigesegment aufleuchten.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Angesteuerte Segmente leuchten nun auf, ansonsten bleibt die Anzeige komplett dunkel !
LE	Latch-Enable-Eingang. Bei logisch-„1“- Pegel werden die Informationen an den Segmenteingängen A..DP direkt zur Anzeige weitergegeben. Eine Änderung an den Eingängen wird sofort angezeigt. Bei logisch-„0“- Pegel wird die letzte Information an den Segmenteingängen im Baustein gespeichert und auf der Anzeige dargestellt.

Segmentzuordnung



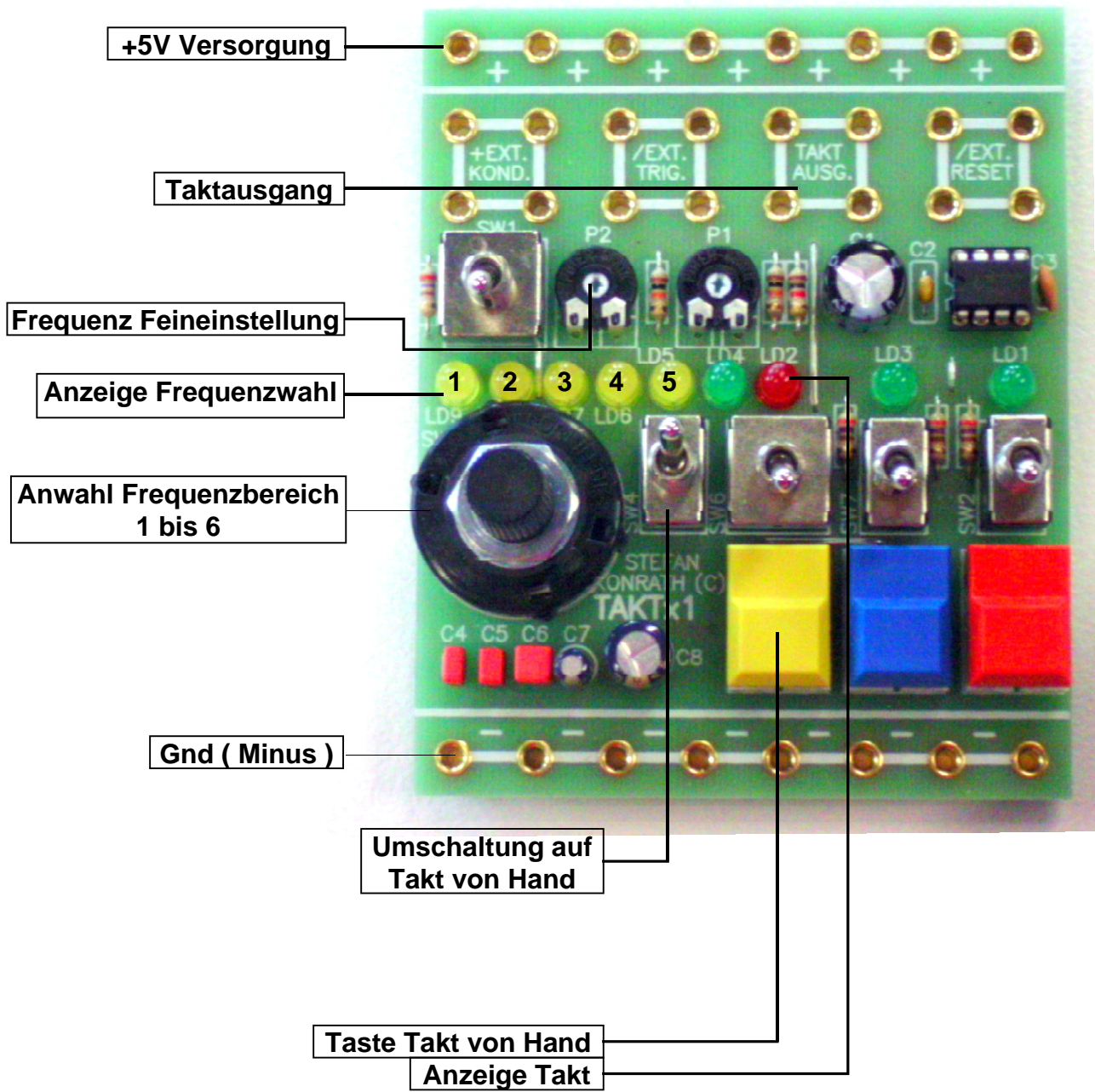
Bestückt mit GAL16V8D	Dieser Baustein ist ein sog. programmierbarer Logikbaustein mit der Funktion als Siebensegment-Hex-Dekoder. Die diskrete Ansteuerung der Anzeigensegmente geschieht hier über ein logisches Schaltnetz im Baustein. Wie auf der rechten Seite der Platinenabbildung exemplarisch dargestellt, erfolgt die Ansteuerung über die 4 Dateneingänge A..D.
Dateneingänge A..D	Je nach Ansteuerung dieser Eingänge wird die entsprechende Hexadezimalzahl auf der Anzeige dargestellt. Dabei ist „A“ der Eingang mit der niedersten Wertigkeit (2^0).
DP Eingang	Dezimalpunkt -Eingang. Bei logisch-„1“- Pegel wird der Dezimalpunkt der Anzeige angesteuert.
LT	Lampen-Test-Eingang. Bei logisch-„1“ - Pegel werden alle Segmente der Anzeige angesteuert und beginnen zu leuchten. Hierbei kann ein eventueller elektrischer oder mechanischer Defekt der Anzeige aufgezeigt werden.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Die anzuzeigende Hexadezimalzahl kann nun aufleuchten.

Hexadezimale Darstellung (Hex. 0..F für dez. 0..15)



Taktgenerator (Bedienelemente)

Für den Laborbetrieb sind die Kippschalter hier in der Standardeinstellung abgebildet.



Erläuterung der Bedienelemente und Steckfelder

<p><u>+5V Versorgung</u> :</p>	<p>8 Steckfelder. Anschluß der +5V Spannungsversorgung. Auf Polung achten !</p>
<p><u>Taktausgang</u> :</p>	<p>4 Steckfelder. Hier liegt der erzeugte Takt zur Weiterverarbeitung an.</p>
<p><u>Frequenz Feineinstellung</u> :</p>	<p>Trimpotentiometer zur Frequenzeinstellung in dem durch den Drehschalter gewählten Frequenzbereich. Siehe hierzu auch „<u>Anzeige Frequenz- Zeitwahl</u>“ .</p> <p style="padding-left: 40px;"> Frequenzbereich 1 : 0,1 Hz bis 1 Hz Frequenzbereich 2 : 1 Hz bis 10 Hz Frequenzbereich 3 : 10 Hz bis 100 Hz Frequenzbereich 4 : 100 Hz bis 1 kHz Frequenzbereich 5 : 1 kHz bis 10 kHz Frequenzbereich 6 : benutzerdefiniert, für „<u>+Ext. Kondensator</u>“ </p> <p>Die linke Potentiometereinstellung entspricht der niedrigsten Frequenz, die rechte Potentiometereinstellung der höchsten Frequenz im gewählten Bereich.</p>
<p><u>Anzeige Frequenz- Zeitwahl</u> :</p>	<p>LED-Anzeige des gewählten Frequenz- bzw. Zeitbereiches.</p>
<p><u>Anwahl Frequenz- bzw. Zeitbereich 1 bis 6</u> :</p> <p>Standard : Frequenzbereich 1</p>	<p>Drehschalter für die gewünschte Einstellung. Siehe hierzu auch „<u>Anzeige Frequenz- Zeitwahl</u>“</p>
<p><u>Gnd (Minus)</u> :</p>	<p>8 Steckfelder. Anschluss „Gnd“ der Spannungsversorgung. Auf Polung achten !</p>
<p><u>Umschaltung auf Takt von Hand</u> :</p> <p>Standard : Untere Kippstellung</p>	<p>Umschaltung zwischen Takt vom Timerbaustein oder manuellem Takt über „<u>Taste Takt von Hand</u>“ .</p> <p style="padding-left: 40px;">Obere Kippstellung : Takterzeugung per Taste „<u>Taste Takt von Hand</u>“</p> <p style="padding-left: 40px;">Untere Kippstellung : Takt vom Timerbaustein.</p>
<p><u>Taste Takt von Hand</u> :</p>	<p>Taste zur Erzeugung des manuellen Taktes. Siehe hierzu auch „<u>Umschaltung Taktquelle</u>“ und „<u>Umschaltung Low / High Aktiv bei Takt von Hand</u>“ .</p>
<p><u>Anzeige Takt</u> :</p>	<p>LED-Anzeige des aktuellen Taktsignals am „<u>Taktausgang</u>“</p>



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	PWM (<u>P</u> uls <u>w</u> eiten <u>m</u> odulation)
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Technische Informatik

Studiengang *Angewandte Informatik*

Praktische Aufgabe

Zu entwerfen und im Labor physikalisch zu realisieren ist eine Schaltung zur Erzeugung von PWM-Signalen (Pulsweitenmodulation) . Diese Signale dienen z.B. zur Steuerung der Heiz- bzw. Lichtleistung von Heizkörpern oder LED-Beleuchtungskörpern. Verwenden Sie zur kompakten Realisierung u.a. den Baustein SN74LS85 und SN74LS175 !

Auf den folgenden 3 Blättern ist beispielhaft die prinzipielle Arbeitsweise der PWM dargelegt. Die Zeitangaben dienen im Beispiel nur zur Veranschaulichung.

Das vorletzte Blatt dient als Grundlage für Ihre Ausarbeitung. Hierbei sind die Spalten unter „Vorgabe“ der Vorgabewert der Helligkeitssteuerung. Am besten lässt sich dieses mit 4 Kippschaltern bewerkstelligen. Jede Zeile entspricht einem bestimmten Helligkeitswert. Bei der ersten Zeile ist der Helligkeitswert „0%“ (Lampe aus), bei der letzten Zeile „100%“ (Lampe volle Helligkeit).

Visualisieren Sie den Vorgabewert der Kippschalter über eine Siebensegment-Anzeigeeinheit und über Leuchtdioden. Die Siebensegment-Anzeigeeinheit dient gleichzeitig als Lampenersatz und wird über Ihre Applikation helligkeitsgesteuert. Dieser Schaltungsteil ist im Blockschaltbild (letztes Blatt) nicht aufgeführt. Beachten Sie ebenfalls die Steuereingänge des Vergleichers und der Siebensegment-Anzeigeeinheit !

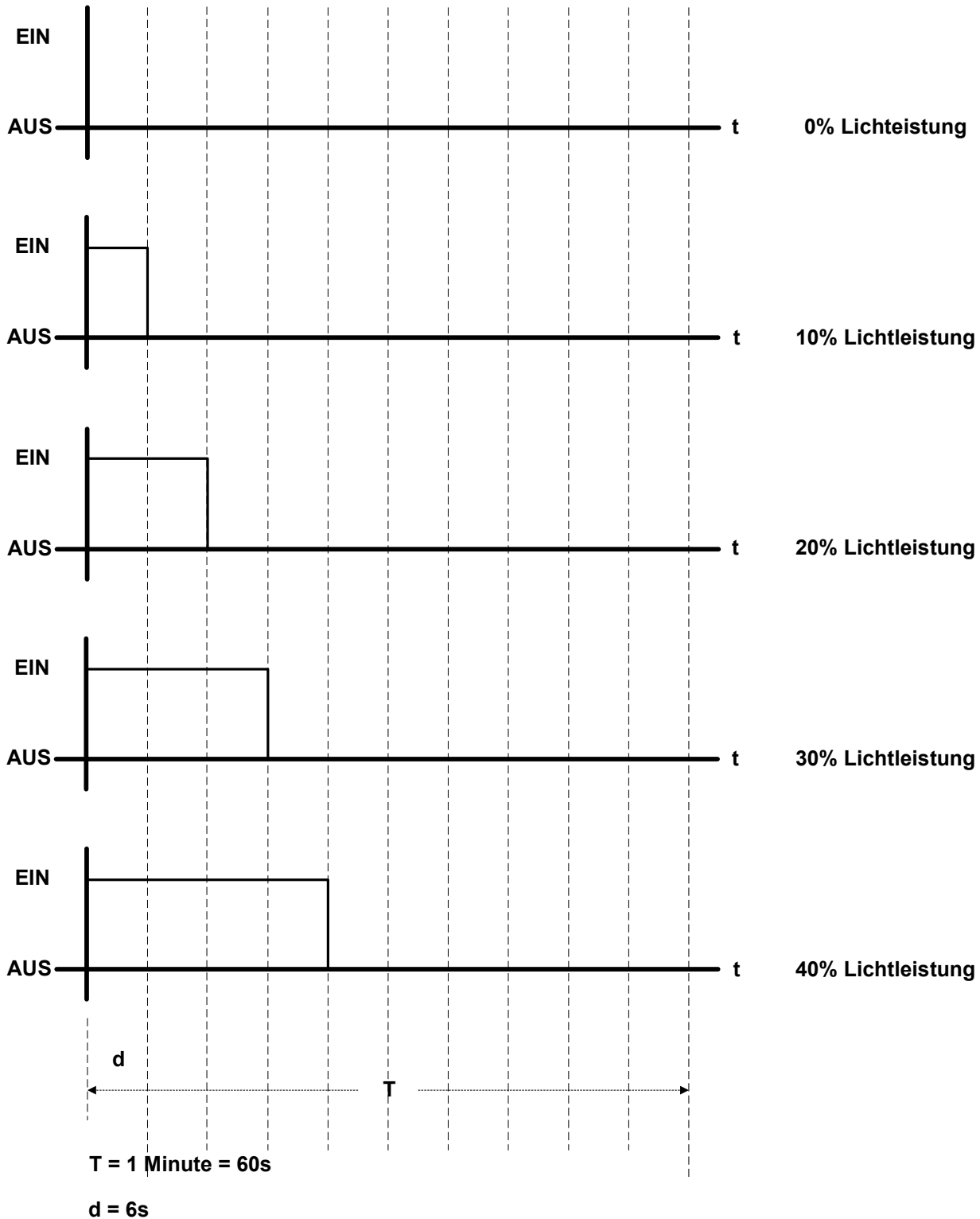
Um eine flackerfreie Helligkeitssteuerung zu realisieren ist mit einer Wiederholfrequenz von 100Hz zu arbeiten, d.h. für die Bearbeitung einer Zeile (eines bestimmten Helligkeitswertes) sind 10ms notwendig. Da hier jede Zeile 15 Zeitspalten (T0 ... T14) aufweist, wird für jede Spalte $10\text{ms} / 15 = 666,66\ \mu\text{s}$ benötigt. Berechnen Sie die notwendige Frequenz des Taktgenerators und stellen Sie die errechnete Frequenz dort ein. Tragen Sie den Frequenzwert ebenfalls in das vorgesehene Feld im Blockschaltbild ein. Überprüfen Sie Ihre Einstellung mit entsprechenden Hilfsgeräten.

Zum Aufbau der Schaltung stehen im Labor ein Siebensegment-Decoder und eine Siebensegment-Anzeige zur Verfügung. Der Siebensegment-Decoder muss folglich nicht entworfen werden.

Im Simulator „Logisim“ verwenden Sie zur Anzeige des Vorgabewertes das Element „Hexadezimale Anzeige“ unter der Rubrik „Ein-/Ausgabe“. Dieser Anzeigebaustein enthält bereits den Siebensegment-Decoder.

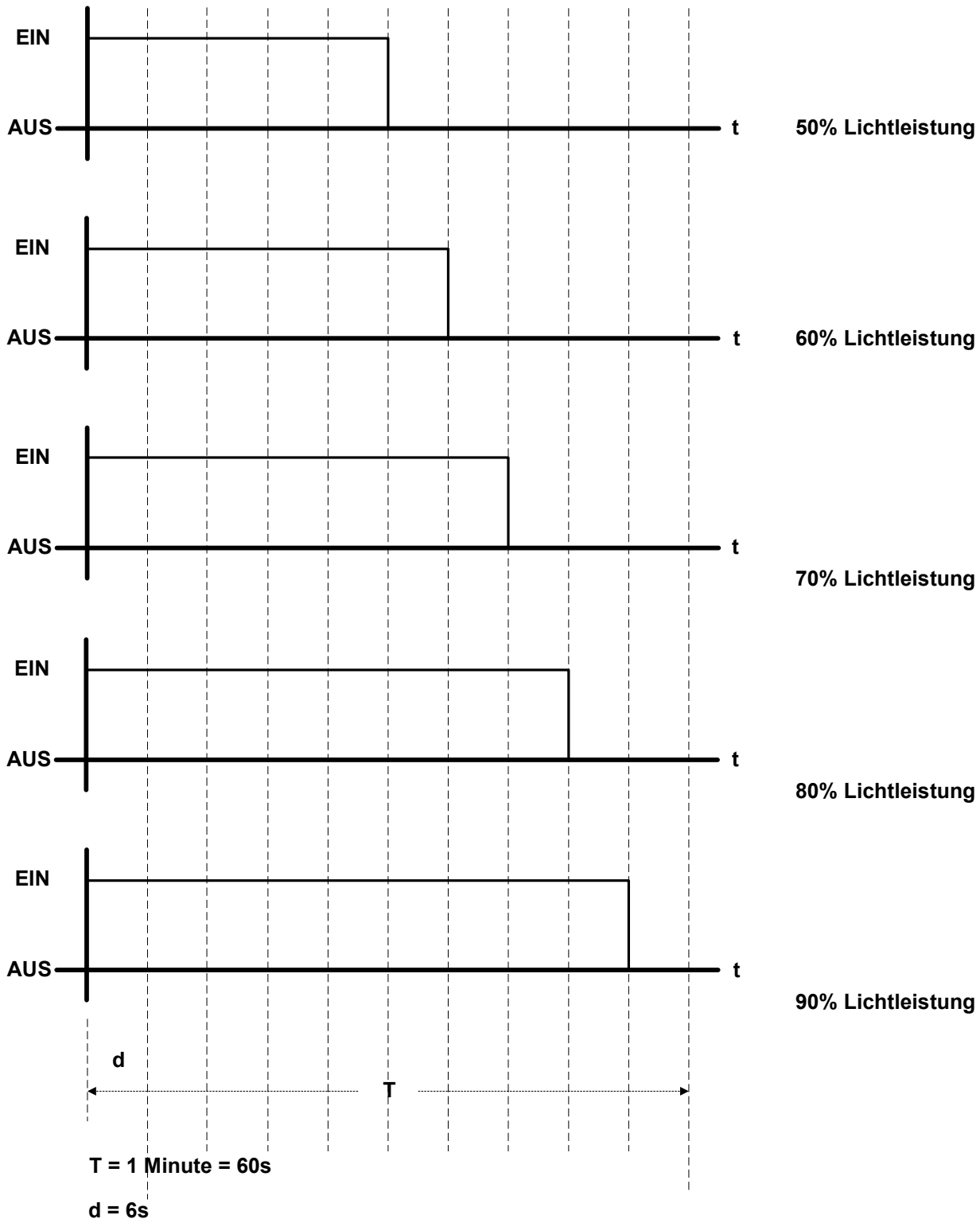


Pulsweitenmodulation (PWM)

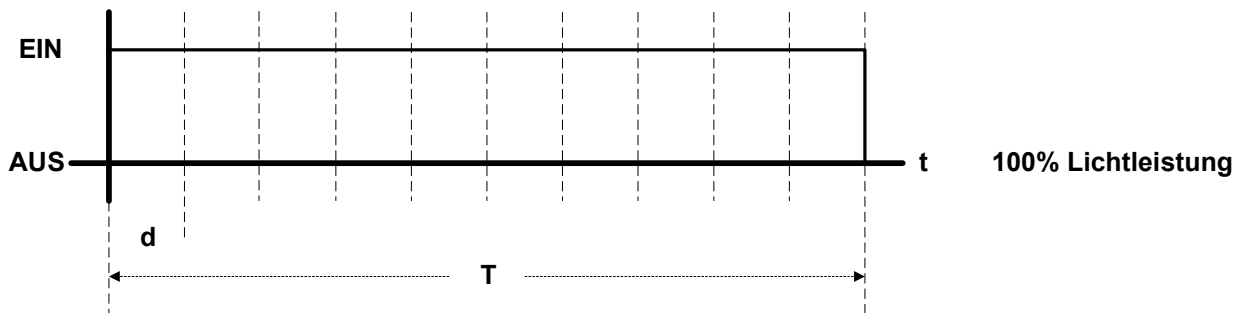




Pulsweitenmodulation (PWM)



Pulsweitenmodulation (PWM)



$T = 1 \text{ Minute} = 60\text{s}$

$d = 6\text{s}$



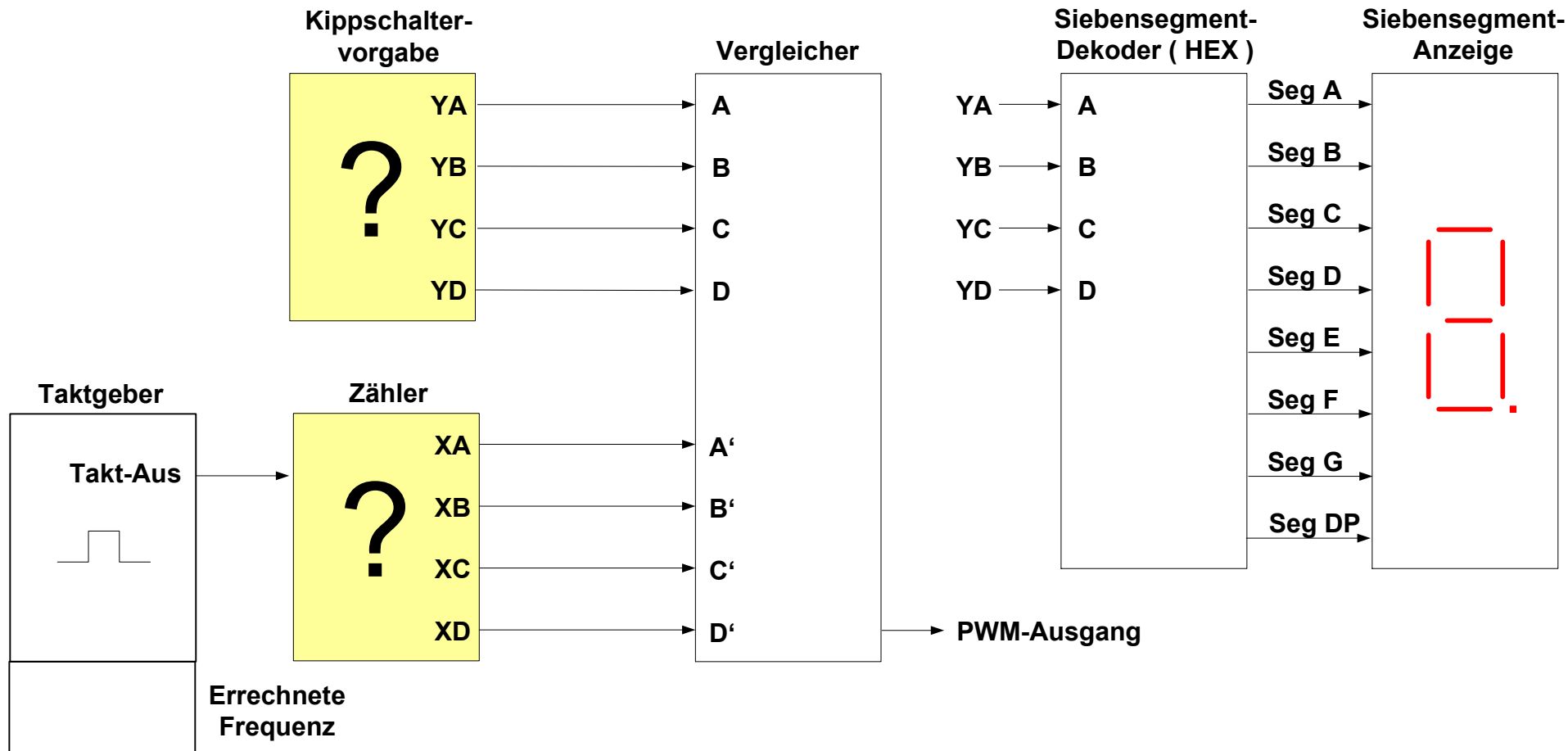
Vorgabe				PWM-Ausgang														
D	C	B	A	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Technische Informatik

PWM-Tabelle

Dipl. Inf. (FH) Stefan Konrath

4-Bit-PWM



10.09.2018

Hinweis : Die mit ? hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	RGB-Würfel Variante 1
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der LED's sind nur NOT, OR, AND, NOR, NAND und XOR - Gatter zugelassen.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	RGB-Würfel Variante 2
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des Zählers nur Flip-Flops des Typs „SN74LS175“. Für die Schaltnetze zur Ansteuerung der LED's verwenden Sie u.a. Bausteine vom Typ „SN74LS139“.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

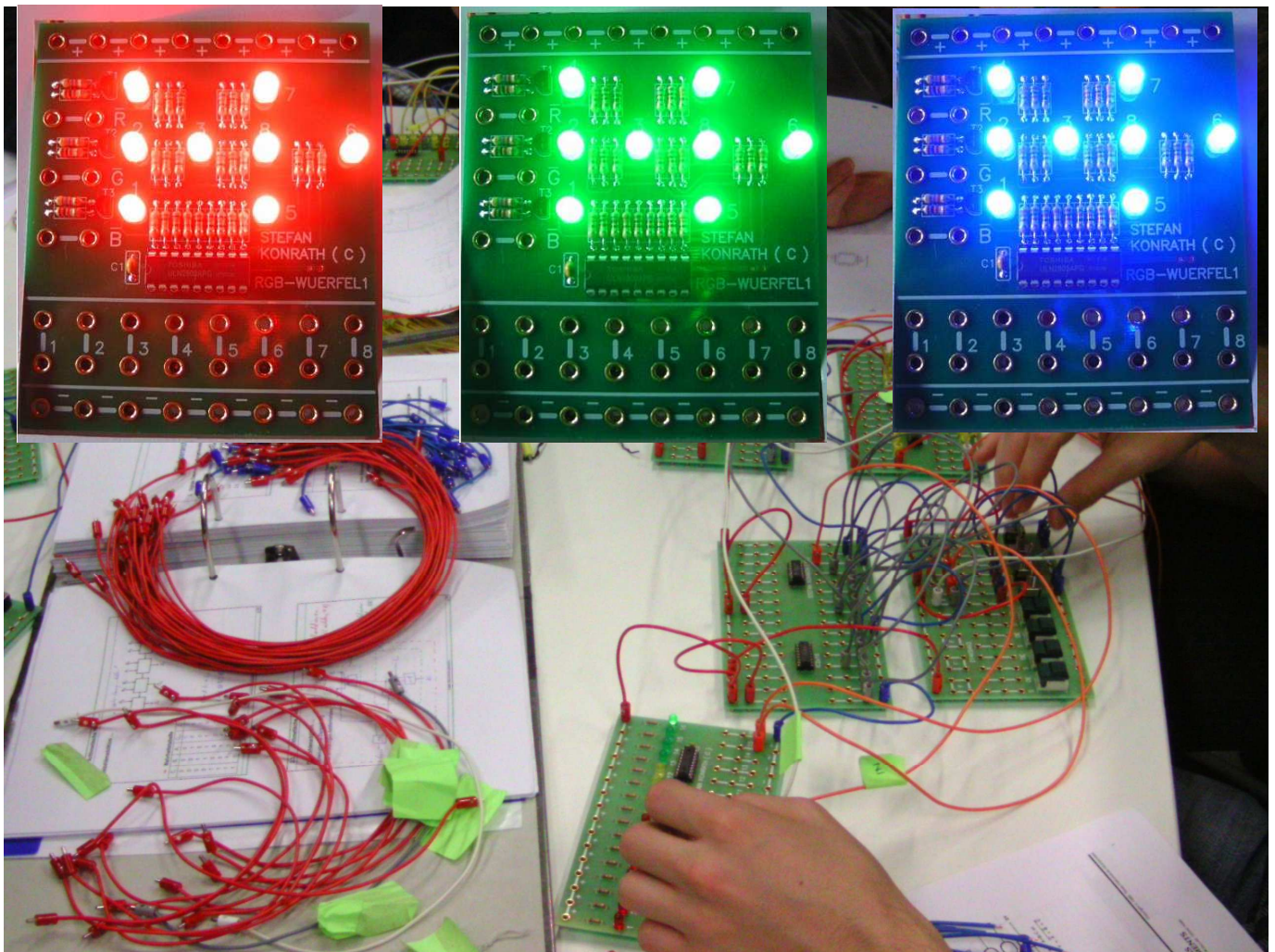


Dipl. Inf. (FH) Stefan Konrath

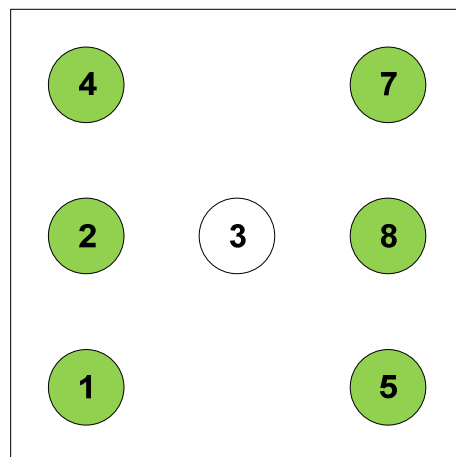
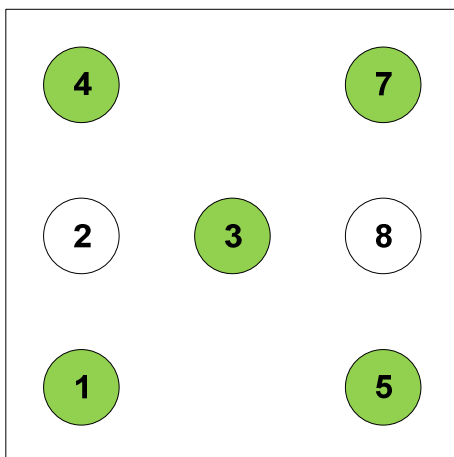
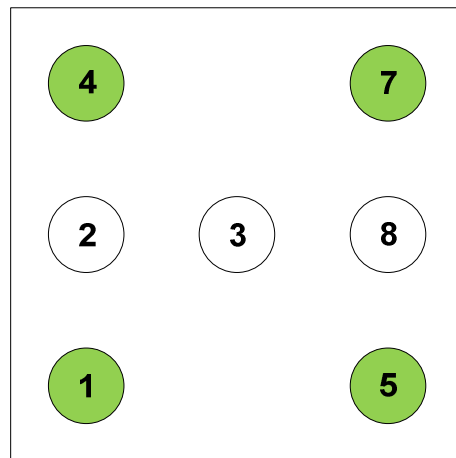
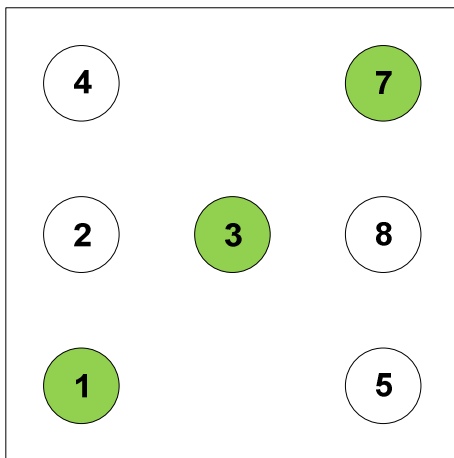
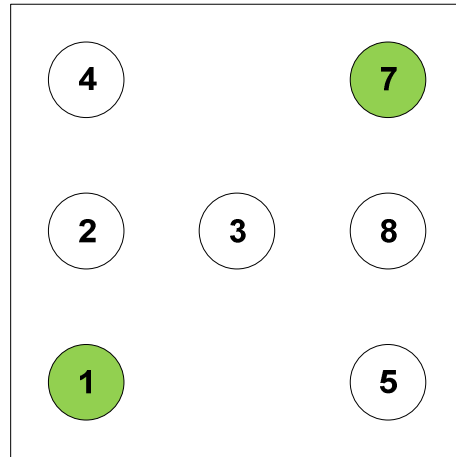
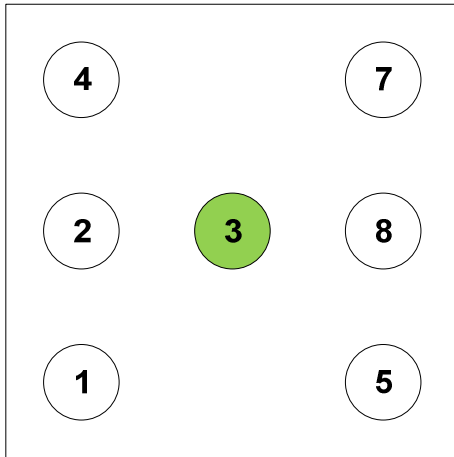
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

<http://www.hs-kl.de/~stefan.konrath>

Laborversuch „RGB-Würfel“



Realisierung eines „RGB-Würfels“



Die Ziffern in den o.a. Leuchtdioden (LED's) entsprechen den Anschlußbezeichnungen auf der Labor-Steckplatine „RGB-Wuerfel1“.

Realisieren Sie einen synchronen Zähler, aufgebaut mit **D-Flip Flops (SN74LS175)**, mit der sich wiederholenden Zählfolge von 0 bis 5 sowie den nachgeschalteten Codewandler zur Ansteuerung der Leuchtdioden, die wie die Augen eines Würfels angeordnet sind und so angezeigt werden. Siehe hierzu auch das beigefügte Blockschaltbild.

Dezimal	ZC	ZB	ZA	LED 1	LED 2	LED 3	LED 4	LED 5	LED 7	LED 8
0										
1										
2										
3										
4										
5										
6										
7										

Tabelle : Zuordnung Zählerstand zu Würfelergebnisanzeige über die LED-Anzeige, ZC, ZB und ZA entsprechen den Ausgängen des **synchronen** Zählers.

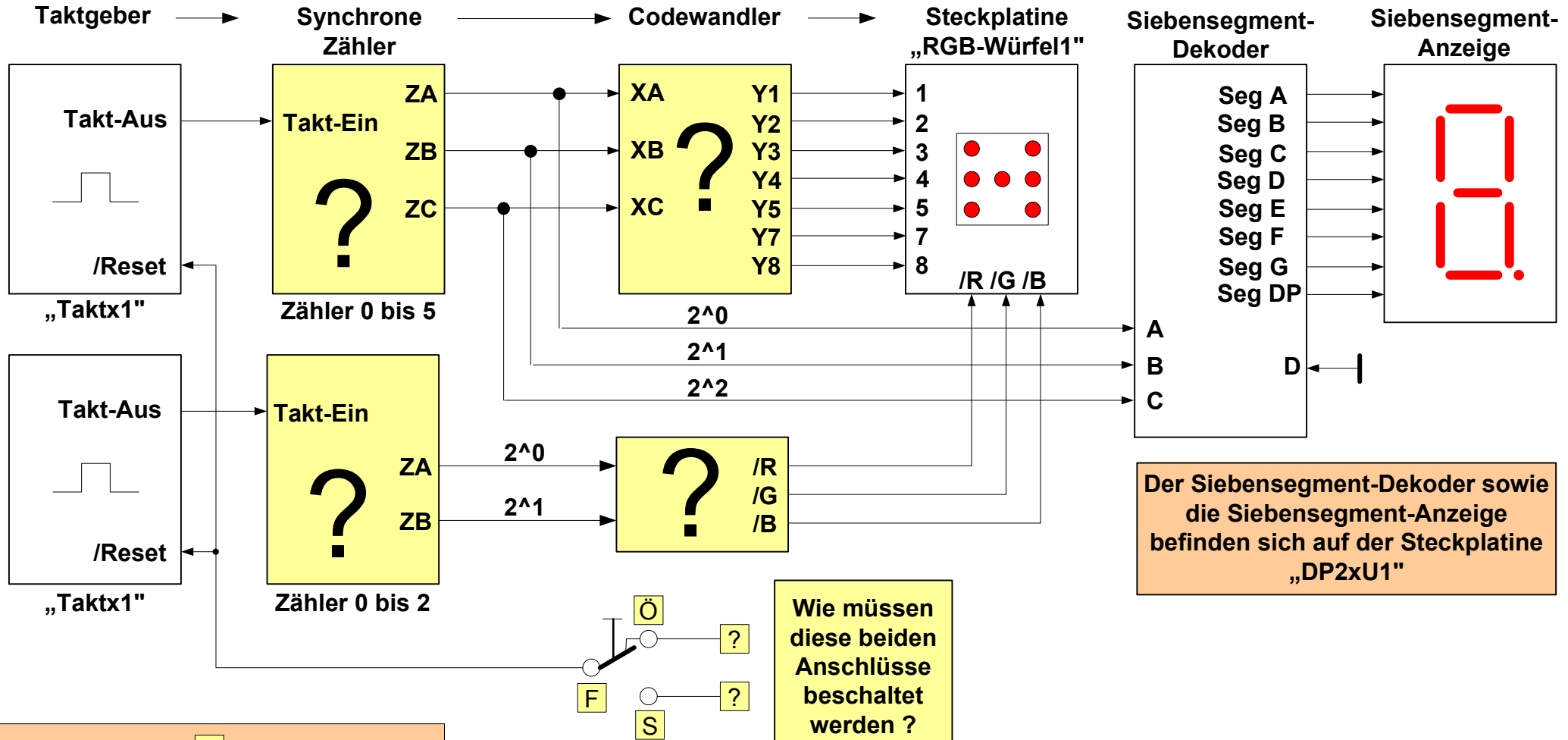
Realisieren Sie einen **weiteren synchronen** Zähler, aufgebaut mit **D-Flip Flops (SN74LS175)**, mit der sich wiederholenden Zählfolge zur Darstellung der 3 Farbdarstellungen für Rot, Grün und Blau.

Bitte ausfüllen !

Mit der vorhandenen RGB-LED sind ohne Pulsweitenmodulation ____ Farben möglich.

Beachten Sie bitte, die beiden Zähler zählen unabhängig zueinander die Zählfolgen durch !

Blockschaltbild „RGB-Würfel“



Hinweis: Alle **?** hinterlegten Funktionsblöcke sind von Ihnen zu realisieren!

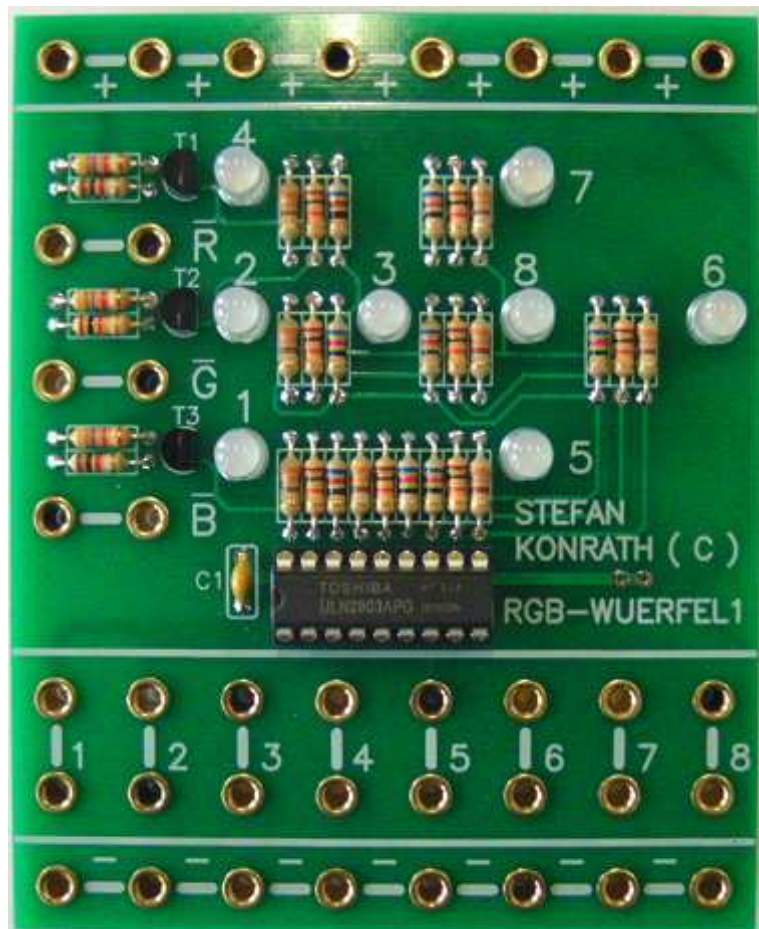
13.09.2018

Bitte Zählfolge der Zähler lt. Aufgabenstellung beachten !!

Technische Informatik
Blockschaltbild „RGB-Würfel“
Dipl. Inf. (FH) Stefan Konrath

„RGB-Würfel (RGB-WUERFEL1)

- +5V Spannungsversorgung
- /R Steuereingang
- /G Steuereingang
- /B Steuereingang
- LED- Ansteuerung
- Gnd (Minus)



Erläuterung der Steckfelder

+	Hier wird der Pluspol der +5V-Spannungsversorgung angeschlossen.
-	Hier wird der Minuspol der +5V-Spannungsversorgung angeschlossen.
(*) LED-Ansteuerung 1...8	Bei Anlegen einer logischen „1“ leuchtet die entsprechende LED in der durch /R, /G und /B angewählten Farbe auf.
(**) /R Steuereingang	Bei Anlegen einer logischen „0“ leuchtet die entsprechende LED in der angewählten Farbe auf.
(**) /G Steuereingang	Bei Anlegen einer logischen „0“ leuchtet die entsprechende LED in der angewählten Farbe auf.
(**) /B Steuereingang	Bei Anlegen einer logischen „0“ leuchtet die entsprechende LED in der angewählten Farbe auf.

Bemerkung :

Das gewünschte Anzeigeresultat wird nur dann erreicht, wenn die Steuereingänge (*) in Kombination mit den Steuereingängen (**) angesteuert werden!



Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

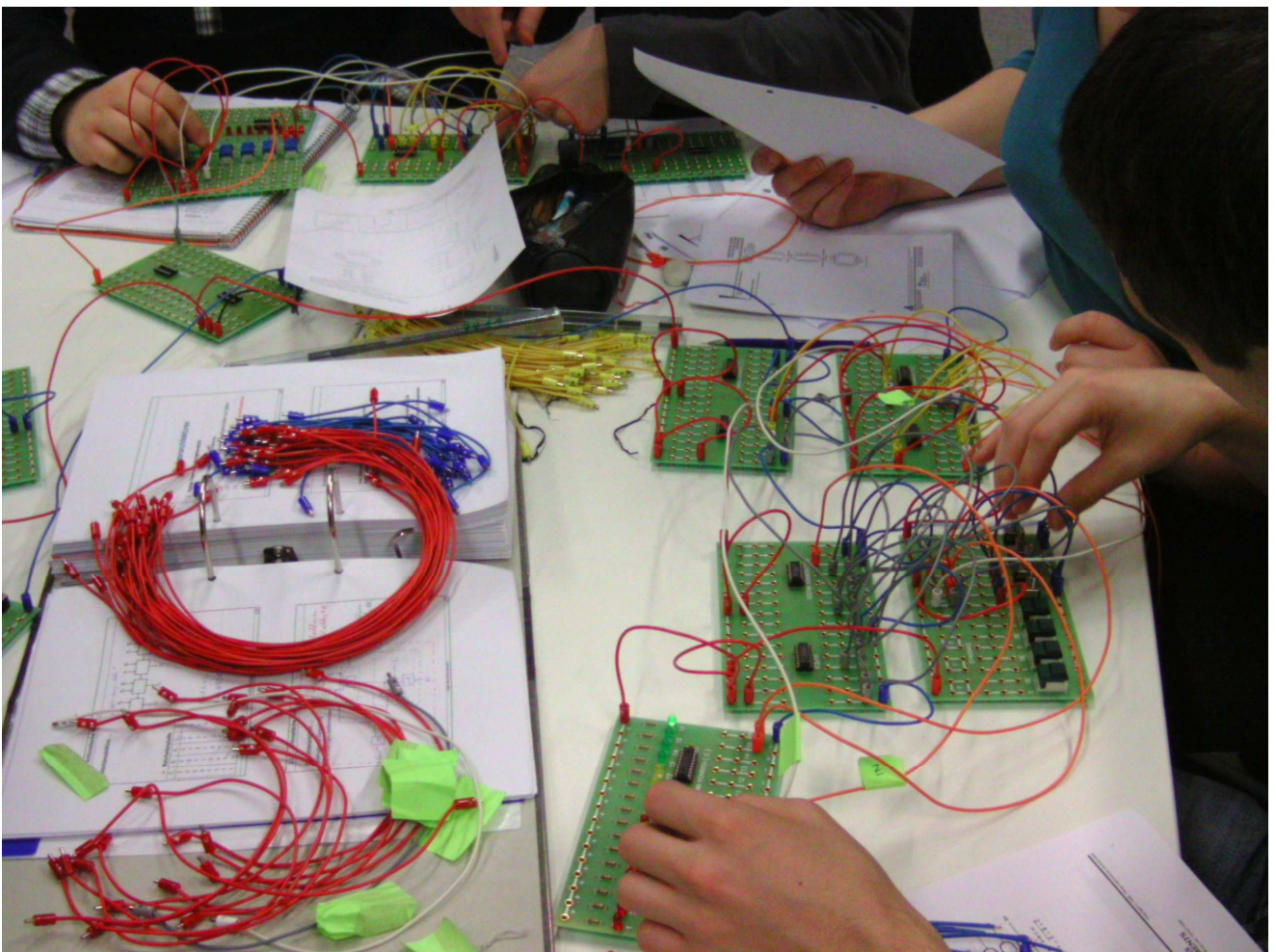
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Siebensegment- Decoder (BCD)
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des synchronen Zählers nur Flip-Flops des Typs „ SN74LS175 “. Beachten Sie die Zählfolge sowie die Darstellung der Pseudotetraden auf dem Siebensegment-Display.
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Laborversuch „Siebensegment-Decoder (BCD)“



Aufgabenstellung :

Zu entwerfen und im Labor physikalisch zu realisieren ist ein **Siebensegment-Decoder**, welcher **BCD-Zahlen** in Ansteuersignale einer Siebensegment-Anzeige (DP2xU1) derart umsetzt, dass die Echten Tetraden als Dezimalziffer und die Pseudotetraden als Zeichen 'E' (Error) dargestellt werden. Die Ansteuerung des Dezimalpunkts der Anzeige visualisiert die Taktfrequenz des Taktgebers (Taktx1).

Der zugehörige **synchrone** Zähler, aufgebaut mit **D-Flip Flops (SN74LS175)**, soll BCD-Zahlen einschließlich der Pseudotetraden in **absteigender** Reihenfolge zyklisch generieren.

Zählfolge (hexadez.) : F, E, D, ..., A, 9, 8, ..., 1, 0, F, E, ...

Vorgehen :

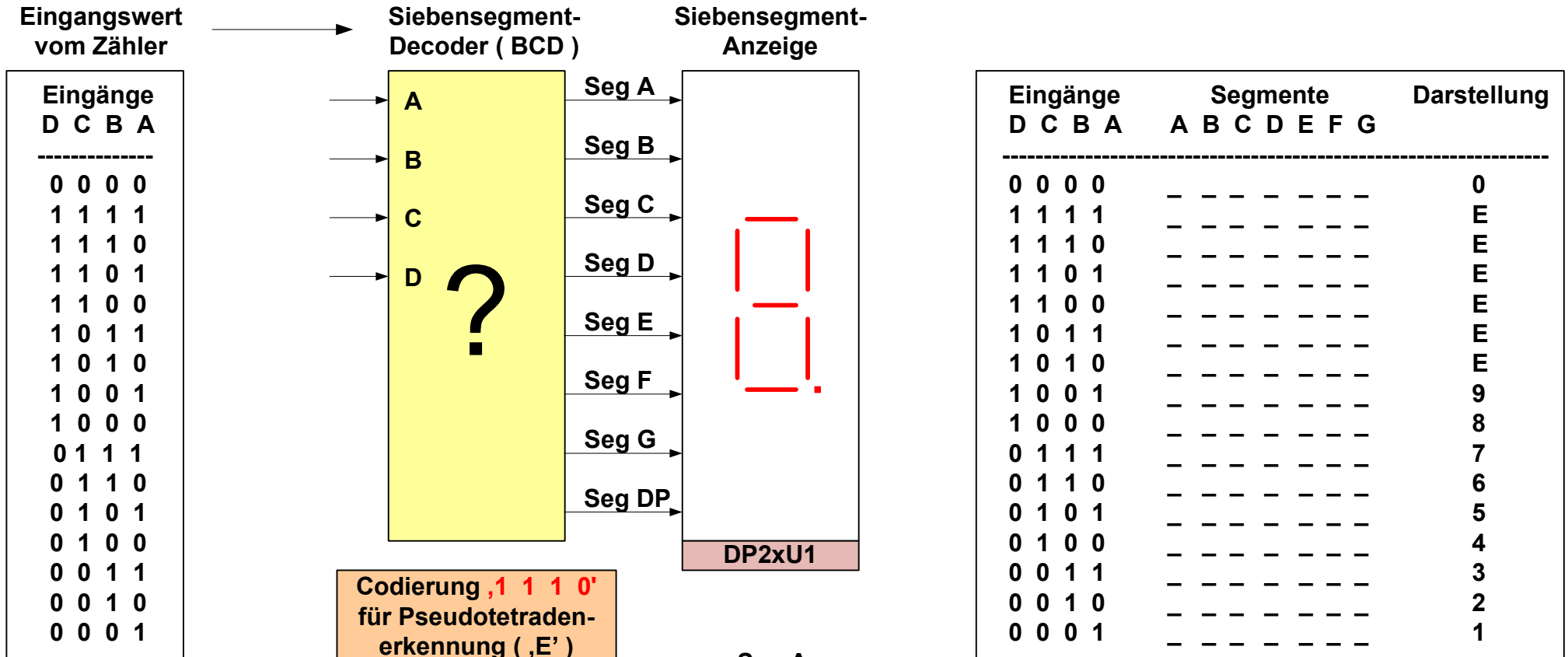
Die Aufgabe gliedert sich in zwei Bereiche. Den synchronen Zähler mit der vorgegebenen Zählfolge und den Siebensegment-Decoder (BCD). Realisieren Sie zunächst den Zähler und anschließend den Siebensegment-Decoder (BCD). Koppeln Sie nun die 4 Zählerausgänge (ZA .. ZD) mit den entsprechenden Eingängen (A .. D) des Siebensegment-Decoders (BCD). Durch dieses Vorgehen ist der Siebensegment-Decoder (BCD) nicht vom Design des Zählers abhängig. D.h. er decodiert **jeden** beliebigen Eingangswert in die entsprechende Ziffer.

Treffen Sie für Ihr Design unbedingt eine optimale Bausteinauswahl aus der Liste „Verfügbare Digital-Bausteine“.

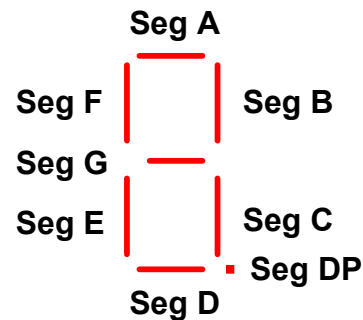
Weitere Details entnehmen Sie bitte den folgenden Blockschaltbildern.

Die Dokumentationen zu den Platinen „DP2xU1“ und „Taktx1“ finden Sie im Anhang dieses Dokumentes.

Siebensegment-Decoder (BCD)



Hinweis : Alle ? hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !



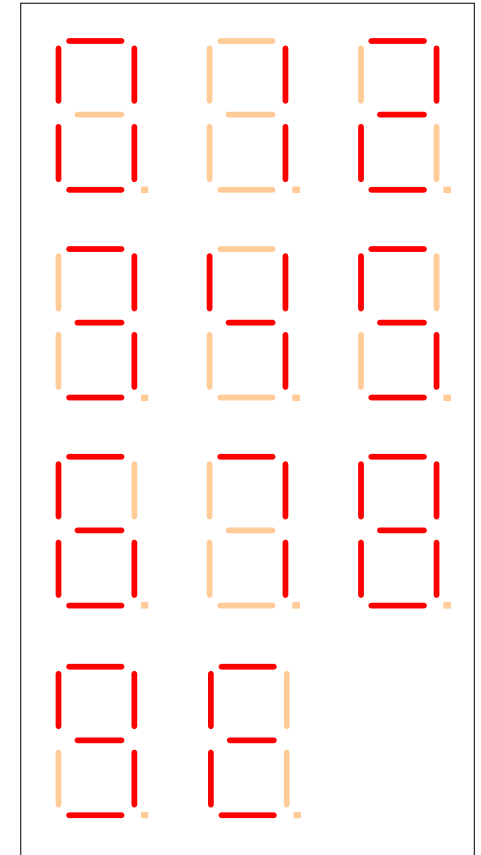
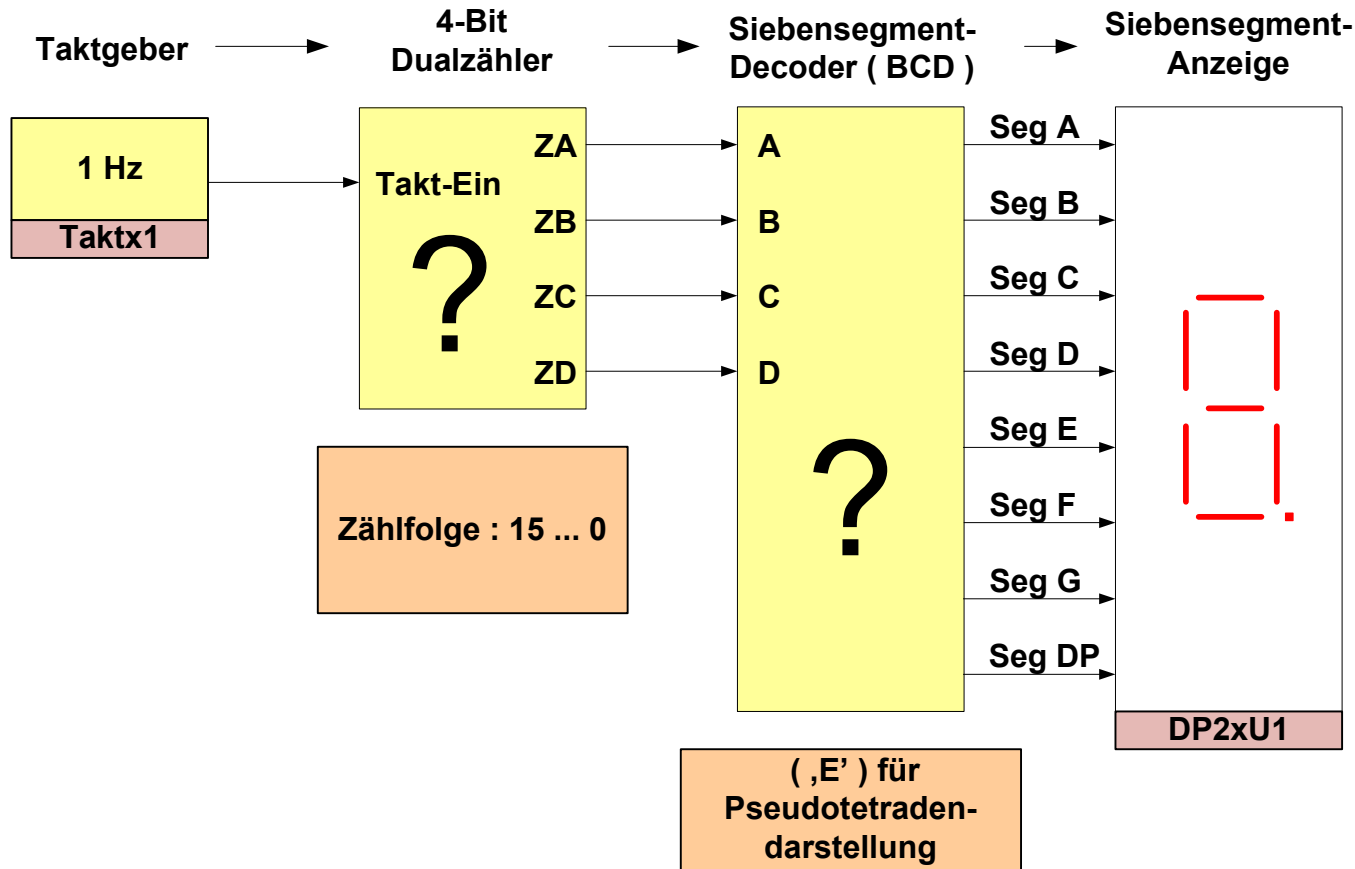
Technische Informatik

Siebensegment-Decoder (BCD)

Dipl. Inf. (FH) Stefan Konrath

11.09.2018

Siebensegment-Decoder (BCD)



Hinweis : Alle ? hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !

Technische Informatik

Siebensegment-Decoder (BCD)

Dipl. Inf. (FH) Stefan Konrath

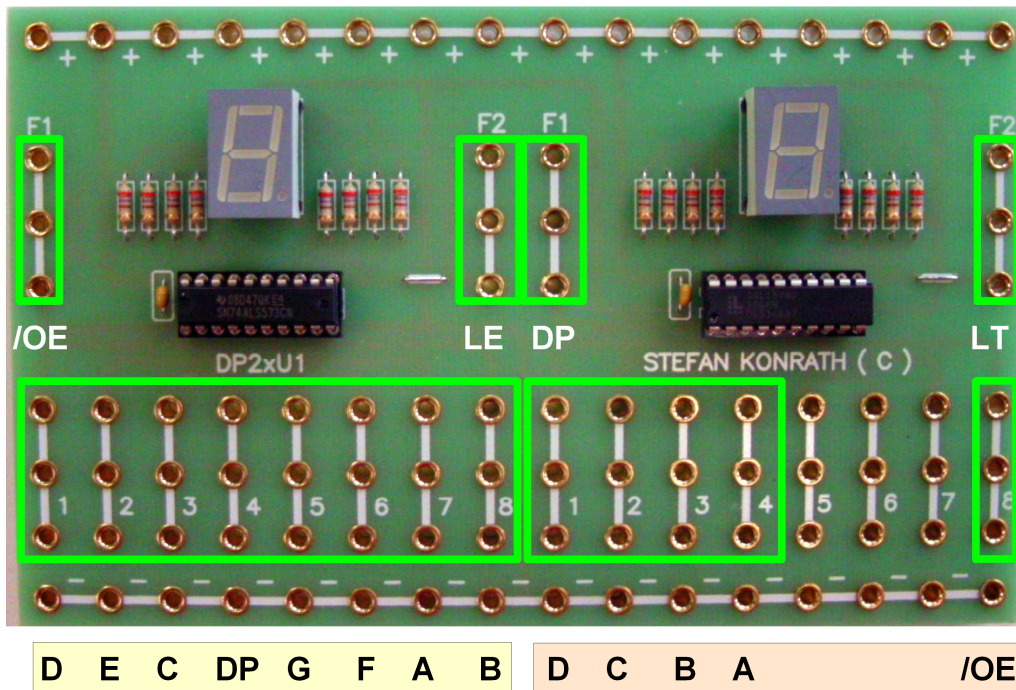
11.09.2018

Verfügbare Digital-Bausteine

Bezeichnung	Funktion		Bezeichnung	Funktion		Bezeichnung	Funktion
SN74LS00	4 x NAND mit je 2 Eingängen		SN74LS32	4 x OR mit je 2 Eingängen		SN74LS139	2 Bit Bin.Dekoder, DEMUX 2 zu 4
SN74LS02	4 x NOR mit je 2 Eingängen		SN74LS51	AND / NOR - Kombination		SN74L151	8 zu 1 Multiplexer
SN74LS04	6 x Inverter		SN74LS55	AND / NOR - Kombination		SN74L161	Synchr.prog. 4-Bit Bin.Zähler,Clr
SN74LS08	4 x AND mit je 2 Eingängen		SN74LS74	Dual-D-Flip-Flop		SN74LS175	Quad, synchr. D-Flip-Flop
SN74LS10	3 x NAND mit je 3 Eingängen		SN74LS85	4 Bit Vergleicher		SN74LS260	2 x NOR mit je 5 Eingängen
SN74LS11	3 x AND mit je 3 Eingängen		SN74LS86	4 x EXOR mit je 2 Eingängen		SN74LS393	Asynchr. 4 Bit Binärzähler
SN74LS14	6 x Inverter, Schmitt-Trigger		SN74LS125	4-Bus-Leitungstreiber, tristate		SN74LS573	8-Bit Latch
SN74LS20	2 x NAND mit je 4 Eingängen		SN74LS126	4-Bus-Leitungstreiber, tristate		GAL16V8	Siebensegment-Dekoder
SN74LS21	2 x AND mit je 4 Eingängen		SN74LS133	1 x NAND mit je 13 Eingängen			
SN74LS27	3 x NOR mit je 3 Eingängen		SN74LS136	4 x EXOR mit je 2 Eingängen			
SN74LS30	1 x NAND mit je 8 Eingängen		SN74LS138	3 Bit Bin.Dekoder, DEMUX 3 zu 8			

Siebensegmentanzeige (DP2xU1)

+5V Spannungsversorgung



Gnd (Minus)

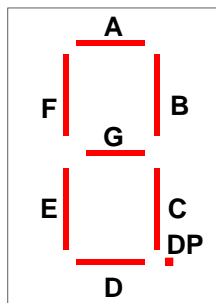
Bei dieser Steckplatine handelt es sich um eine universelle Anzeigeplatine. Je nach Bestückung der beiden IC's stehen verschiedene Funktionalitäten (4 Varianten) zur Verfügung.

Links bestückt mit SN74ALS573	Rechts bestückt mit SN74ALS573
Links bestückt mit SN74ALS573 (Standard)	Rechts bestückt mit GAL16V8D (Standard)
Links bestückt mit GAL16V8D	Rechts bestückt mit SN74ALS573
Links bestückt mit GAL16V8D	Rechts bestückt mit GAL16V8D

Erläuterung der Steckfelder

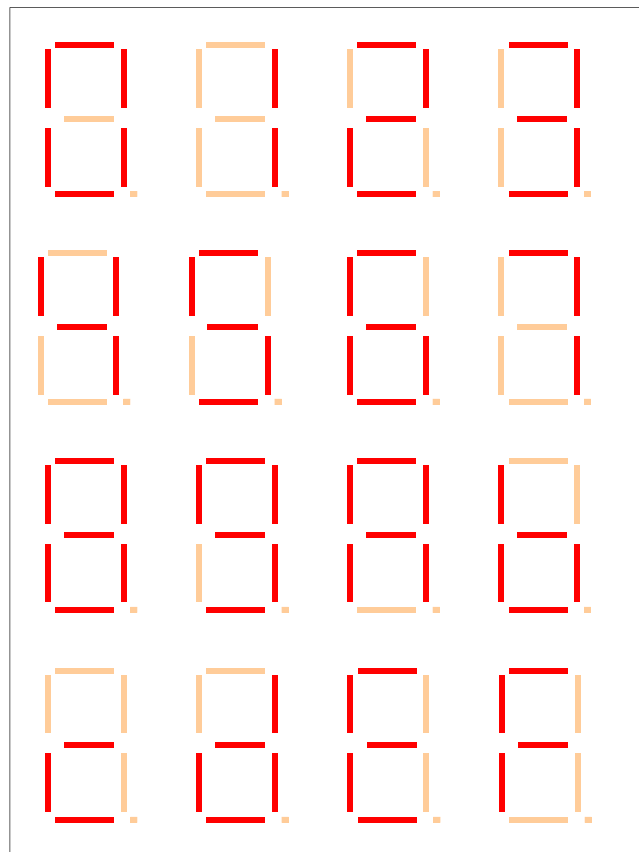
Bestückt mit SN74ALS573	Dieser Baustein ist ein sog. Latch mit der Funktion als Segmenttreiber. Die diskrete Ansteuerung der Anzeigesegmente geschieht wie auf der linken Seite der Platinenabbildung exemplarisch dargestellt über die Segmenteingänge A..DP.
Segmenteingänge A..DP	Logisch-„1“- Pegel lässt das entsprechende Anzeigesegment aufleuchten.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Angesteuerte Segmente leuchten nun auf, ansonsten bleibt die Anzeige komplett dunkel !
LE	Latch-Enable-Eingang. Bei logisch-„1“- Pegel werden die Informationen an den Segmenteingängen A..DP direkt zur Anzeige weitergegeben. Eine Änderung an den Eingängen wird sofort angezeigt. Bei logisch-„0“- Pegel wird die letzte Information an den Segmenteingängen im Baustein gespeichert und auf der Anzeige dargestellt.

Segmentzuordnung



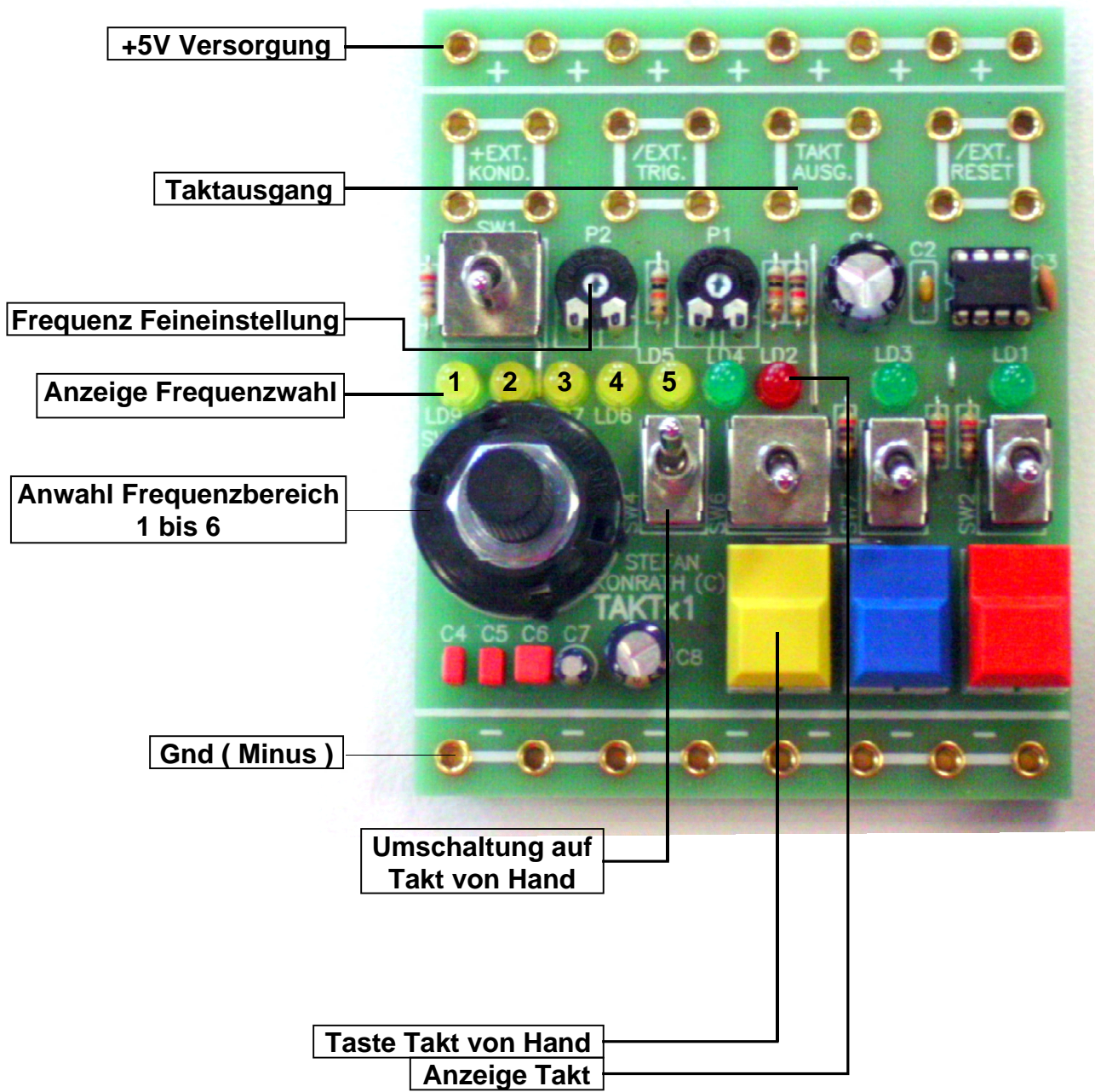
Bestückt mit GAL16V8D	Dieser Baustein ist ein sog. programmierbarer Logikbaustein mit der Funktion als Siebensegment-Hex-Dekoder. Die diskrete Ansteuerung der Anzeigensegmente geschieht hier über ein logisches Schaltnetz im Baustein. Wie auf der rechten Seite der Platinenabbildung exemplarisch dargestellt, erfolgt die Ansteuerung über die 4 Dateneingänge A..D.
Dateneingänge A..D	Je nach Ansteuerung dieser Eingänge wird die entsprechende Hexadezimalzahl auf der Anzeige dargestellt. Dabei ist „A“ der Eingang mit der niedersten Wertigkeit (2^0).
DP Eingang	Dezimalpunkt -Eingang. Bei logisch-„1“- Pegel wird der Dezimalpunkt der Anzeige angesteuert.
LT	Lampen-Test-Eingang. Bei logisch-„1“ - Pegel werden alle Segmente der Anzeige angesteuert und beginnen zu leuchten. Hierbei kann ein eventueller elektrischer oder mechanischer Defekt der Anzeige aufgezeigt werden.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Die anzuzeigende Hexadezimalzahl kann nun aufleuchten.

Hexadezimale Darstellung (Hex. 0..F für dez. 0..15)



Taktgenerator (Bedienelemente)

Für den Laborbetrieb sind die Kippschalter hier in der Standardeinstellung abgebildet.



Erläuterung der Bedienelemente und Steckfelder

<p><u>+5V Versorgung</u> :</p>	<p>8 Steckfelder. Anschluß der +5V Spannungsversorgung. Auf Polung achten !</p>
<p><u>Taktausgang</u> :</p>	<p>4 Steckfelder. Hier liegt der erzeugte Takt zur Weiterverarbeitung an.</p>
<p><u>Frequenz Feineinstellung</u> :</p>	<p>Trimpotentiometer zur Frequenzeinstellung in dem durch den Drehschalter gewählten Frequenzbereich. Siehe hierzu auch „<u>Anzeige Frequenz- Zeitwahl</u>“ .</p> <p style="padding-left: 40px;"> Frequenzbereich 1 : 0,1 Hz bis 1 Hz Frequenzbereich 2 : 1 Hz bis 10 Hz Frequenzbereich 3 : 10 Hz bis 100 Hz Frequenzbereich 4 : 100 Hz bis 1 kHz Frequenzbereich 5 : 1 kHz bis 10 kHz Frequenzbereich 6 : benutzerdefiniert, für „<u>+Ext. Kondensator</u>“ </p> <p>Die linke Potentiometereinstellung entspricht der niedrigsten Frequenz, die rechte Potentiometereinstellung der höchsten Frequenz im gewählten Bereich.</p>
<p><u>Anzeige Frequenz- Zeitwahl</u> :</p>	<p>LED-Anzeige des gewählten Frequenz- bzw. Zeitbereiches.</p>
<p><u>Anwahl Frequenz- bzw. Zeitbereich 1 bis 6</u> :</p> <p>Standard : Frequenzbereich 1</p>	<p>Drehschalter für die gewünschte Einstellung. Siehe hierzu auch „<u>Anzeige Frequenz- Zeitwahl</u>“</p>
<p><u>Gnd (Minus)</u> :</p>	<p>8 Steckfelder. Anschluss „Gnd“ der Spannungsversorgung. Auf Polung achten !</p>
<p><u>Umschaltung auf Takt von Hand</u> :</p> <p>Standard : Untere Kippstellung</p>	<p>Umschaltung zwischen Takt vom Timerbaustein oder manuellem Takt über „<u>Taste Takt von Hand</u>“ .</p> <p style="padding-left: 40px;">Obere Kippstellung : Takterzeugung per Taste „<u>Taste Takt von Hand</u>“</p> <p style="padding-left: 40px;">Untere Kippstellung : Takt vom Timerbaustein.</p>
<p><u>Taste Takt von Hand</u> :</p>	<p>Taste zur Erzeugung des manuellen Taktes. Siehe hierzu auch „<u>Umschaltung Taktquelle</u>“ und „<u>Umschaltung Low / High Aktiv bei Takt von Hand</u>“ .</p>
<p><u>Anzeige Takt</u> :</p>	<p>LED-Anzeige des aktuellen Taktsignals am „<u>Taktausgang</u>“</p>



Dipl. Inf. (FH) Stefan Konrath
stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357
<http://www.hs-kl.de/~stefan.konrath>

Technische Informatik - Laborbesuch

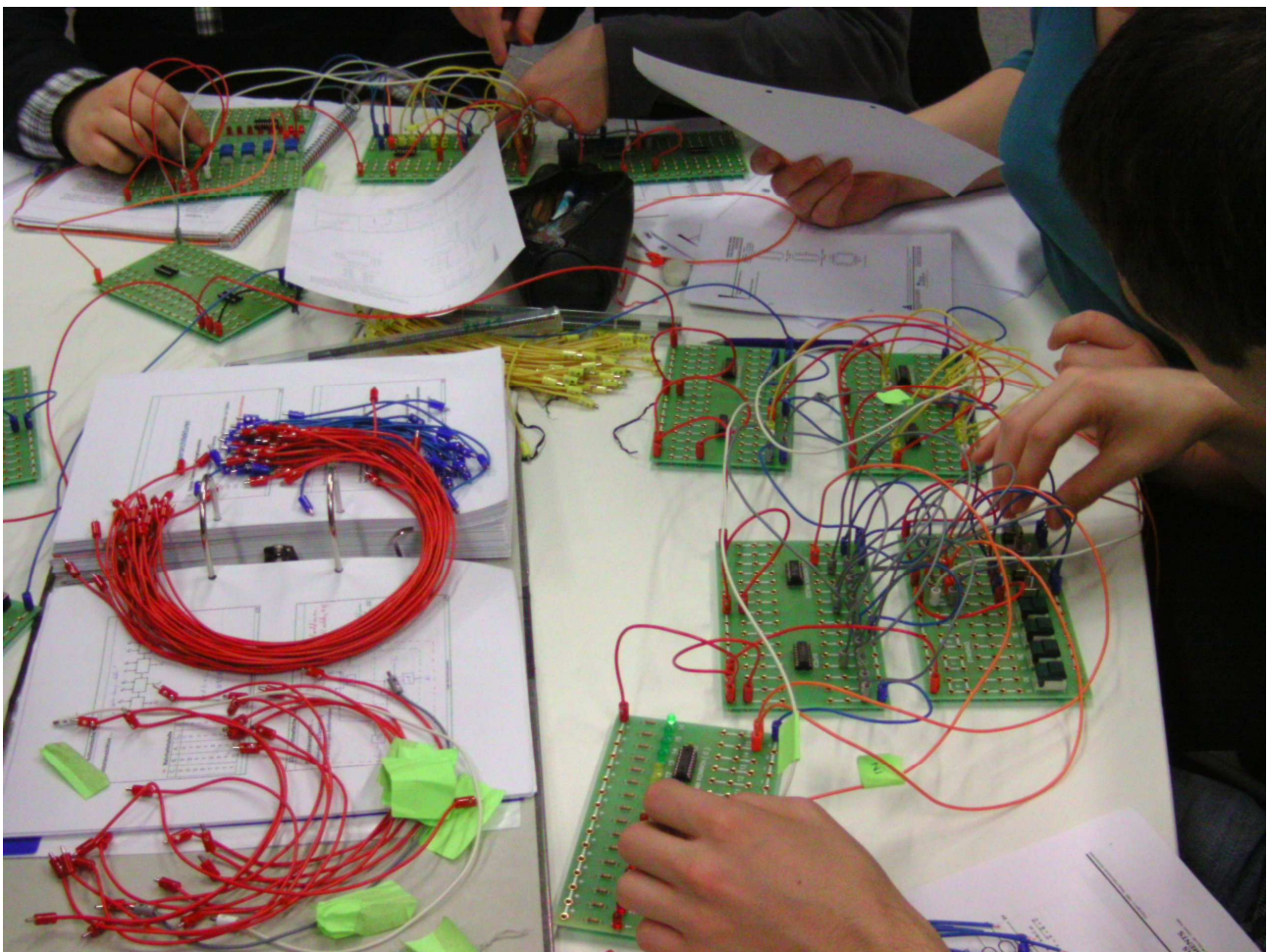
Vorname / Name	
Matrikelnummer	
Kurzbezeichnung der Aufgabe	Siebensegment- Decoder (Hex)
Wichtige Hinweise zur Bausteinauswahl	Verwenden Sie zur Realisierung des synchronen Zählers nur Flip-Flops des Typs „SN74LS175“. Beachten Sie die Zählfolge des Zählers..
Datum der Laborübung	
Unterschrift (Ich habe die Laborübung eigenständig und ohne wesentliche fremde Hilfe erstellt)	

Dipl. Inf. (FH) Stefan Konrath

stefan.konrath@hs-kl.de, Raum O228, Tel. 0631-3724-5357

<http://www.hs-kl.de/~stefan.konrath>

Laborversuch „Siebensegment-Decoder (Hex)“



Aufgabenstellung :

Zu entwerfen und im Labor physikalisch zu realisieren ist ein **Siebensegment-Decoder**, welcher **Hex-Zahlen** in Ansteuersignale einer Siebensegment-Anzeige (DP2xU1) derart umsetzt, dass der aktuelle Zählerstand als Zeichen dargestellt wird. Die Ansteuerung des Dezimalpunkts der Anzeige visualisiert die Taktfrequenz des Taktgebers (Taktx1).

Der zugehörige **synchrone** Zähler, aufgebaut mit **D-Flip Flops (SN74LS175)**, soll Hex-Zahlen in **aufsteigender** Reihenfolge zyklisch generieren.

Zählfolge (hexadez.) : 0, 1, 2, ..., 9, A, B, ..., E, F, 0, 1, ...

Vorgehen :

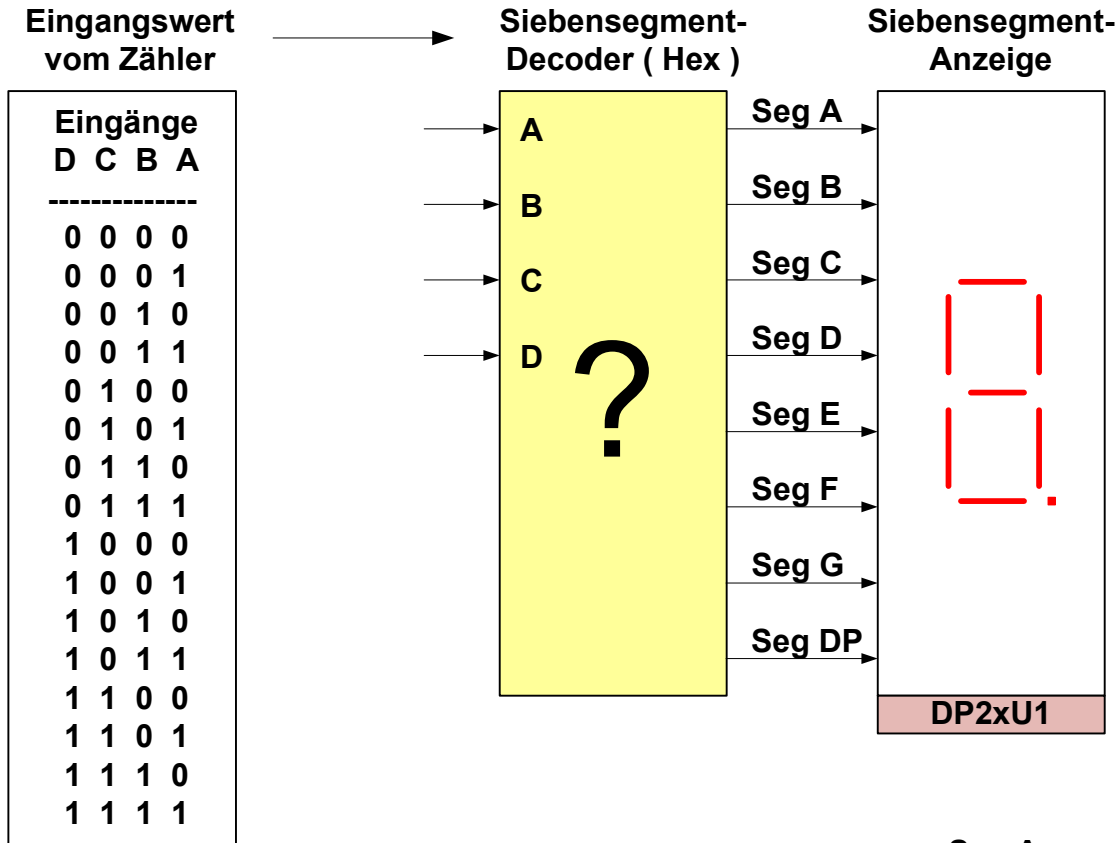
Die Aufgabe gliedert sich in zwei Bereiche. Den synchronen Zähler mit der vorgegebenen Zählfolge und den Siebensegment-Decoder (Hex). Realisieren Sie zunächst den Zähler und anschließend den Siebensegment-Decoder (Hex). Koppeln Sie nun die 4 Zählerausgänge (ZA ... ZD) mit den entsprechenden Eingängen (A ... D) des Siebensegment- Decoders (Hex). Durch dieses Vorgehen ist der Siebensegment-Decoder (Hex) nicht vom Design des Zählers abhängig. D.h. er decodiert **jeden** beliebigen Eingangswert in die entsprechende Ziffer.

Treffen Sie für Ihr Design unbedingt eine optimale Bausteinauswahl aus der Liste „Verfügbare Digital-Bausteine“.

Weitere Details entnehmen Sie bitte den folgenden Blockschaltbildern.

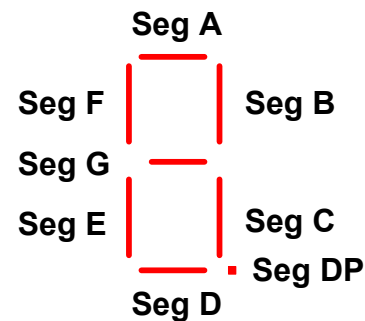
Die Dokumentationen zu den Platinen „DP2xU1“ und „Taktx1“ finden Sie im Anhang dieses Dokumentes.

Siebensegment-Decoder (Hex)



Eingänge				Segmente							Darstellung
D	C	B	A	A	B	C	D	E	F	G	
0	0	0	0	-	-	-	-	-	-	-	0
0	0	0	1	-	-	-	-	-	-	-	1
0	0	1	0	-	-	-	-	-	-	-	2
0	0	1	1	-	-	-	-	-	-	-	3
0	1	0	0	-	-	-	-	-	-	-	4
0	1	0	1	-	-	-	-	-	-	-	5
0	1	1	0	-	-	-	-	-	-	-	6
0	1	1	1	-	-	-	-	-	-	-	7
1	0	0	0	-	-	-	-	-	-	-	8
1	0	0	1	-	-	-	-	-	-	-	9
1	0	1	0	-	-	-	-	-	-	-	A
1	0	1	1	-	-	-	-	-	-	-	b
1	1	0	0	-	-	-	-	-	-	-	C
1	1	0	1	-	-	-	-	-	-	-	d
1	1	1	0	-	-	-	-	-	-	-	E
1	1	1	1	-	-	-	-	-	-	-	F

Hinweis : Alle hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !



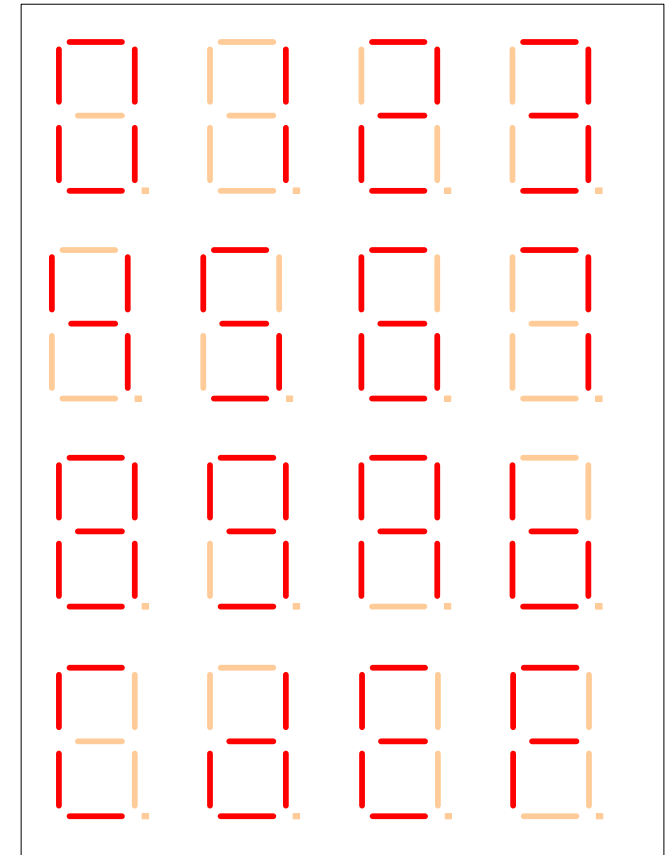
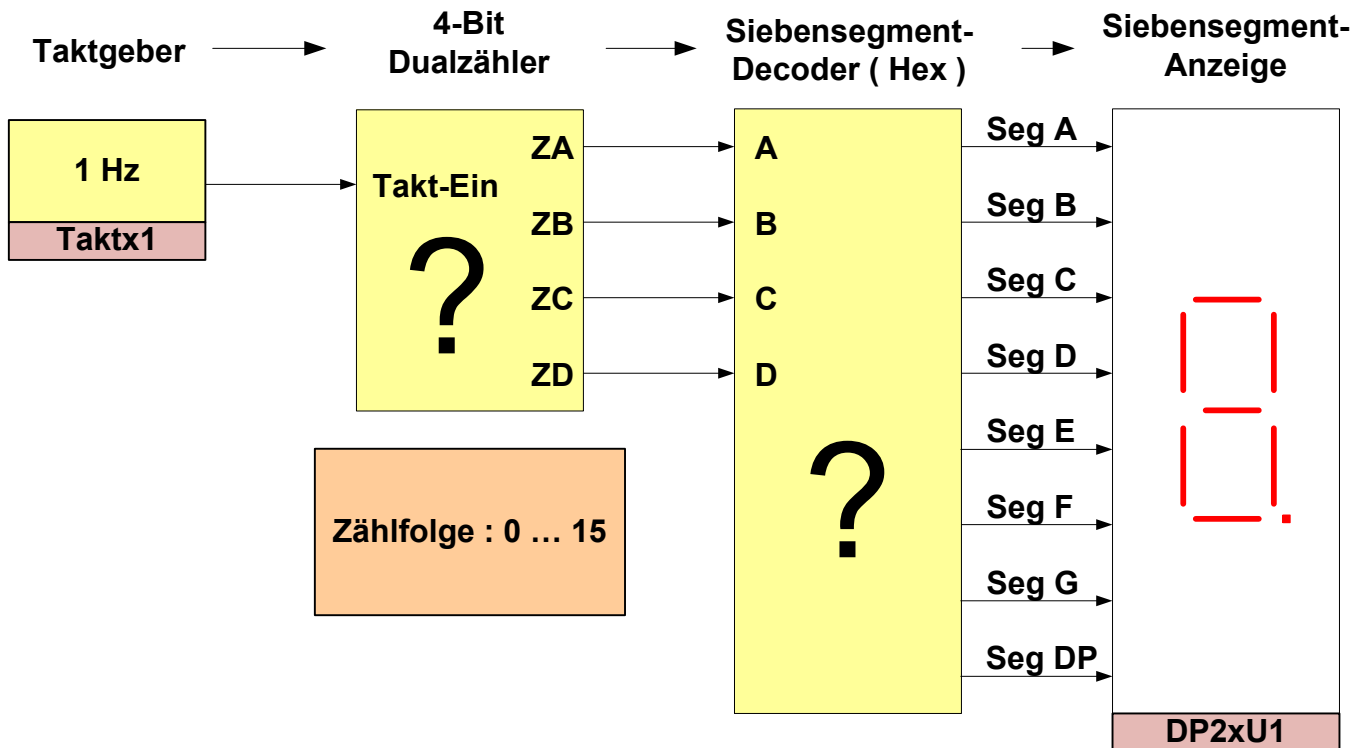
Technische Informatik

Siebensegment-Decoder (Hex)

Dipl. Inf. (FH) Stefan Konrath

11.09.2018

Siebensegment-Decoder (Hex)



Hinweis : Alle hinterlegten Funktionsblöcke sind von Ihnen zu realisieren !

Technische Informatik

Siebensegment-Decoder (Hex)

Dipl. Inf. (FH) Stefan Konrath

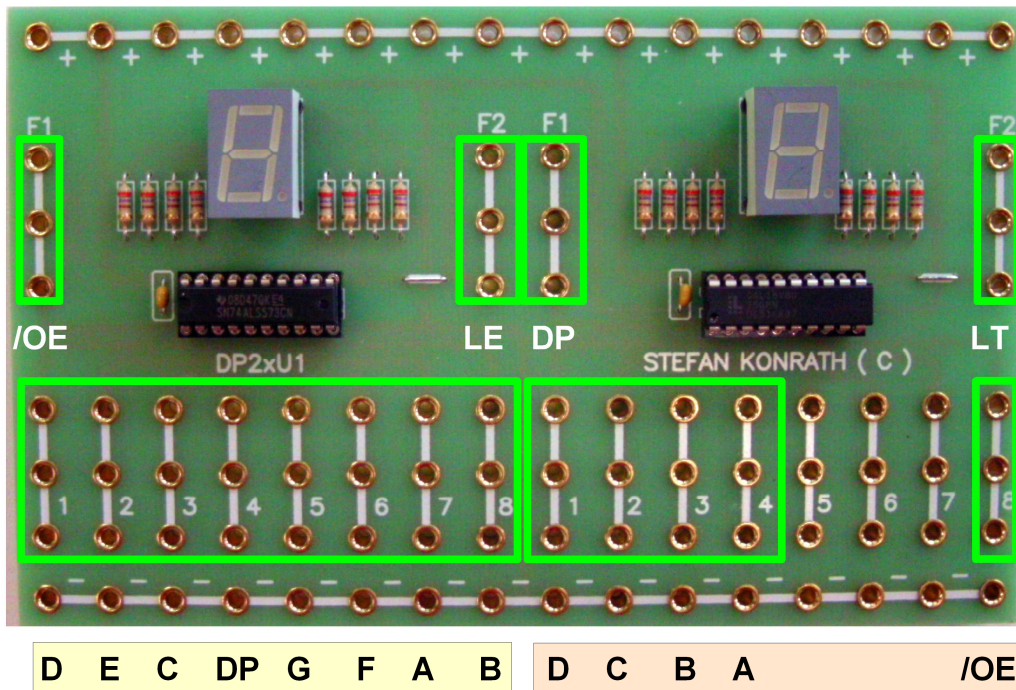
11.09.2018

Verfügbare Digital-Bausteine

Bezeichnung	Funktion		Bezeichnung	Funktion		Bezeichnung	Funktion
SN74LS00	4 x NAND mit je 2 Eingängen		SN74LS32	4 x OR mit je 2 Eingängen		SN74LS139	2 Bit Bin.Dekoder, DEMUX 2 zu 4
SN74LS02	4 x NOR mit je 2 Eingängen		SN74LS51	AND / NOR - Kombination		SN74L151	8 zu 1 Multiplexer
SN74LS04	6 x Inverter		SN74LS55	AND / NOR - Kombination		SN74L161	Synchr.prog. 4-Bit Bin.Zähler,Clr
SN74LS08	4 x AND mit je 2 Eingängen		SN74LS74	Dual-D-Flip-Flop		SN74LS175	Quad, synchr. D-Flip-Flop
SN74LS10	3 x NAND mit je 3 Eingängen		SN74LS85	4 Bit Vergleicher		SN74LS260	2 x NOR mit je 5 Eingängen
SN74LS11	3 x AND mit je 3 Eingängen		SN74LS86	4 x EXOR mit je 2 Eingängen		SN74LS393	Asynchr. 4 Bit Binärzähler
SN74LS14	6 x Inverter, Schmitt-Trigger		SN74LS125	4-Bus-Leitungstreiber, tristate		SN74LS573	8-Bit Latch
SN74LS20	2 x NAND mit je 4 Eingängen		SN74LS126	4-Bus-Leitungstreiber, tristate		GAL16V8	Siebensegment-Dekoder
SN74LS21	2 x AND mit je 4 Eingängen		SN74LS133	1 x NAND mit je 13 Eingängen			
SN74LS27	3 x NOR mit je 3 Eingängen		SN74LS136	4 x EXOR mit je 2 Eingängen			
SN74LS30	1 x NAND mit je 8 Eingängen		SN74LS138	3 Bit Bin.Dekoder, DEMUX 3 zu 8			

Siebensegmentanzeige (DP2xU1)

+5V Spannungsversorgung



Gnd (Minus)

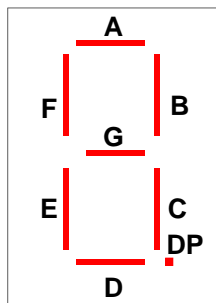
Bei dieser Steckplatine handelt es sich um eine universelle Anzeigeplatine. Je nach Bestückung der beiden IC's stehen verschiedene Funktionalitäten (4 Varianten) zur Verfügung.

Links bestückt mit SN74ALS573	Rechts bestückt mit SN74ALS573
Links bestückt mit SN74ALS573 (Standard)	Rechts bestückt mit GAL16V8D (Standard)
Links bestückt mit GAL16V8D	Rechts bestückt mit SN74ALS573
Links bestückt mit GAL16V8D	Rechts bestückt mit GAL16V8D

Erläuterung der Steckfelder

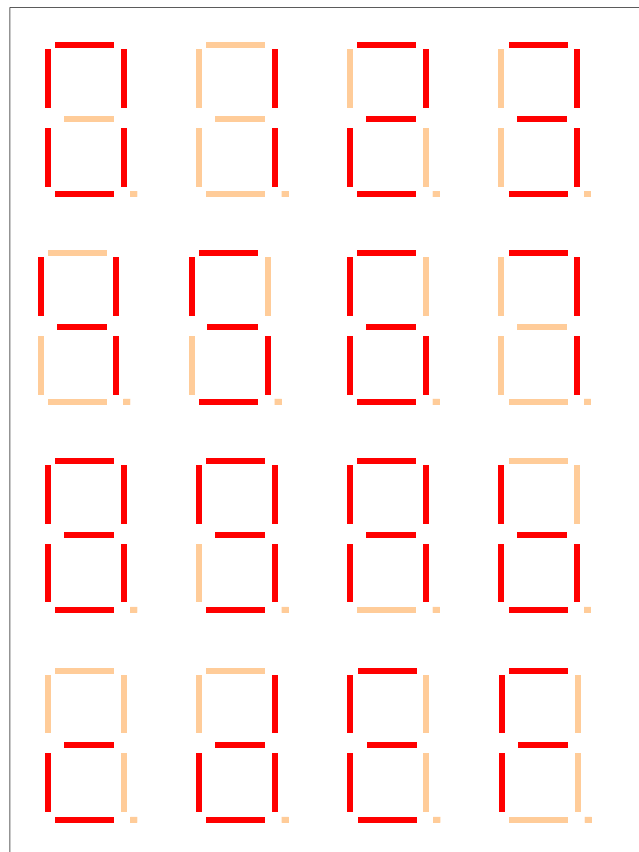
Bestückt mit SN74ALS573	Dieser Baustein ist ein sog. Latch mit der Funktion als Segmenttreiber. Die diskrete Ansteuerung der Anzeigesegmente geschieht wie auf der linken Seite der Platinenabbildung exemplarisch dargestellt über die Segmenteingänge A..DP.
Segmenteingänge A..DP	Logisch-„1“- Pegel lässt das entsprechende Anzeigesegment aufleuchten.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Angesteuerte Segmente leuchten nun auf, ansonsten bleibt die Anzeige komplett dunkel !
LE	Latch-Enable-Eingang. Bei logisch-„1“- Pegel werden die Informationen an den Segmenteingängen A..DP direkt zur Anzeige weitergegeben. Eine Änderung an den Eingängen wird sofort angezeigt. Bei logisch-„0“- Pegel wird die letzte Information an den Segmenteingängen im Baustein gespeichert und auf der Anzeige dargestellt.

Segmentzuordnung



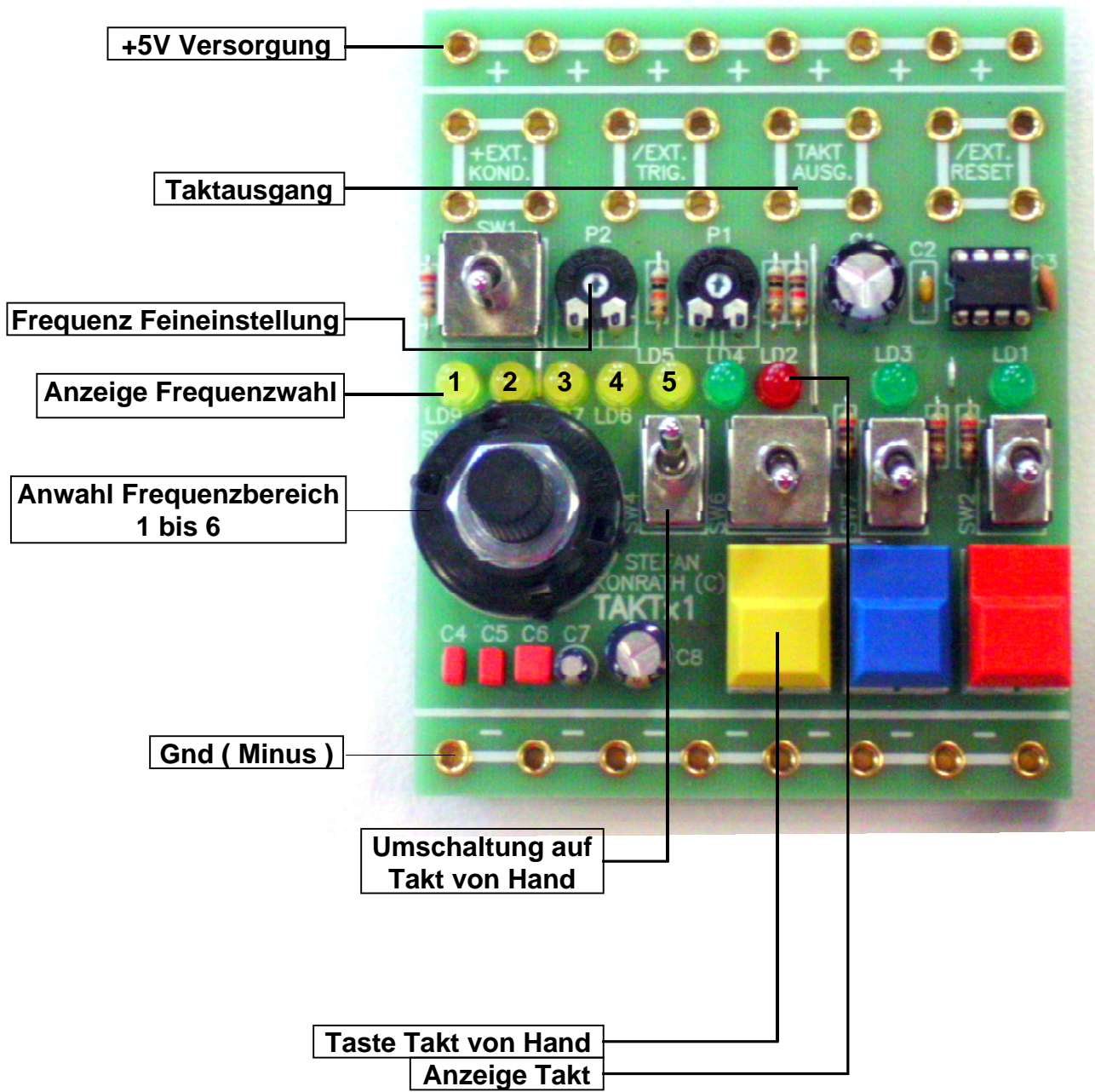
Bestückt mit GAL16V8D	Dieser Baustein ist ein sog. programmierbarer Logikbaustein mit der Funktion als Siebensegment-Hex-Dekoder. Die diskrete Ansteuerung der Anzeigensegmente geschieht hier über ein logisches Schaltnetz im Baustein. Wie auf der rechten Seite der Platinenabbildung exemplarisch dargestellt, erfolgt die Ansteuerung über die 4 Dateneingänge A..D.
Dateneingänge A..D	Je nach Ansteuerung dieser Eingänge wird die entsprechende Hexadezimalzahl auf der Anzeige dargestellt. Dabei ist „A“ der Eingang mit der niedrigsten Wertigkeit (2^0).
DP Eingang	Dezimalpunkt -Eingang. Bei logisch-„1“- Pegel wird der Dezimalpunkt der Anzeige angesteuert.
LT	Lampen-Test-Eingang. Bei logisch-„1“ - Pegel werden alle Segmente der Anzeige angesteuert und beginnen zu leuchten. Hierbei kann ein eventueller elektrischer oder mechanischer Defekt der Anzeige aufgezeigt werden.
/OE Eingang	Output-Enable-Eingang. Bei logisch-„0“- Pegel werden die Ausgangsstufen des Bausteines freigegeben ansonsten gesperrt. Die anzuzeigende Hexadezimalzahl kann nun aufleuchten.

Hexadezimale Darstellung (Hex. 0..F für dez. 0..15)



Taktgenerator (Bedienelemente)

Für den Laborbetrieb sind die Kippschalter hier in der Standardeinstellung abgebildet.



Erläuterung der Bedienelemente und Steckfelder

+5V Versorgung :	8 Steckfelder. Anschluß der +5V Spannungsversorgung. Auf Polung achten !
Taktausgang :	4 Steckfelder. Hier liegt der erzeugte Takt zur Weiterverarbeitung an.
Frequenz Feineinstellung :	<p>Trimpotentiometer zur Frequenzeinstellung in dem durch den Drehschalter gewählten Frequenzbereich. Siehe hierzu auch „Anzeige Frequenz- Zeitwahl“ .</p> <p>Frequenzbereich 1 : 0,1 Hz bis 1 Hz Frequenzbereich 2 : 1 Hz bis 10 Hz Frequenzbereich 3 : 10 Hz bis 100 Hz Frequenzbereich 4 : 100 Hz bis 1 kHz Frequenzbereich 5 : 1 kHz bis 10 kHz Frequenzbereich 6 : benutzerdefiniert, für „+Ext. Kondensator“</p> <p>Die linke Potentiometereinstellung entspricht der niedrigsten Frequenz, die rechte Potentiometereinstellung der höchsten Frequenz im gewählten Bereich.</p>
Anzeige Frequenz- Zeitwahl :	LED-Anzeige des gewählten Frequenz- bzw. Zeitbereiches.
Anwahl Frequenz- bzw. Zeitbereich 1 bis 6 : Standard : Frequenzbereich 1	Drehschalter für die gewünschte Einstellung. Siehe hierzu auch „ Anzeige Frequenz- Zeitwahl “
Gnd (Minus) :	8 Steckfelder. Anschluss „Gnd“ der Spannungsversorgung. Auf Polung achten !
Umschaltung auf Takt von Hand : Standard : Untere Kippstellung	<p>Umschaltung zwischen Takt vom Timerbaustein oder manuellem Takt über „Taste Takt von Hand“ .</p> <p>Obere Kippstellung : Takterzeugung per Taste „Taste Takt von Hand“</p> <p>Untere Kippstellung : Takt vom Timerbaustein.</p>
Taste Takt von Hand :	Taste zur Erzeugung des manuellen Taktes. Siehe hierzu auch „ Umschaltung Taktquelle “ und „ Umschaltung Low / High Aktiv bei Takt von Hand “ .
Anzeige Takt :	LED-Anzeige des aktuellen Taktsignals am „ Taktausgang “