

## Praktikum Digitaltechnik SS 2014

### Versuch 2

Stand: 20. 5. 2014

#### Versuchsplattform:

CPLD-Lehrgerät 12 mit CPLD Evaluation Board (Pollin). Eingebaute Ein- und Ausgabemittel: vier Taster, acht LEDs, serielle Schnittstelle. Zusätzlich Siebensegmentanzeige 09/13. Entwicklungssoftware: Xilinx ISE 10.1. Erprobung des Schnittstellentesters: am PC mittels Terminalprogramm.

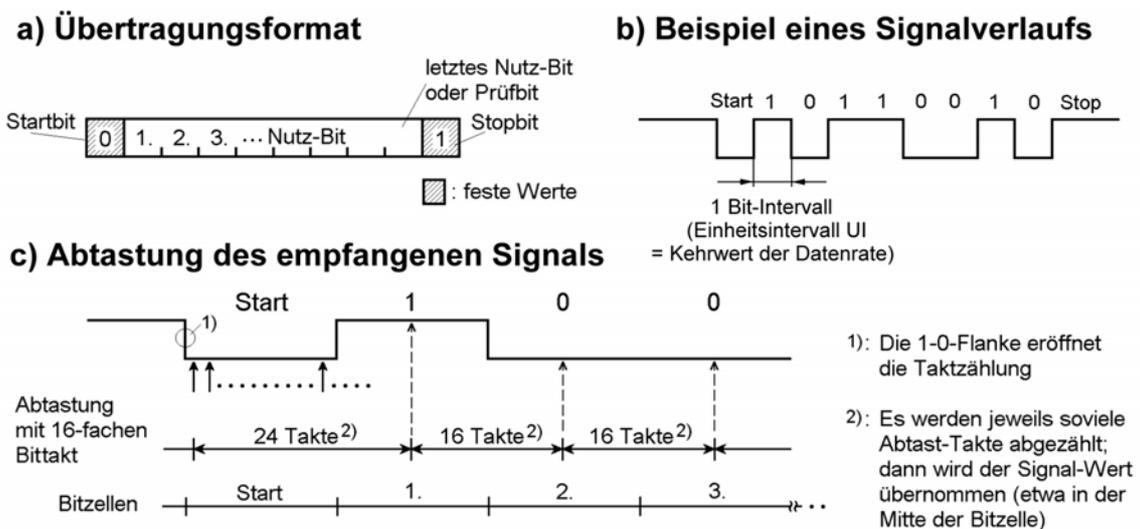
#### Aufgaben:

- Den Ausgangszustand herstellen.
- Schnittstellentester mit Dauerbetrieb.
- Schnittstellentester mit Einzelauslösung.
- Schnittstellentester mit Dauer- und Einzelbetrieb.
- Handzähler.
- Stoppuhr.
- Ziehen von Lottozahlen.

– Was nicht fertig wird, bleibt liegen. –

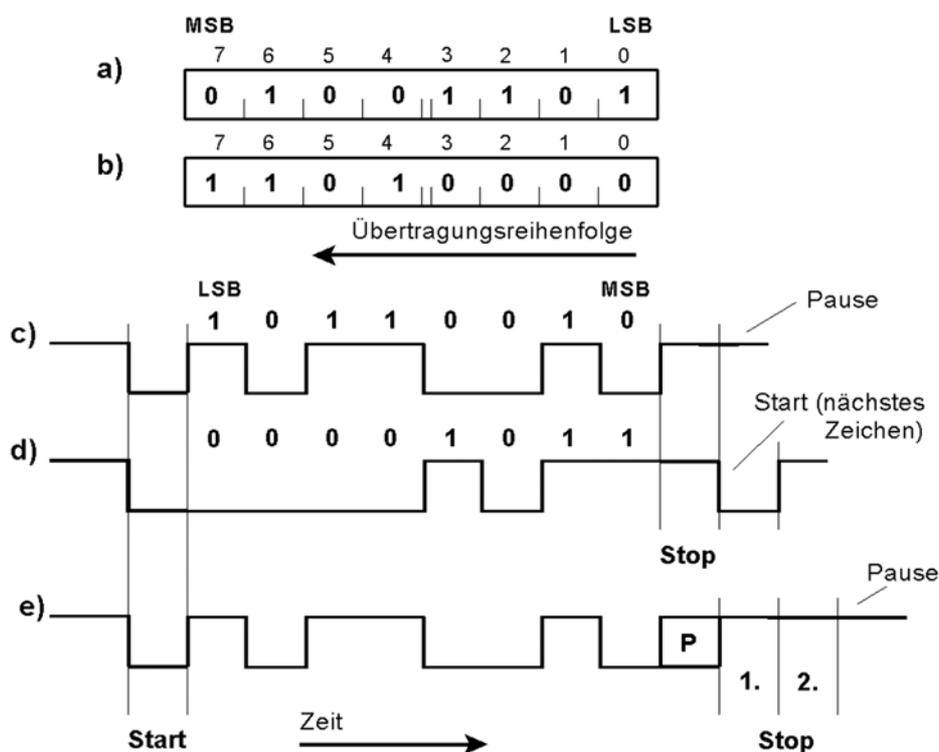
### Der Schnittstellentester. Kurzausbildung an der seriellen Schnittstelle

Die serielle Schnittstelle ist ein bitserielles asynchrones Interface. Je Richtung ist eine Datenleitung vorgesehen. Es werden einzelne Zeichen übertragen (Start-Stop-Verfahren). Der Übertragung liegt ein festes Zeitraster (Übertragungsrate) zugrunde. Die Zeichen werden mit zusätzlichen Start- und Stopbits voneinander abgegrenzt. Erkennt der Empfänger ein Startbit, so beginnt er, den ankommenden Datenstrom mit seinem Takt abzutasten. Das Abtasten endet mit dem Empfang des (bzw. der) Stopbits. Danach wartet der Empfänger auf das nächste Startbit. Codierung der Bitfolgen: NRZ. Übertragungsreihenfolge: von der niedrigstwertigen zur höchstwertigen Bitposition (LSB => MSB).



**Abb. 1** Signalübertragung nach dem Start-Stop-Verfahren (1). Grundlagen.

Der Ruhezustand wird durch einen Eins-Pegel signalisiert. Die Übertragung eines Zeichens beginnt mit einem Nullbit (Startbit). Der erste Eins-Null-Übergang – aus dem Ruhezustand heraus – veranlaßt den Empfänger, mit dem Abtasten des ankommenden Signals zu beginnen. Jede Bitzelle wird mehrmals abgetastet, beispielsweise mit einem Takt, der die 16fache Frequenz des Bittaktes hat. Trifft der erste Abtastimpuls auf den Eins-Null-Übergang, so hat man nach weiteren 24 solchen Impulsen ziemlich sicher die Mitte der nachfolgenden Bitzelle getroffen (diese enthält das erste Nutz-Bit des übertragenen Zeichens). Mit jeweils 16 Abtastimpulsen Abstand werden dann die weiteren Bitzellen näherungsweise in der Mitte abgetastet. Sind alle Zeichenbits (und ggf. ein zusätzliches Paritätsbit) übertragen worden, wird ein Einsbit als Endekennung (Stopbit) erwartet. Kommt keine 1, liegt ein Fehler vor (Fachbegriff: Framing Error). Daraufhin gelangt der Empfänger in den Ruhezustand und erwartet das nächste Startbit. Es gibt auch Übertragungsformate mit 2 oder mit  $1\frac{1}{2}$  Stopbits (" $1\frac{1}{2}$  Bits" bedeutet, daß der High-Pegel wenigstens  $1\frac{1}{2}$  Bitzellen lang anliegen muß).



**Abb. 2** Signalübertragung nach dem Start-Stop-Verfahren (2). Übertragungsbeispiele.

- a), b) Zwei zu übertragende Bytes (Beispiele). Die Übertragung beginnt stets mit dem niedrigstwertigen Bit (LSB).
- c) Übertragung von Byte a). 10-Bit-Format. Keine Parität, 1 Stopbit. Pause nach Übertragung des Zeichens.
- d) Übertragung von Byte b). 10-Bit-Format. Keine Parität, 1 Stopbit. Nach dem Stopbit folgt sofort das Startbit des nächsten Zeichens (schnellste Übertragungsfolge).
- e) Übertragung von Byte a). 12-Bit-Format. Paritätsbit (P), 2 Stopbits. Pause nach Übertragung des Zeichens. Das Paritätsbit dient zur Fehlerkontrolle. Je nach Einstellung der Paritätsprüfung wird es so gesetzt, daß die Anzahl der Einsen insgesamt entweder gerade oder ungerade ist. Im Beispiel werden 4 Einsen übertragen. Deshalb ist P bei gerader Parität = 0, bei ungerader = 1.

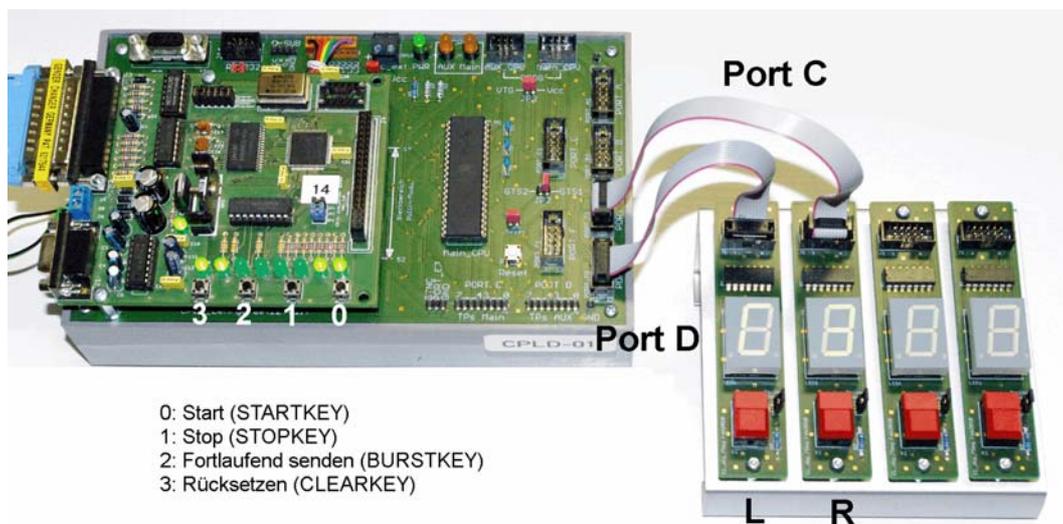
Gängige Übertragungsraten (in Bits/s):

50	75	<u>110</u>	134,5	<u>150</u>	200	<u>300</u>
<u>600</u>	<u>1200</u>	1800	2000	<u>2400</u>	3600	<u>4800</u>
7200	<u>9600</u>	14400	<u>19200</u>	38400	57600	115200

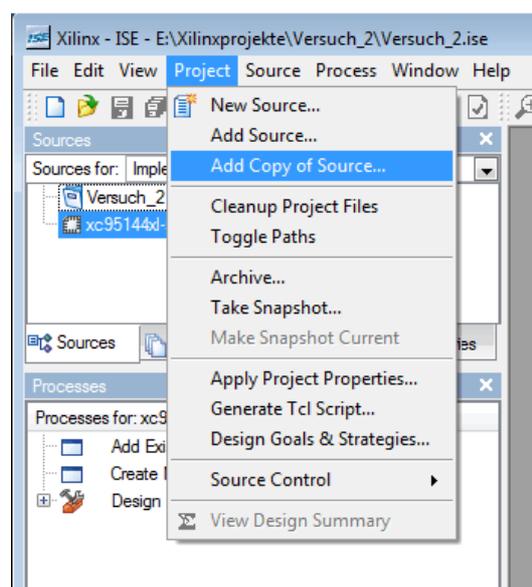
## Aufgabe 1: Den Ausgangszustand herstellen

Die Grundschaltung wurde bereits im 1. Versuch ausprobiert. Wir brauchen zwei 4-Bit-Zähler mit nachgeschalteten Siebensegmentanzeigen. Sie werden über die Tasten bedient, die unter den Anzeigen angeordnet sind. Wir begnügen uns mit dem Vorwärtszählen.

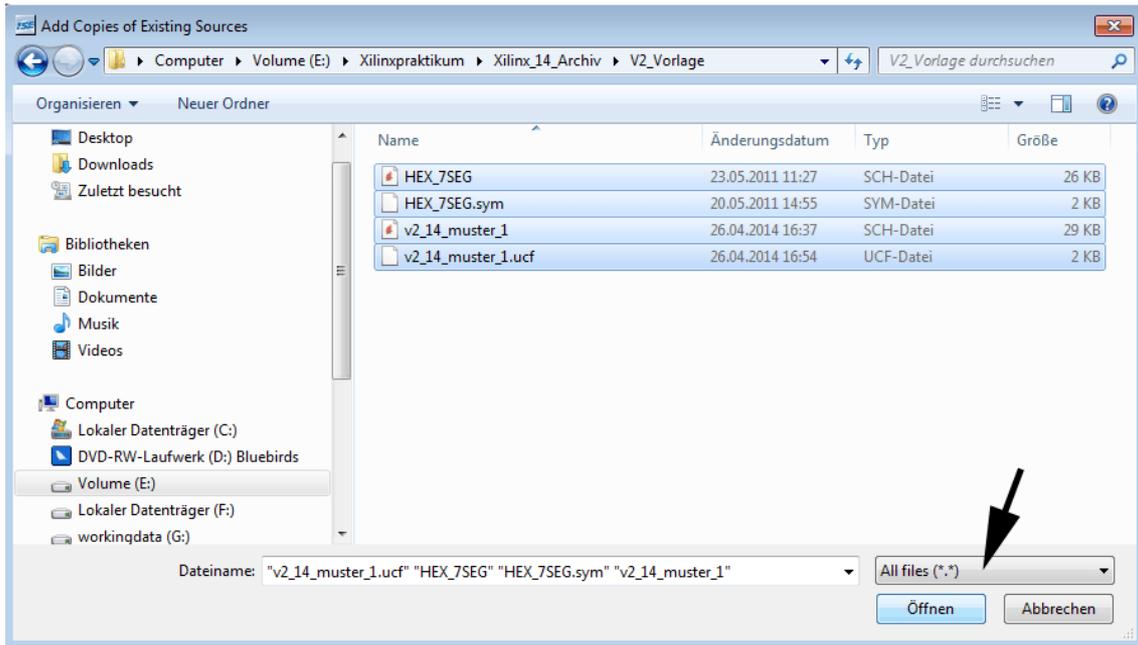
1. Lehrgerät anschließen
2. Xilinx-Umgebung aufrufen und ein neues Projekt anlegen, aber ohne eigene Quellen.
3. Quellen importieren.
4. Projekt durchlaufen lassen.
5. IMPACT aufrufen und Schaltkreis programmieren.
6. Funktionserprobung.



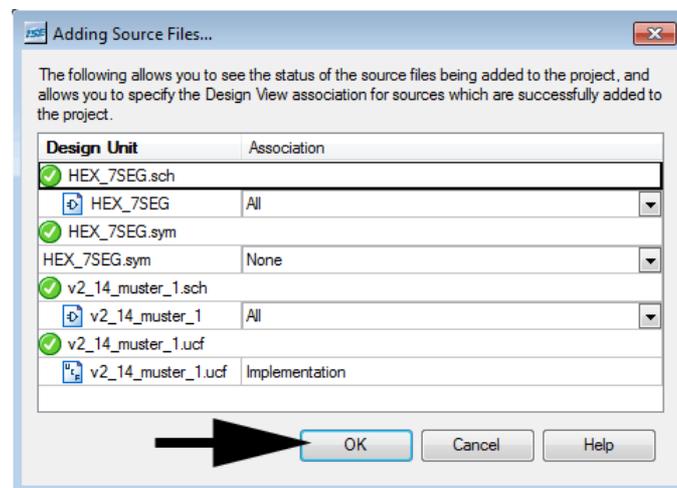
**Abb. 3** Der erste Versuchsaufbau. CPLD-Lehrgerät 12 und Siebensegmentanzeige 09/13. Wir nutzen zunächst nur zwei Anzeigemodule. Das linke (L) an Port D, das rechte (R) an Port C anschließen. Die Abbildung zeigt auch, wie die Tasten des Pollin-Platine in nachfolgenden Aufgaben genutzt werden, um den Schnittstellentester zu bedienen.



**Abb. 4** Projekt anlegen. Aber ohne Quellen. Dann Quellen importieren (Add Copy of Source). Zu den Handgriffen s. die Anleitung des Versuchs 1.



**Abb. 5** So werden die Quellen importiert. Zum Verzeichnis (auf der Netzfestplatte) s. die Anleitung des Versuchs 1. Wie das Unterverzeichnis heißen könnte, liegt nahe (z. B. "V2\_Vorlage"). Wir brauchen den Schaltplan, den Siebensegmentdecoder und die (vorgefertigte) Constraintsdatei. Hierzu müssen wir **alle** Dateien zu sehen bekommen (Pfeil).



**Abb. 6** So müssen die Dateien ankommen. OK.

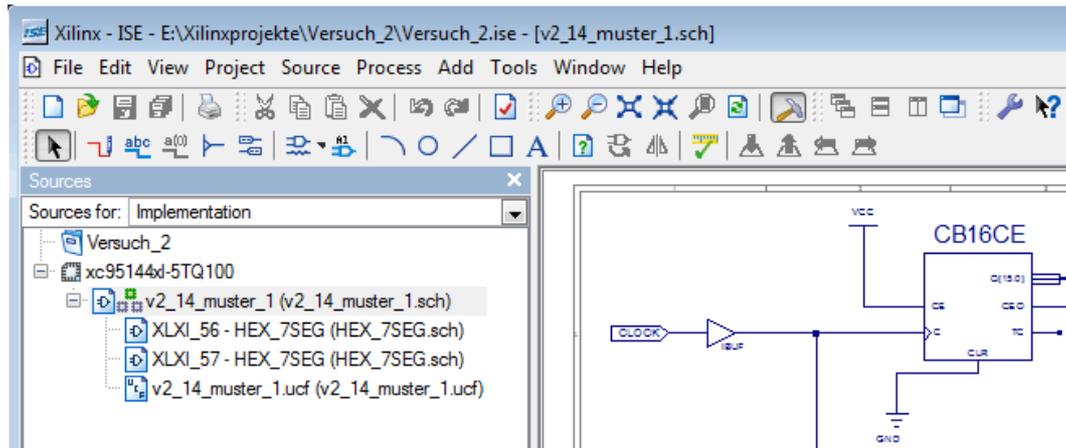


Abb. 7 Unser Projekt muß aus diesen Quellen bestehen.

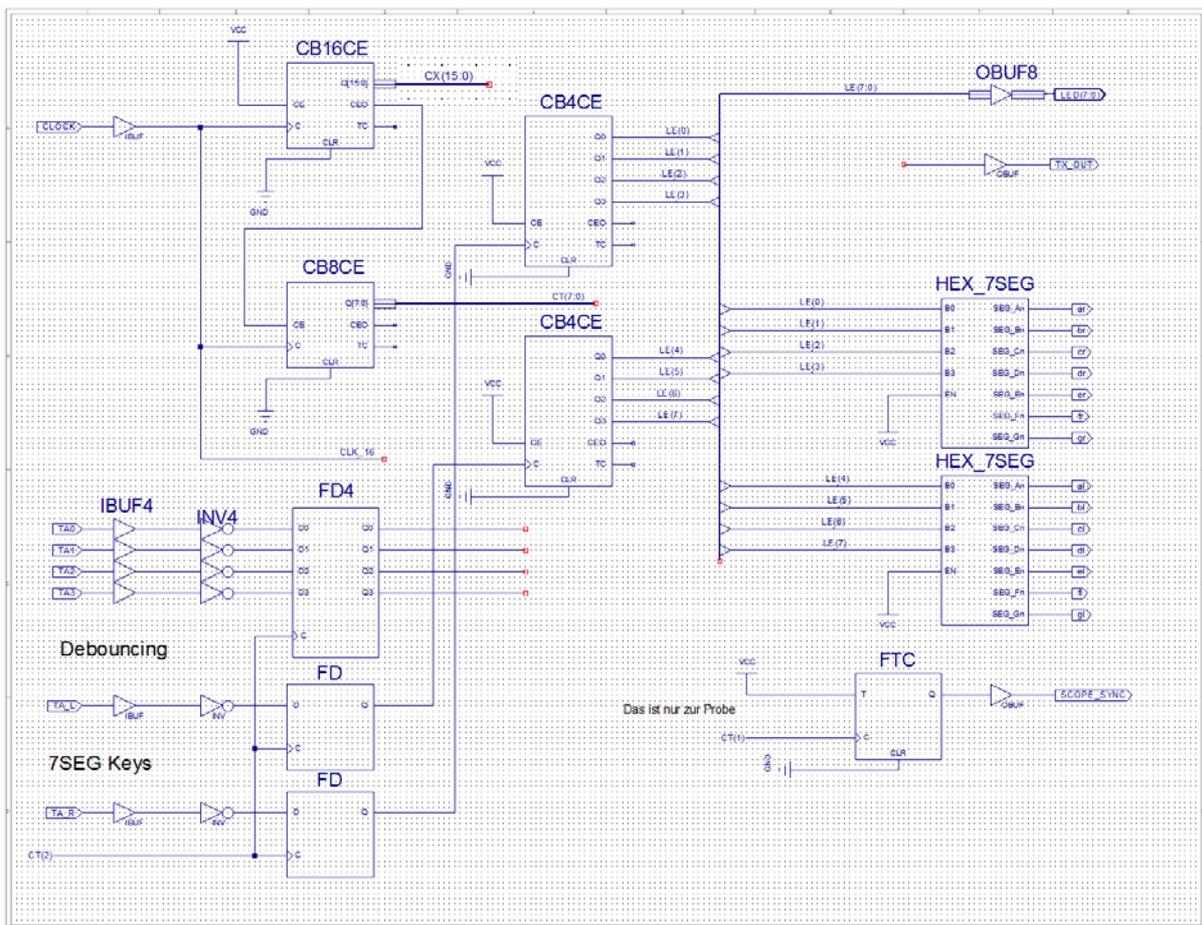


Abb. 8 Der ursprüngliche Schaltplan. Die Signalliste der Constraintsdatei finden Sie auf S. 33. Implementieren und programmieren. Beim Betätigen der Tasten unter den Siebensegmentanzeigen muß die jeweils zugehörige Anzeige zyklisch vorwärts zählen.

### Aufgabe 2: Schnittstellentester mit Dauerbetrieb

Die Aufgabe des Schnittstellentesters besteht darin, ein einziges Prüfzeichen zu senden. Der Zeichencode wird hexadezimal eingestellt (siehe Codetabelle (S. 9)). Die Übertragungsparameter: 8 Datenbits, kein Paritätsbit, 1 Stopbit (8, N, 1). Datenrate: 2400 Bits/s. Das gesamte Prüfmuster ist 16 Bits lang:

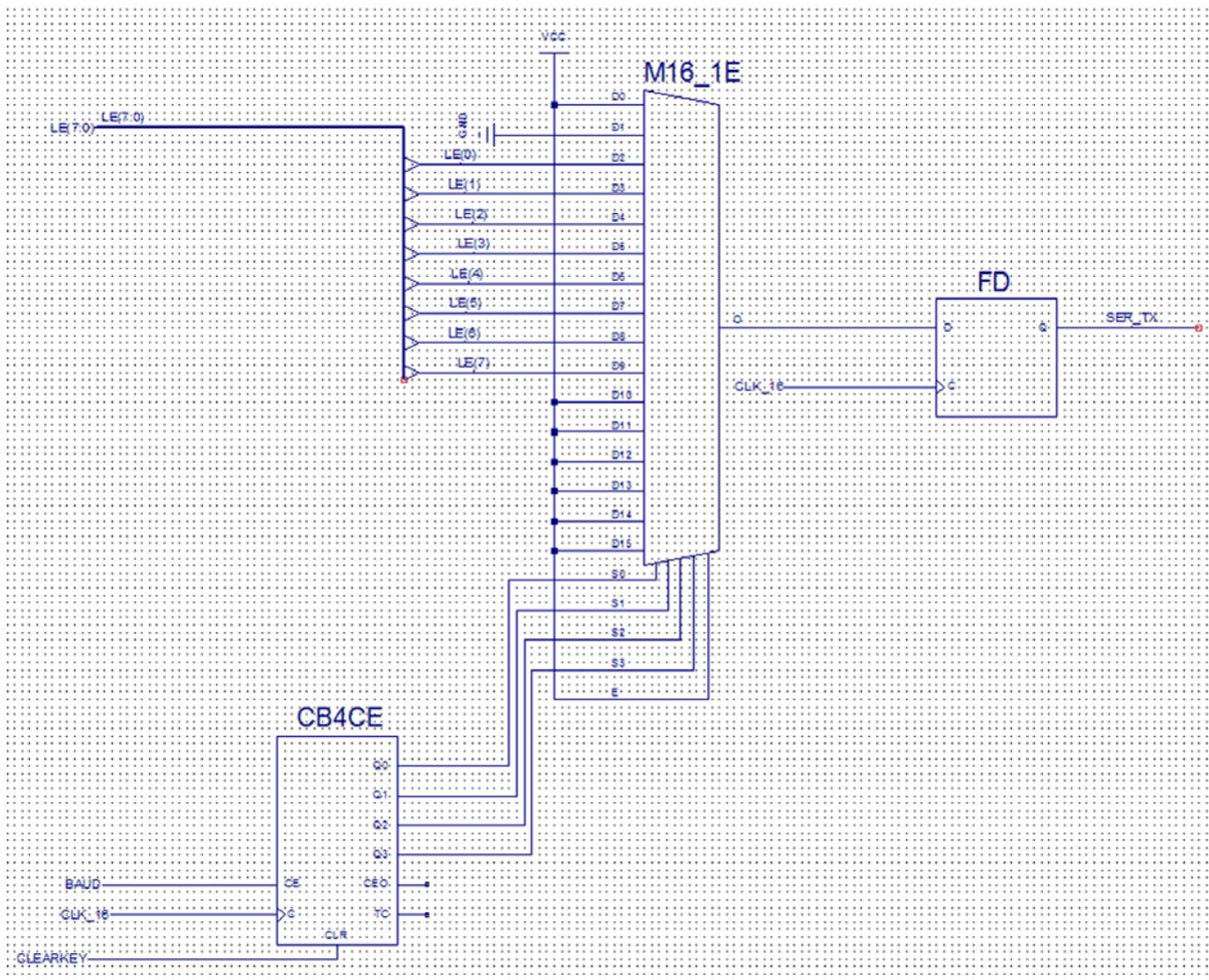
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Start	Prüfzeichen (ASCII)								Stopbit + Ruhezustand							
0	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	1	1	1	1	1	1	1	

Das Prüfzeichen (ASCII):

Einstellzähler 2 (links)				Einstellzähler 1 (rechts)			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Grundschtaltung: Das zu sendende Bitmuster wird an einen Multiplexer angelegt und zyklisch abgefragt (Abfragezähler). Wir haben aber keine 16 Kippschalter. Deshalb machen wir nur die acht Bitpositionen des eigentlichen Prüfzeichens veränderlich und beschalten diese mit zwei Einstellzählern zu je vier Bits, deren Ausgänge auf die Siebensegmentanzeigen geführt und deren Takteingänge von zwei Tasten angesteuert werden. Die verbleibenden Bitpositionen werden gemäß Prüfmuster fest beschaltet – und zwar so, daß im Ruhezustand (z. B. Stellung 0 des Abfragezählers) eine Eins abgegeben wird.

Den Schaltplan um zwei Blätter erweitern (wie das geht, steht auf Seite 21).



**Abb. 9** Eine der Lösungen, um ein beliebiges serielles Bitmuster herzustellen (Grundschtaltung eines Bitmustergenerators mit Multiplexer und Abfragezähler). Der Eingang des Output-Puffers TX\_OUT (Blatt 1) muß als SER\_TX benannt werden. Zum Umbenennen von Signalen siehe Seite 23.

Wir müssen mit 2400 Bits/s ausgeben. Unser Takt ist aber 16 MHz.  $16\,000\,000 : 2400 = 6666,66\dots$

Also teilen wir die 16 MHz durch 6667. Hierzu können wir uns einen neuen 16-Bit-Zähler leisten. Er muß einen synchronen Rücksetzeingang haben (nicht CLR = Clear, sondern R = Reset). Er zählt von Null bis 6666 und wird mit dem 6667. Takt wieder zurückgesetzt. In diesem Taktzyklus wird gleichzeitig der Abfragezähler am Multiplexer weitergezählt.

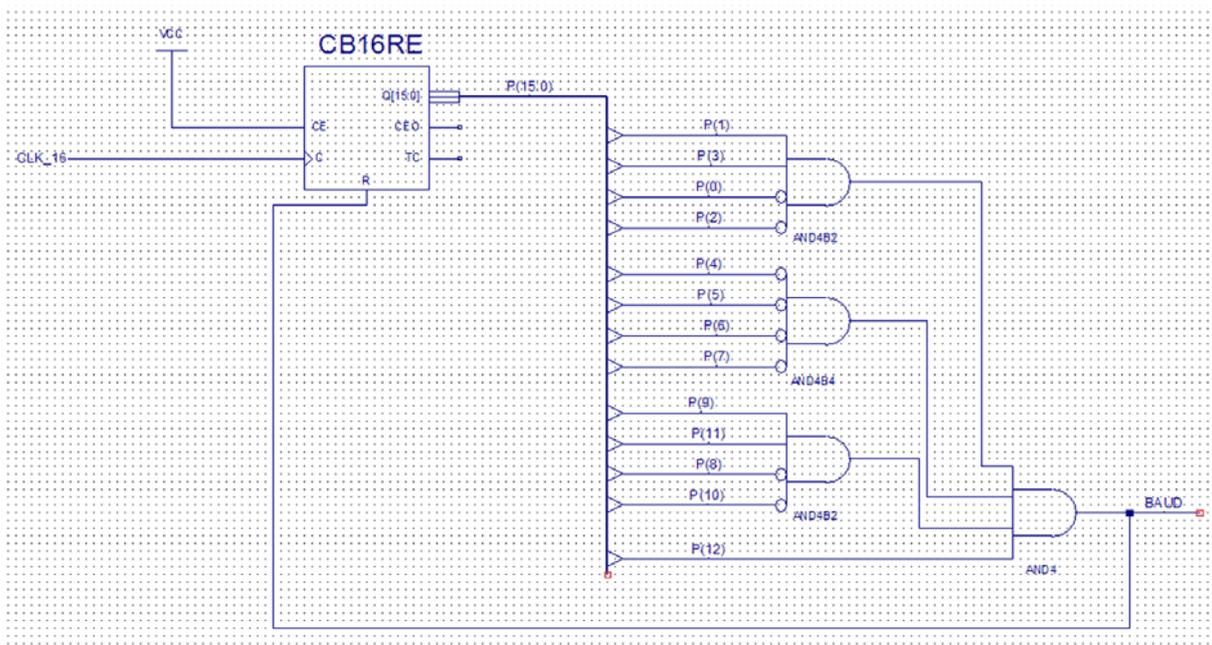
6666 ist binär 1 1010 0000 1010.

Wir brauchen also ein UND-Gatter mit folgender Beschaltung:

P(13)	P(12)	P(11)	P(10)	P(9)	P(8)	P(7)	P(6)	P(5)	P(4)	P(3)	P(2)	P(1)
1	1	0	1	0	0	0	0	0	1	0	1	0

Wenn man sich passende UND-Gatter (mit invertierten Eingängen) aussucht, braucht man keine Negatoren.

Wie das Schnittstellensignal aussieht? Wer sich das ansehen möchte, muß das Oszilloskop verwenden. Zum Anschließen des Tastkopfs s. Seite 10 oder 24. Signalverlaufsbeispiele auf Seite 24f.



**Abb. 10** Das Signal BAUD wird rund 2400mal je Sekunde einen Taktzyklus lang aktiv. Hierzu dient diese Zähschaltung.

Erprobung:

1. Serielle Schnittstelle an die Pollin-Platine anschließen (Kabel mit 9poligem D-Sub-Stecker vom PC).
2. Terminal aufrufen (Windows-Desktop). Es ein ganz einfaches Programm, an dem nichts einzustellen ist.
3. Den COM-Port auswählen. Es ist entweder COM-Port 1 oder COM-Port 2. Wenn es nicht klappt, mit dem jeweils anderen COM-Port probieren oder das andere Schnittstellenkabel anschließen. S. auch Abb. 15 (S. 10).
4. An der Siebensegmenanzeige einen passenden ASCII-Code einstellen, beispielsweise 41H = "A".

Die Bildschirmfläche des Terminals muß mit dem gewählten Zeichen vollgeschrieben werden. Zum Löschen des Bildschirms mit ESC beenden und wieder von vorn anfangen.

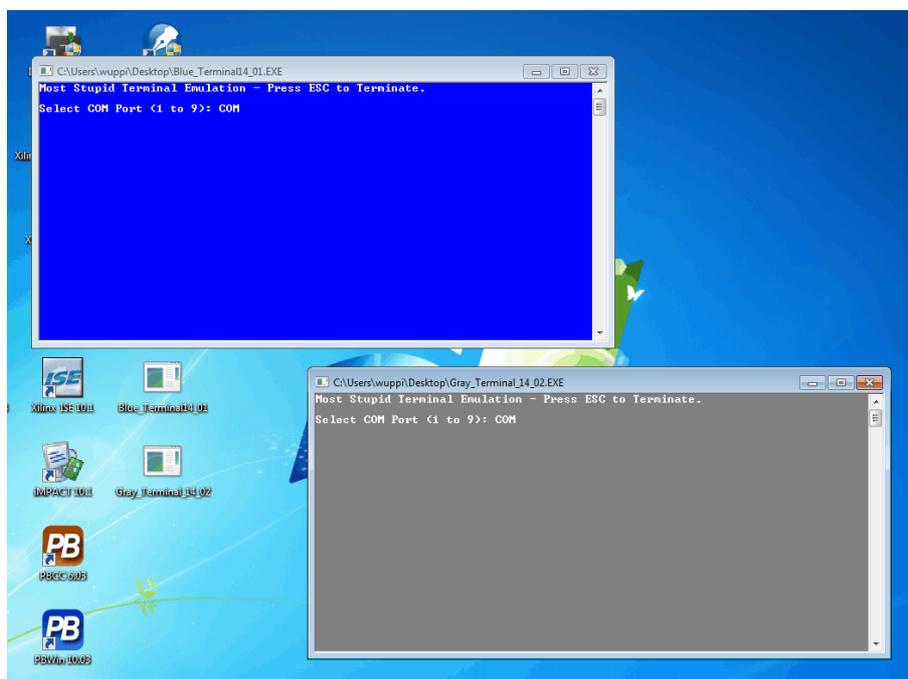
Es kann sein, daß es nicht richtig klappt. Ggf. die Taste 3 der Pollin-Platine (CLEARKEY) kurzzeitig betätigen.

Mit verschiedenen Zeichen probieren.

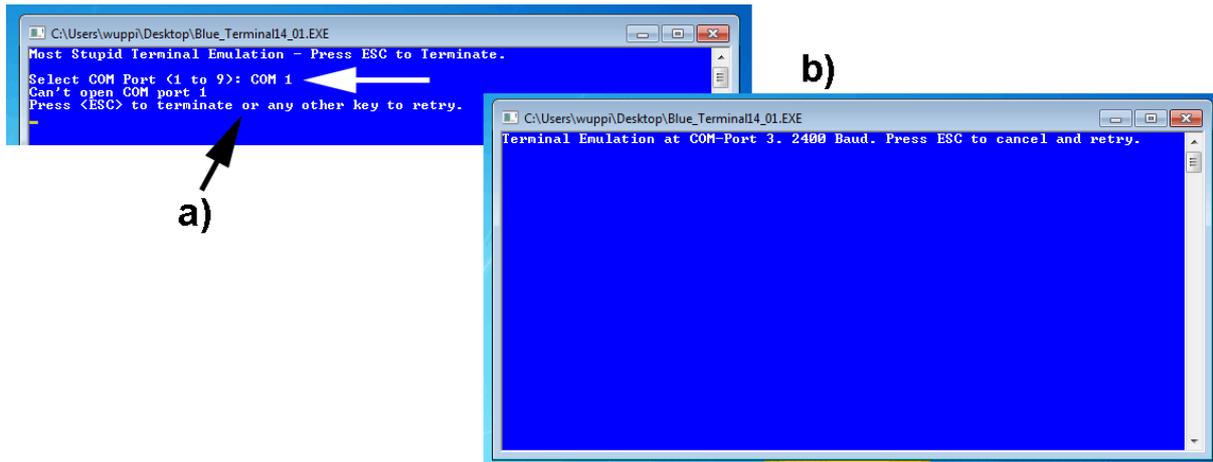
Es ist keine perfekte Lösung, denn wenn der Empfänger im PC nicht genau auf unsere Start- und Stopbits trifft, zeigt er falsche Zeichen an oder bleibt gar hängen. Das kann schon dann passieren, wenn wir mitten in der Übertragung ein neues Zeichen einstellen.



**Abb. 11** Ein Terminalprogramm auswählen.



**Abb. 12** Die Wahl ist reine Geschmackssache. Die Nutzung erklärt sich von selbst.



**Abb. 13** Zur Auswahl des COM-Ports die Nummer (1 oder 2) eingeben (Pfeil).  
 a) erscheint, wenn es nicht klappt, b) wenn alles in Ordnung ist.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20:	!	"	#	\$	%	&	'	<	>	*	+	,	-	.	/	
30:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70:	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

**Abb. 14** Der ASCII-Code (Codetabelle). Beispiel: Prüfzeichen "1" = 31H, Prüfzeichen "A" = 41H.

*Eine kleine Fehlersuchhilfe*

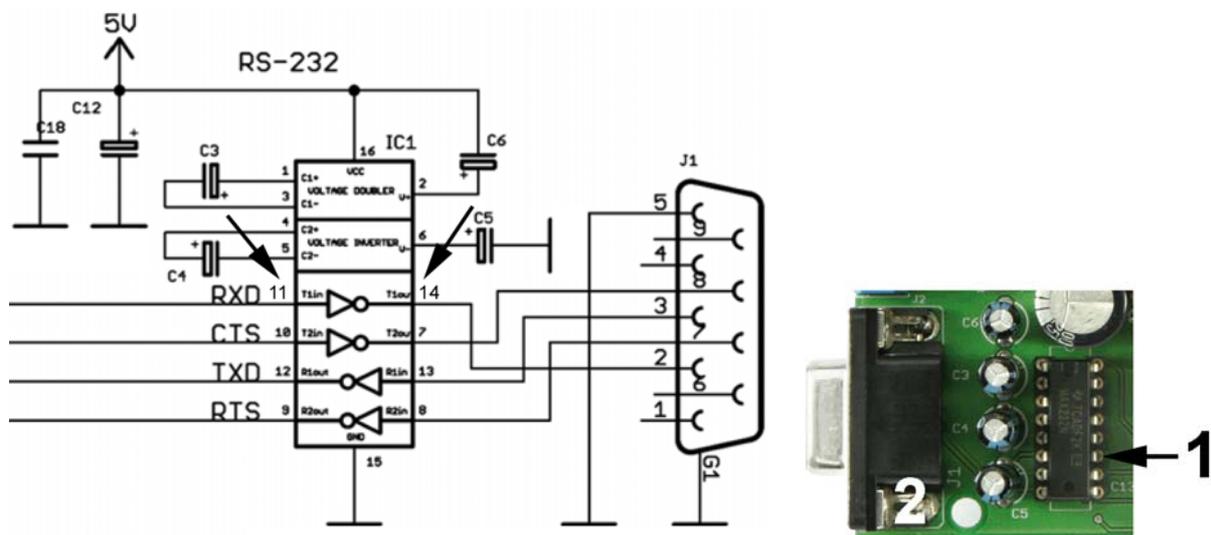
Wenn im Terminalfenster nichts zu sehen ist:

- Richtiges Schnittstellenkabel angesteckt?
- Eingang des OBUFs TX\_OUT (Blatt 1) richtig benannt?
- Ggf. nachmessen (Oszilloskop). Signalverlaufsbeispiele s. Seite 24f.

Achtung: Beim Umbenennen von Signalen ggf. aufpassen. Siehe Seite 23.



**Abb. 15** Ein Praxistip zwischendurch: Welches Kabel gehört zu welchem COM-Port? Das kann man mit einem Kurzschlußstecker experimentell herausfinden. Die Signale TX und RX (Pins 2 und 3) sind hier miteinander verbunden (echte Profis schaffen es auch mit einem Schraubenzieher oder einer aufgebogenen Büroklammer...). Gehört das Kabel zum ausgewählten COM-Port, werden Tastatureingaben auf dem Terminalfenster angezeigt (Echofunktion). Kurzschlußstecker kann bei Bedarf ausgeliehen werden.

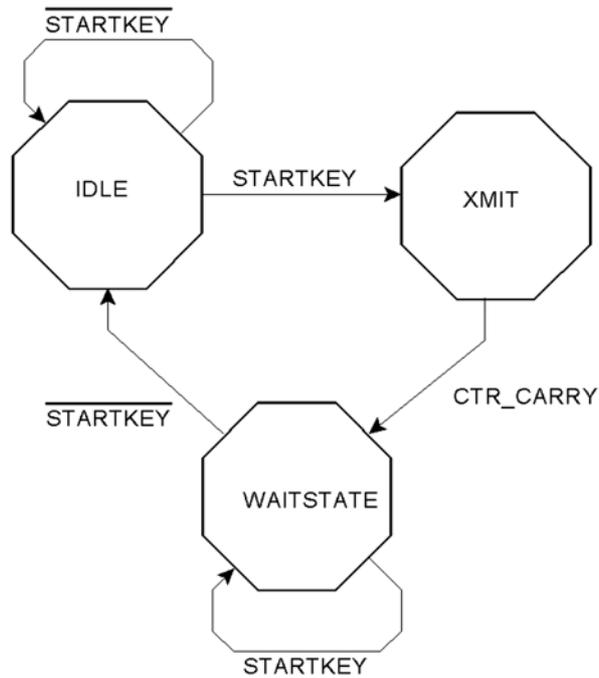


**Abb. 16** Hier messen. An Pin 11 des Schnittstellentreibers kommt das Signal TX\_OUT an. 1 - Tastkopfspitze; 2 - Masseklemme.

### Aufgabe 3: Schnittstellentester mit Einzelauslösung

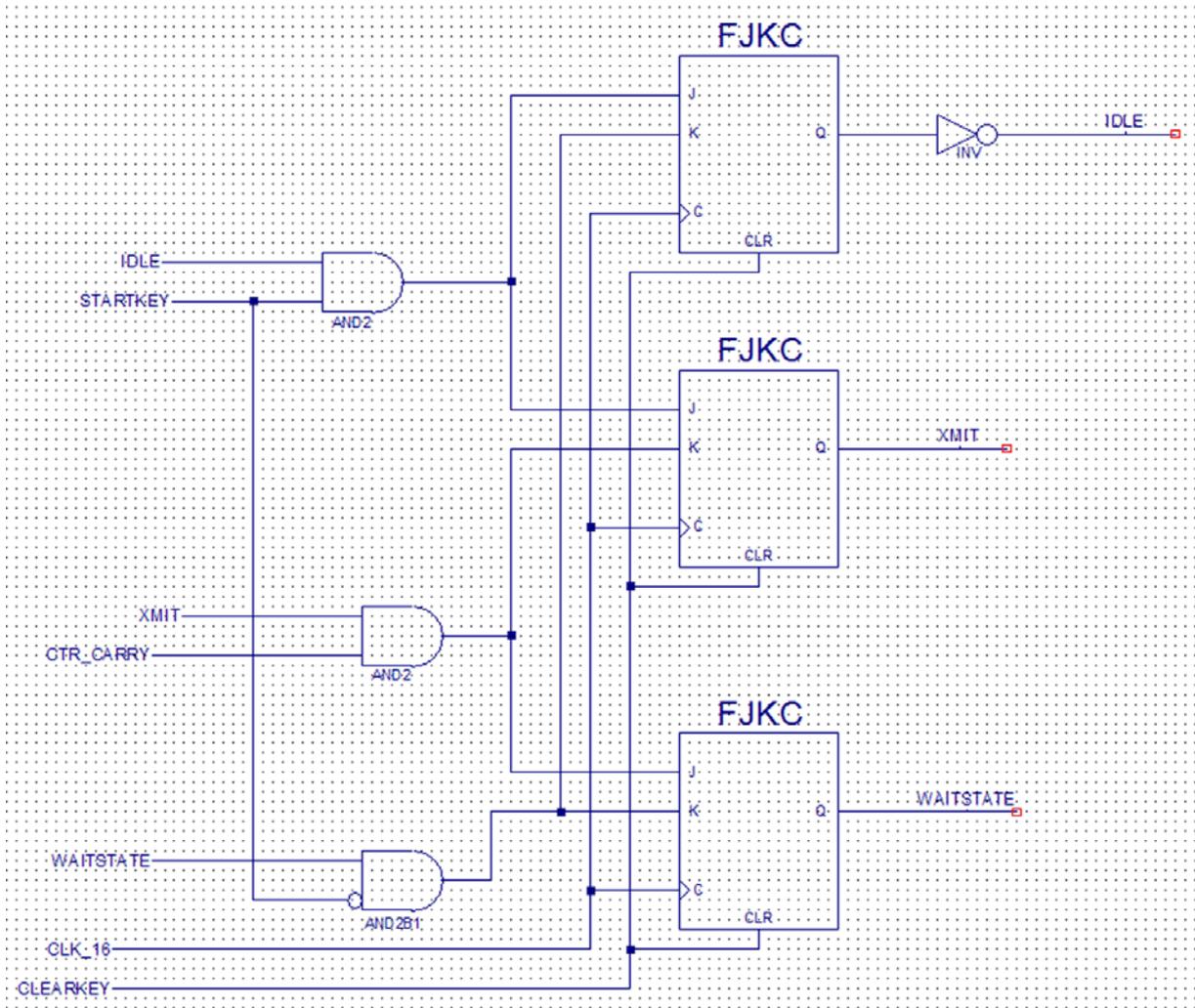
Jetzt soll nur ein einziges Zeichen gesendet werden. Bedienprinzip: zuerst einen Zeichencode einstellen, dann das Zeichen senden. Zum Auslösen des Sendens nehmen wir die Taste 0 auf der Pollin-Platine. Damit wirklich nur ein einziges Zeichen gesendet wird, müssen wir einen Zustandsautomaten bauen. Es ist eine Grundschaltung. Im Fach-Englisch heißt sie "Single Shot Generator".

Da wir genügend Flipflops haben, können wir lehrbuchmäßig im 1-aus-n-Code (OHE) entwerfen. Wir brauchen ein weiteres Blatt für den Schaltplan.

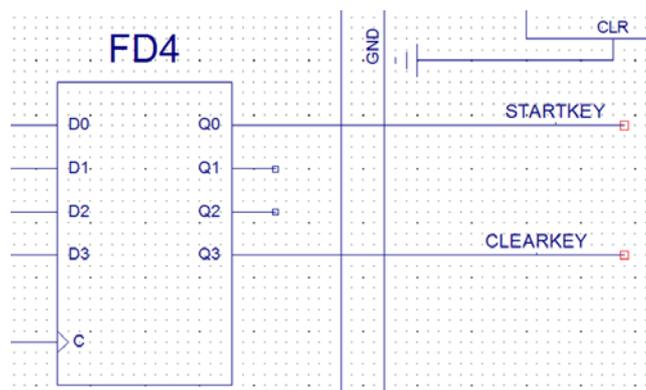


**Abb. 17** Das Zustandsdiagramm. STARTKEY = Pollin-Taste 0, CTR\_CARRY = Ausgangsübertrag (CEO) des Abfragezählers.

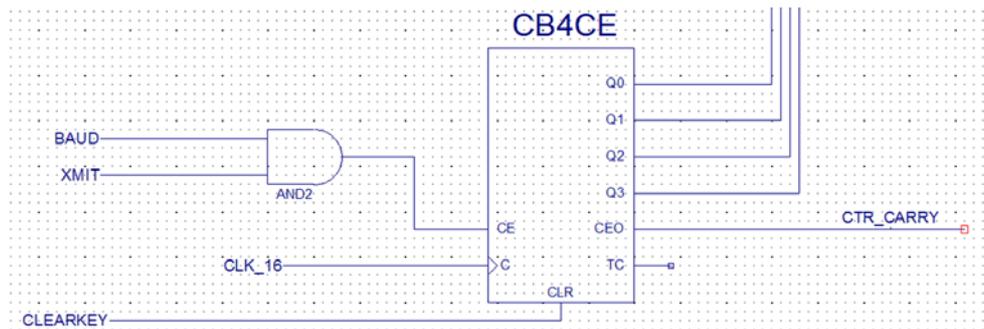
Die einfachste Implementierung ist mit RS-Flipflops. Setzen, wenn alter Zustand und Ereignis; löschen, wenn aktueller Zustand und Wechsel zu einem neuen. Hierzu verwenden wir JK-Flipflops. Am Anfang muß sich der Apparat im Ruhezustand befinden. Man könnte JKs mit Setz- und Rücksetzeingängen nehmen. Bequemer ist es aber, sich auf den Anfangszustand Null zu verlassen, denn der wird nach dem Programmieren oder dem Einschalten des CPLDs immer von selbst eingestellt. Also Flipflop anfänglich auf Null und dafür negiert verwenden (Negator am Ausgang, J = Rücksetzen, K = Setzen).



**Abb. 18** Der Zustandsautomat. Der Sendezustand XMIT wird erst dann, verlassen, nachdem der Abfragezähler durchgelaufen ist (CTR\_CARRY).



**Abb. 19** Tastensignale umbenennen.

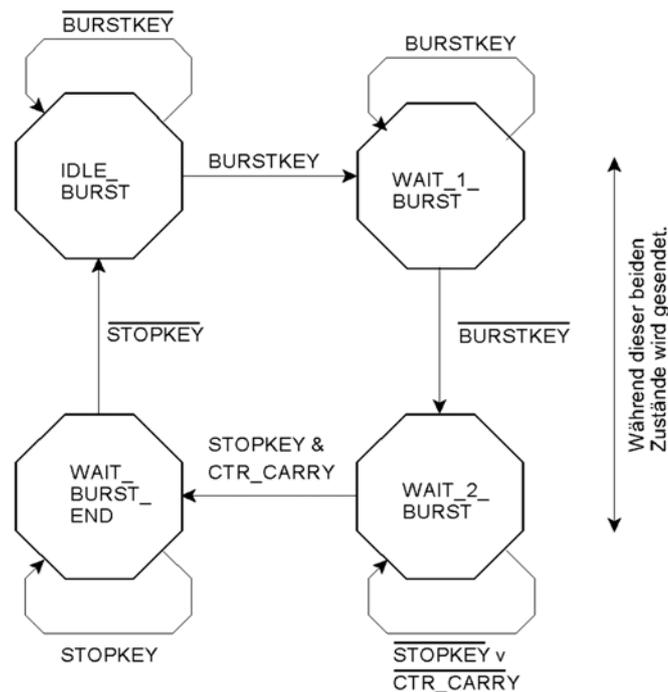


**Abb. 20** Änderungen am Abfragezähler. 1. Er darf nur dann zählen, wenn der Sendezustand XMIT aktiv ist. Hierzu brauchen wir eine UND-Verknüpfung. 2. Das Ende des Zählerdurchlaufs bewirkt einen Zustandswechsel. Hierzu wird der Ausgangsübertrag angezapft.

Nun die Schaltung erproben. Es darf immer nur ein Zeichen gesendet werden. Und es muß das sein, was wir eingestellt haben (siehe Codetabelle S. 9).

#### Aufgabe 4: Schnittstellentester mit Dauer- und Einzelbetrieb

Jetzt sollen noch zwei Tasten hinzukommen. Wir nehmen die auf der Pollin-Platine. Taste 2 löst ein dauerndes Senden des eingestellten Prüfzeichens aus, Taste 1 hält das Senden an. Diese Übertragungsweise soll als Burst-Übertragung bezeichnet werden. Das Senden einzelner Zeichen über Taste 0 soll weiterhin möglich sein. Wir brauchen noch einen Zustandsautomaten – und dafür ein neues Blatt im Schaltplan.



**Abb. 21** Der neue Zustandsautomat. Es sind zwei Tasten. Das Drücken der ersten (BURSTKEY) löst das Senden aus. Nun muß darauf gewartet werden, bis sie wieder losgelassen wird. Dann wird aber immer noch gesendet, und zwar solange, bis die zweite Taste (STOPKEY) betätigt wird und der Abfragezähler durchgelaufen ist. Jetzt ist noch zu warten, bis auch diese Taste losgelassen wird.

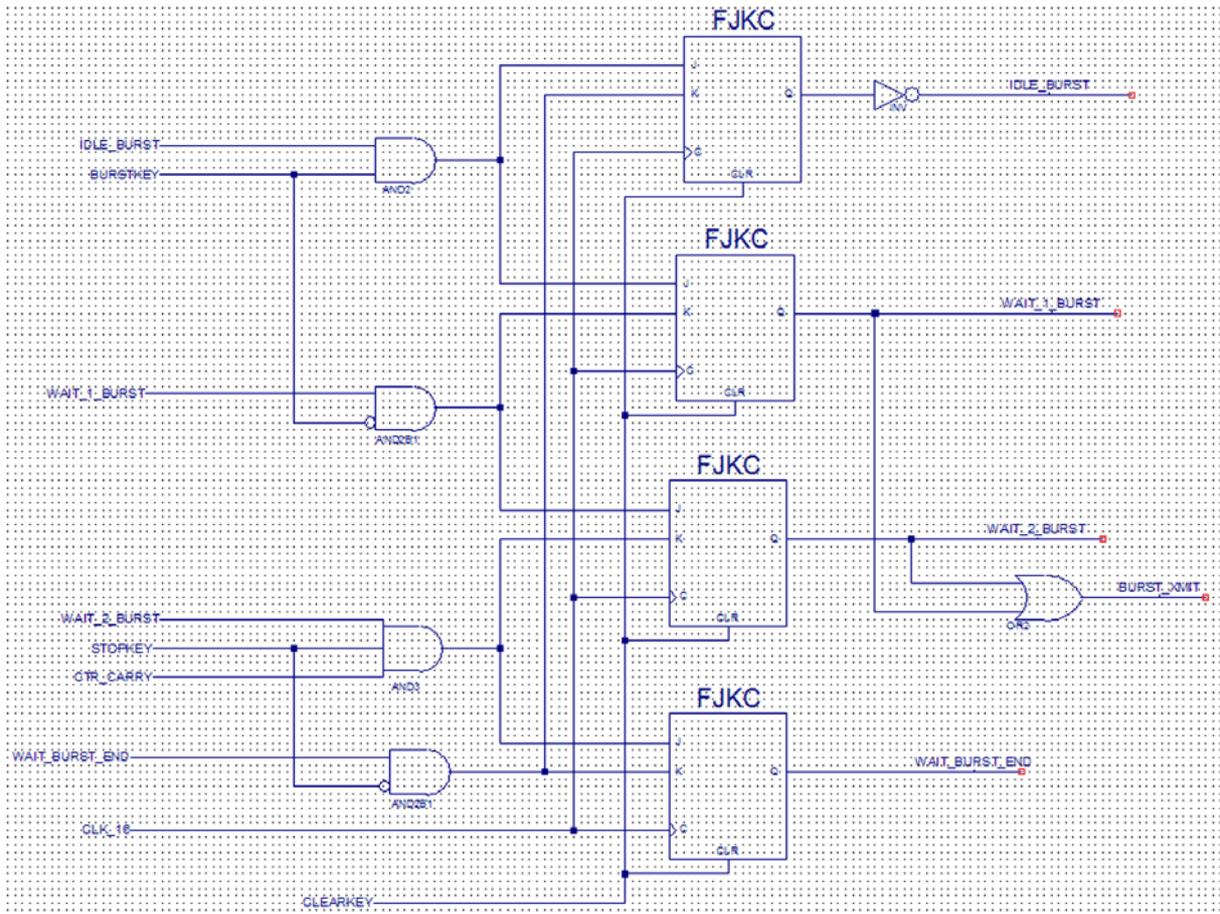


Abb. 22 Der zweite Zustandsautomat.

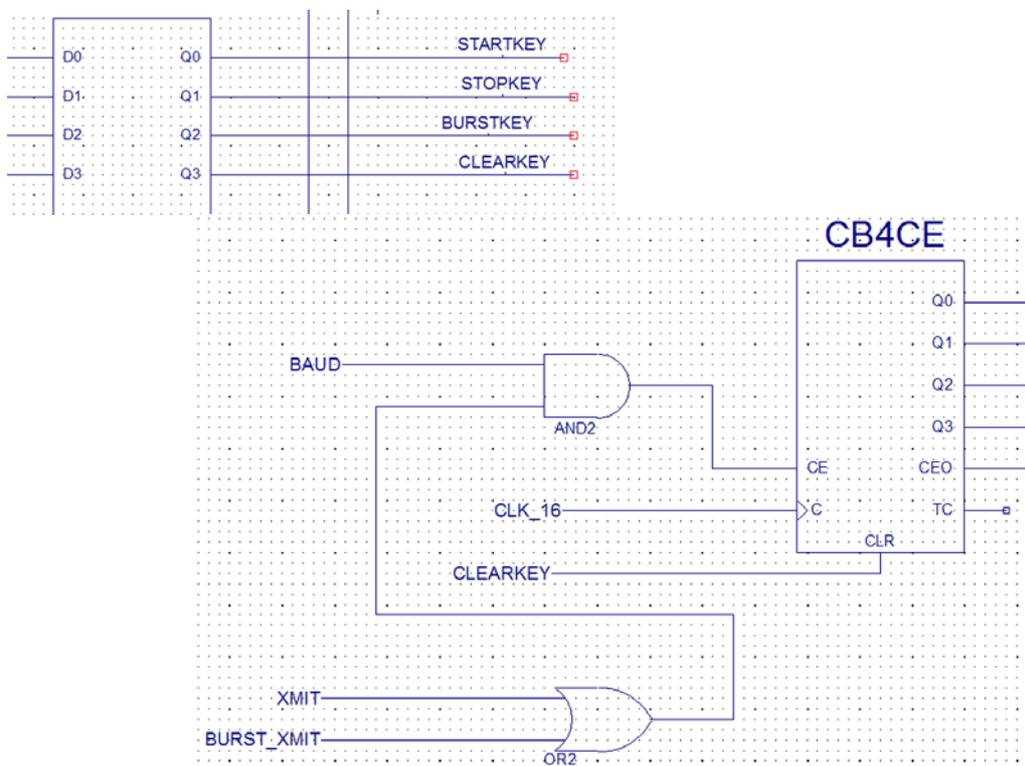
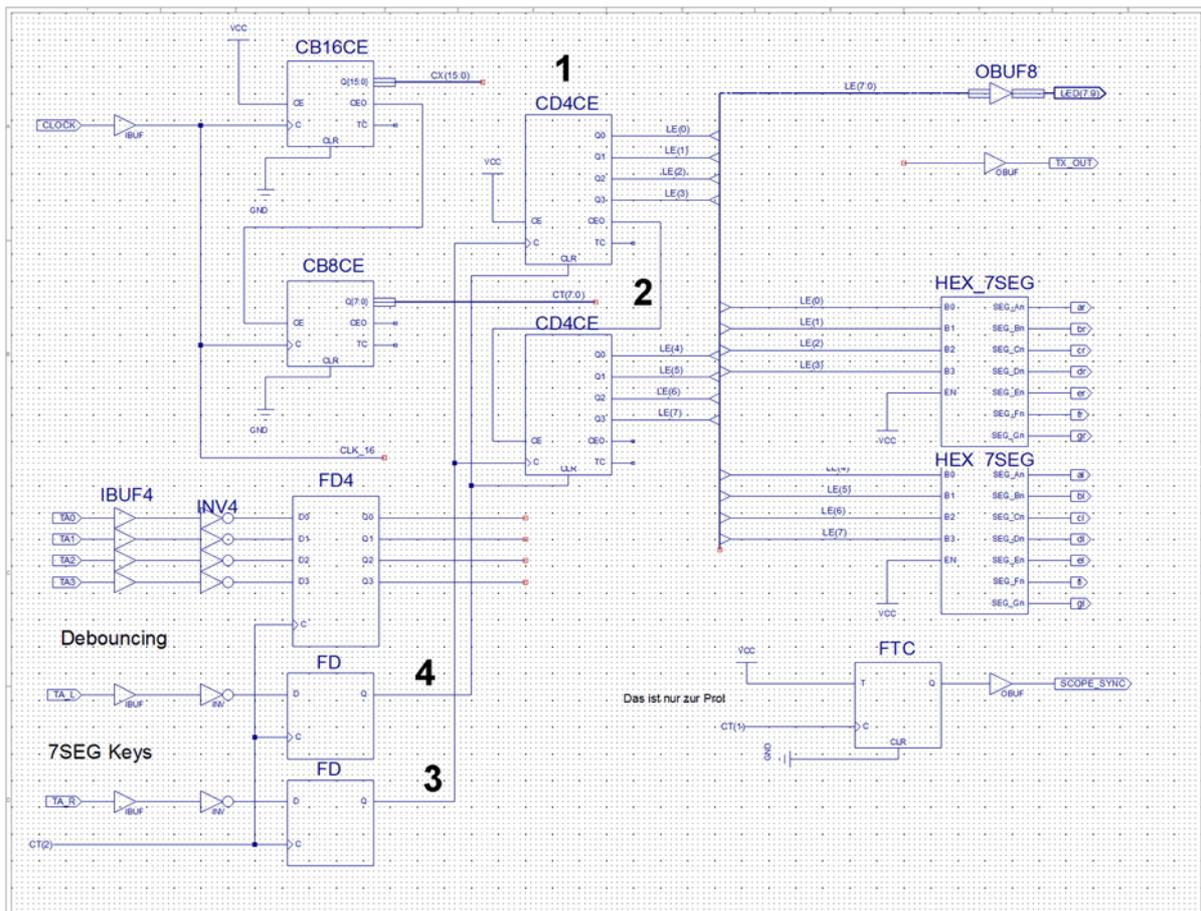


Abb. 23 Änderungen, Änderungen...

## Aufgabe 5: Handzähler

Handzähler gibt es als mechanische und elektronische Geräte. Sie zählen eigentlich nur Tastenbetätigungen. Jede Betätigung (Drücken und wieder Loslassen) erhöht den Zählerstand um 1. So kann man Autos, Schafe, BVB-Fans usw. zählen. Es ist eine der einfachsten Zählaufgaben. Unser Zähler ist zweistellig. Die Taste unter der rechten Siebensegmentanzeige dient zum Zählen, die unter der linken zum Löschen.

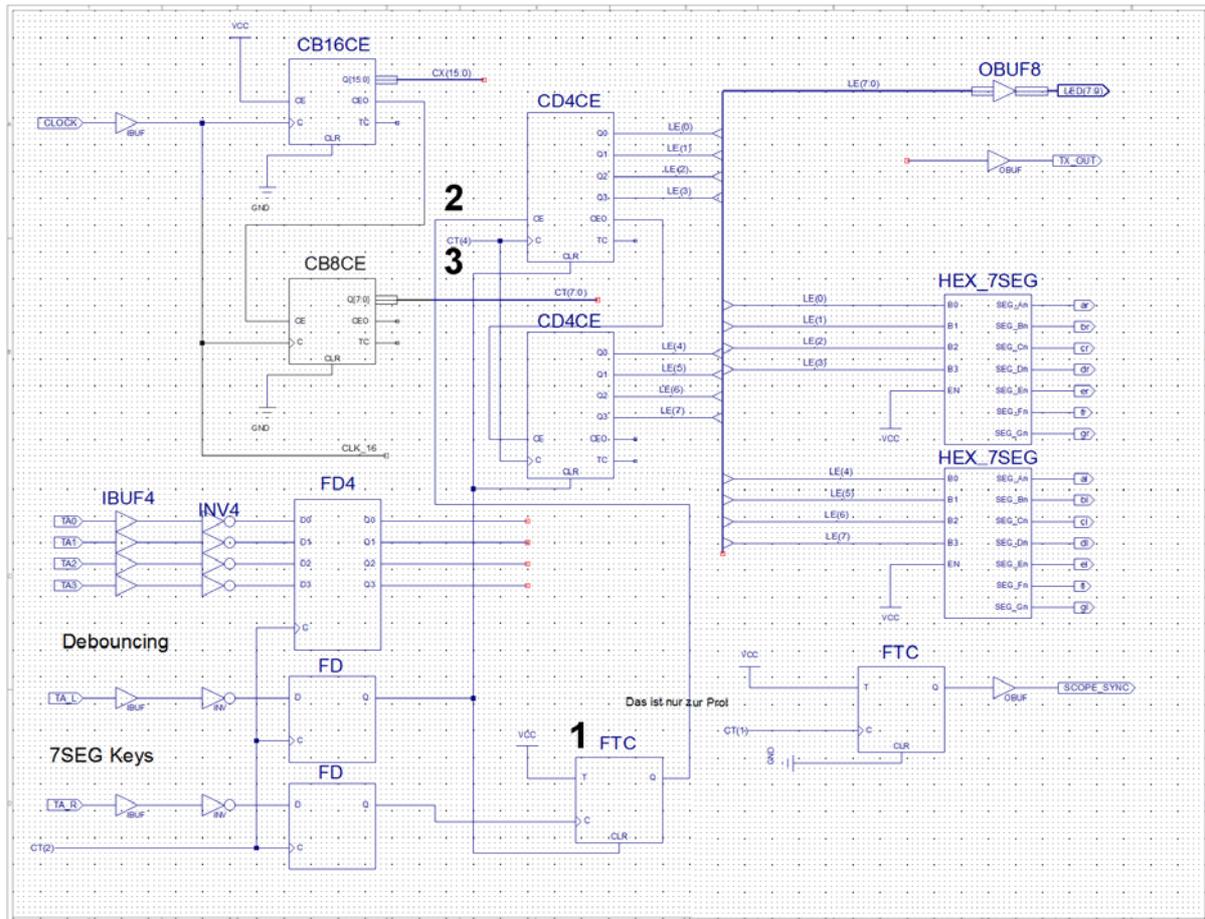
Wir kommen mit dem ersten Blatt des Schaltplans aus. Den Inhalt der weiteren Blätter können wir löschen. Die leeren Blätter werden später wieder gebraucht.



**Abb. 24** Der Handzähler. 1 - die Binärzähler gegen Dezimalzähler auswechseln. 2 - Zähler in Reihe schalten (CEO mit CE verbinden). 3 - Zähltaste an beide Takteingänge anschließen. 4 - Löschtaste an beide Löscheingänge anschließen. An der Constraintsdatei ist nichts zu ändern. Implementieren, programmieren, ausprobieren...

## Aufgabe 6: Stoppuhr

Die dezimale Zählerschaltung wird vom Handzähler zur Stoppuhr umgebaut. Die Stoppuhr soll im Raster von 0,1 s bis 9,9 s zählen. Betätigung: rechte Siebensegmenttaste = Umschaltung Zählen/Stop (Toggle-Funktion), linke Siebensegmenttaste = löschen. Das Zählen wird von einem Toggle-Flipflop gesteuert, das dem Zählerlaubniseingang der niederwertigen Zählstelle vorgeschaltet ist. Als Zähltakt wählen wir zunächst (zum Ausprobieren) CT(3) oder CT(4). CT(3) ist das Ausgangssignal der 20. Binärzählerstelle (16 Stellen des ersten + 4 Stellen des zweiten Zählers), CT(4) das Ausgangssignal der 21. Stelle. Mit CT(3) werden die 16 MHz durch  $2^{20}$  geteilt (15,26 Hz), mit CT(4) durch  $2^{21}$  (7,63 Hz). Das ergibt eine Taktperiode der Stoppuhr von 65 oder 130 ms.



**Abb. 25** Die Stoppuhr. Grundschtung zur Funktionserprobung. 1 - Toggle-Flipflop einbauen. 2 - Toggle-Flipflop an den Zählerlaubnis Eingang anschließen. 3 - Takt z. B. an CT(4) anschließen (beide Zähler). Ausprobieren...

Die Grundschtung ist schnell aufgebaut, hat aber zwei auffallende Mängel:

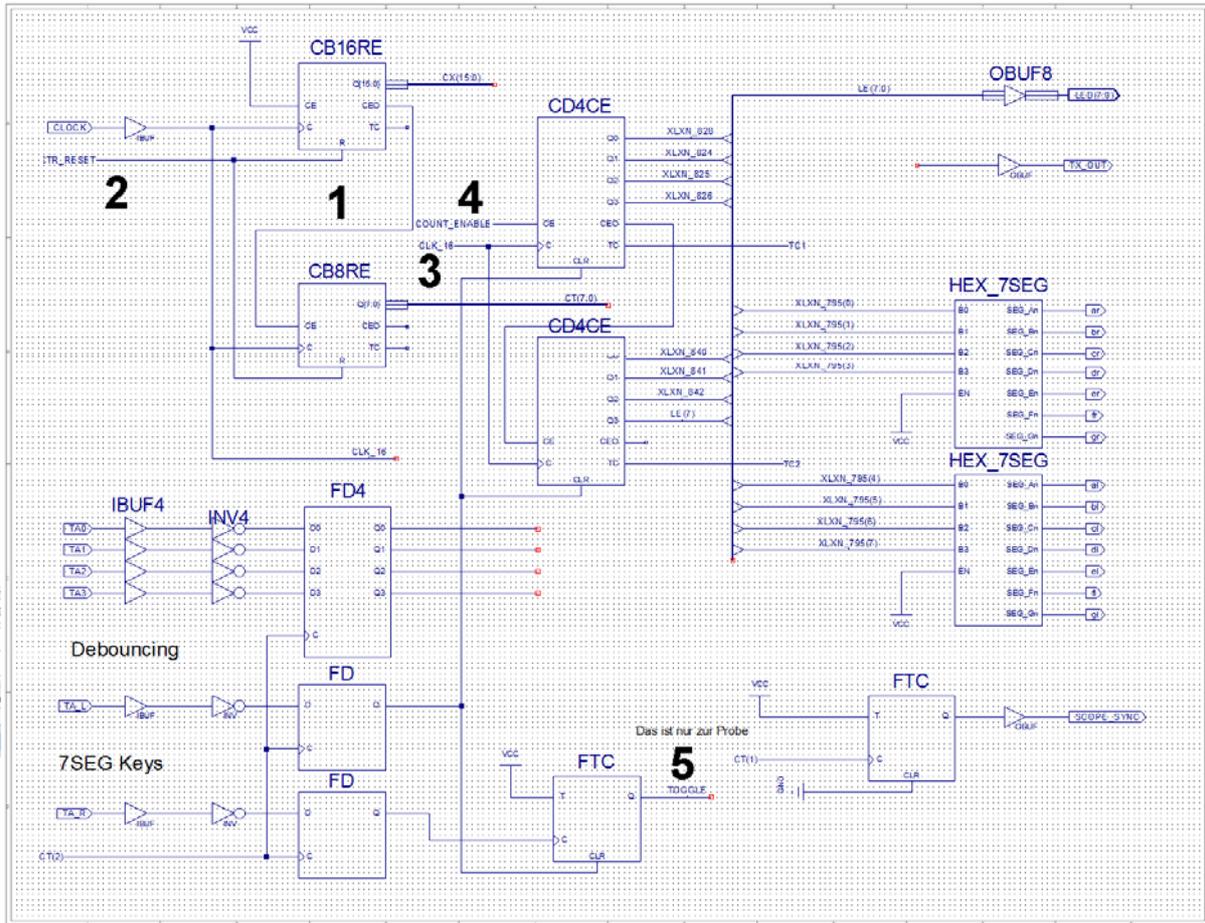
1. Sie zählt nicht genau im Rhythmus einer Zehntelsekunde, würde also zur Zeitmessung nicht zugelassen werden.
2. Sie hält nicht bei 9,9 s an.

Um den ersten Mangel zu beseitigen, müssen wir die 16 MHz exakt durch 1 600 000 teilen. Hierfür ist die Zählerstellung 1 599 999 zu decodieren und damit der Zähler zurückzusetzen. Damit das richtig funktioniert, müssen Zähler mit synchronem Rücksetzen eingesetzt werden.

$$1\ 599\ 999 = 1\ 1000\ 0110\ 1001\ 1111\ 1111B.$$

Wir brauchen also ein UND-Gatter mit 21 Eingängen. Dessen Ausgang wird mit den Rücksetzeingängen der beiden Frequenzzähler verbunden. Das Ausgangssignal liefert zugleich die Zählerlaubnis für die Dezimalzähler.

Um den zweiten Mangel zu beseitigen, ist die Endstellung der Dezimalzählung (99 = 1001 1001B) zu decodieren und damit das Weiterzählen über den Zählerlaubnis Eingang verhindern. Die Decodierung ist schon in die Zähler eingebaut (Ausgänge TC).



**Abb. 26** Die Zählschaltungen sind umzubauen. 1 - die Zähler der Frequenzteilung werden gegen Typen mit synchronem Rücksetzen ausgewechselt. 2 - das neue Rücksetzsignal. 3 - die Zähler der Stoppuhr werden an den 16-MHz-Takt angeschlossen. 4 - ein neues Zählerlaubnissignal. 5 - das Umschaltsignal des Toggle-Flipflops. Die zusätzlichen neuen Schaltungen kommen auf ein weiteres Blatt.

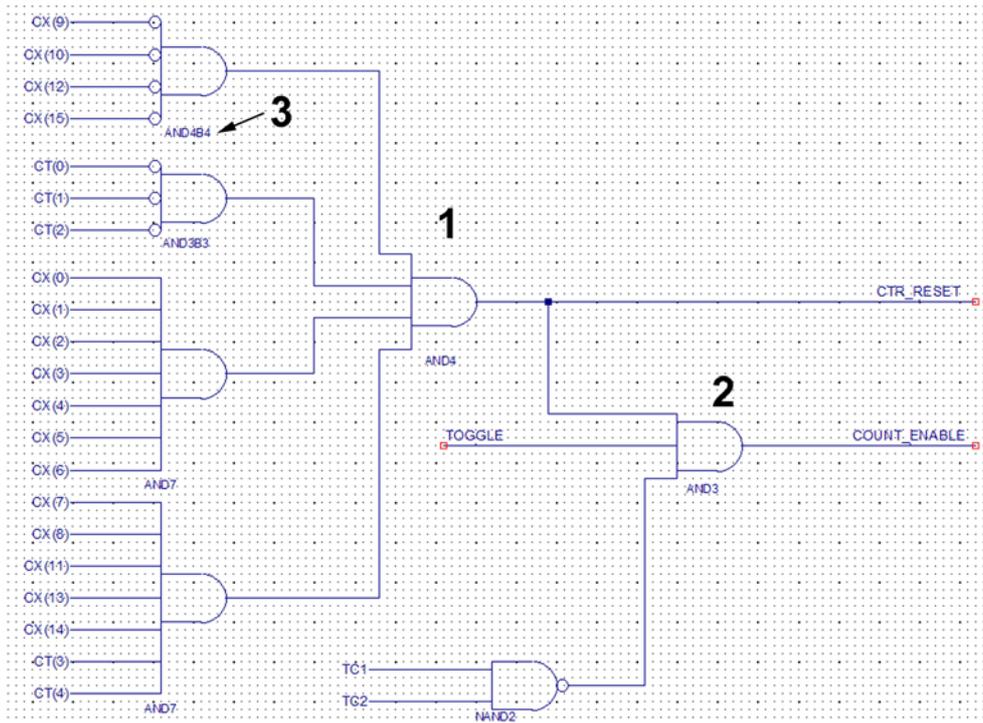
Den Bus des 16-Bit-Zählers heißt CX(15:0). Es sind sieben Nullen und 14 Einsen zu decodieren. Damit wir nicht so viel zu zeichnen haben, bauen wir das große UND-Gatter aus Gattern mit negierten und mit nichtnegierten Eingängen auf. Wir brauchen sieben negierende und 14 nichtnegierende Eingänge.

Die zu decodierende Binärzahl:

CT(...)					CX(...)															
4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1

Zählerlaubnis: Die Stoppuhr soll zählen,

- wenn der Frequenzteiler durch Null geht (also der Decoder aktiv wird) UND
- wenn das Toggle-Flipflop aktiv ist UND
- wenn die Endstellung (99) NICHT erreicht ist, wenn also TC1 und TC2 nicht beide zugleich Eins sind (NAND-Verknüpfung).



**Abb. 27** Die zusätzlichen Schaltungen. 1 - Decoder. 2 - Zählerlaubnis. 3 - so kommen wir zu den UND-Gattern mit negierten Eingängen. Diese Funktionselemente haben ein B in der Bezeichnung. AND3B1 heißt: drei Eingänge, davon einer negiert. Ein Gatter AND4B4 hat demzufolge vier Eingänge, die alle negiert sind.

### Aufgabe 7: Ziehen von Lottozahlen

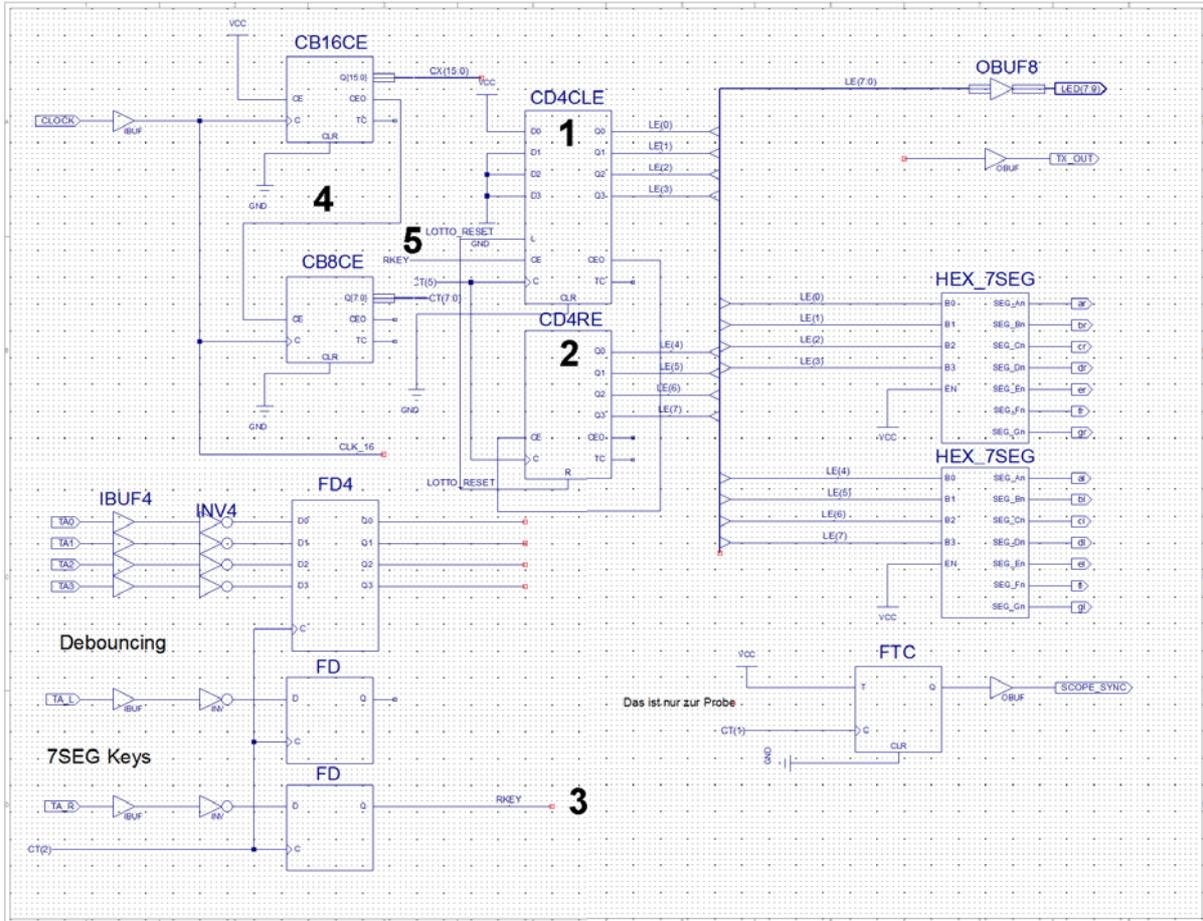
Auch diese Zähltaufgabe ist einfach. Es ist dezimal zyklisch von 1 bis 49 zu zählen, solange eine Taste niedergehalten wird. Wir bauen nur die Zähler auf Blatt 1 um. Die alten Zähler raus, die neuen rein. Den ersten müssen wir mit dem Wert Eins laden, den zweiten löschen. Hierzu brauchen wir einen Dezimalzähler mit synchronem Rücksetzen.

Wir müssen die Dezimalzahl 49 decodieren.  $4 = 100$ ,  $9 = 1001$  binär.

Dazu brauchen wir ein UND-Gatter mit folgender Beschaltung:

LE(6)	LE(5)	LE(4)	LE(3)	LE(2)	LE(1)	LE(0)
1	0	0	1	0	0	1

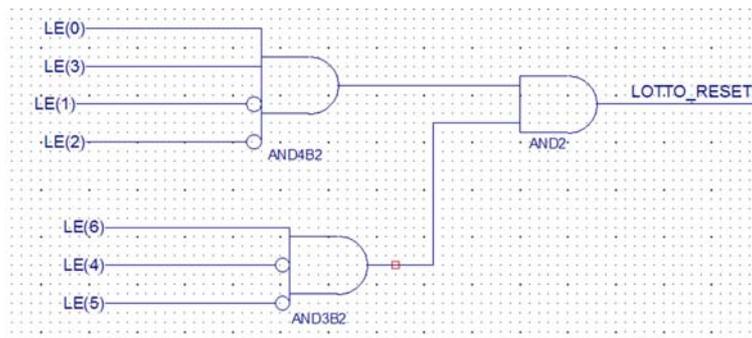
Die Schaltung bringen wir dort unter, wo Platz ist. Ggf. neues Blatt.



**Abb. 28** Ziehen von Lottozahlen. 1 - einen ladbaren Dezimalzähler einbauen. Die Dateneingänge mit dem Festwert Eins beschalten. 2 - einen synchron rücksetzbaren Dezimalzähler einbauen. 3 - das Tastensignal heißt jetzt RKEY. 4 - die Frequenzteiler entsprechen hier dem alten Stand (Teilung  $1:2^{24}$ ). Es kann aber auch der Umbau für die Stoppuhrausgabe (Teilung  $1 : 1,6$  Millionen) beibehalten werden. 5 - Zähleranschlüsse so beschalten, wie nachfolgend angegeben.

Beschaltung der Dezimalzähler:

1. Den Ladeeingang des ersten und den Rücksetzeingang des zweiten Zählers verbinden. Das Signal heißt LOTTO\_RESET.
2. Den Zählerlaubnis Eingang an das Tastensignal RKEY anschließen.
3. Die Takteingänge beider Zähler zunächst mit CT(5) verbinden, um den Zählablauf beobachten zu können. Dann auf CLK\_16 ändern.

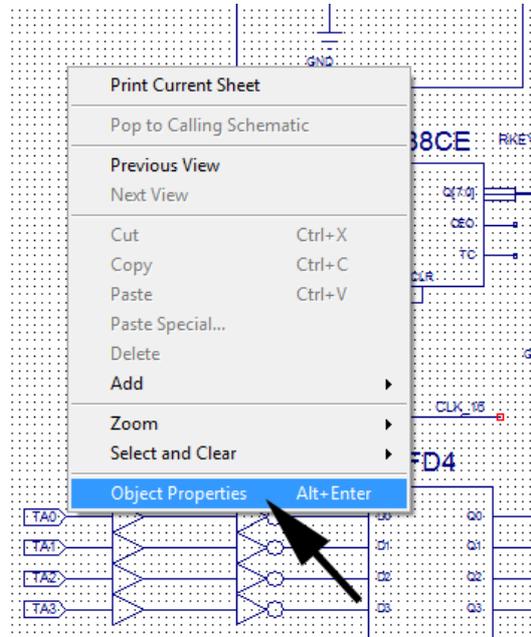


**Abb. 29** Der Lottozahlendecoder.

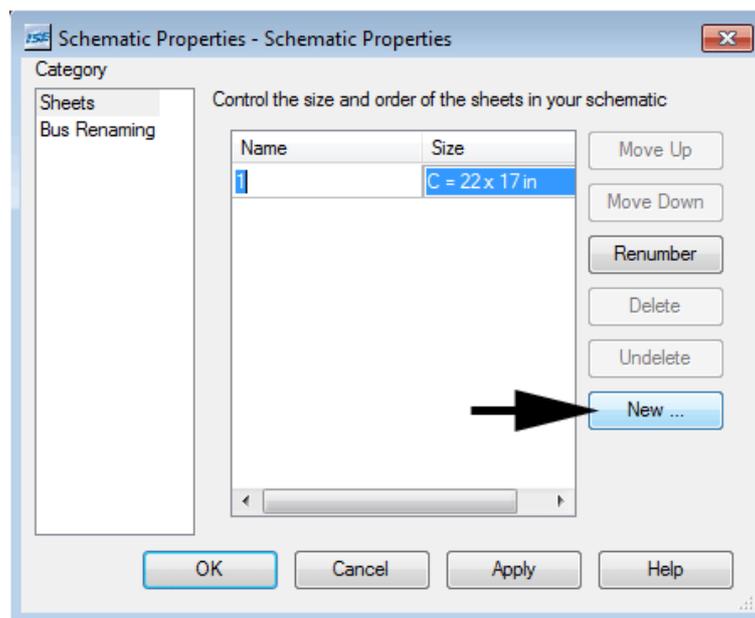
Erprobung: Wir haben den Takt der Zähler zunächst mit CT(5) verbunden. Wenn wir die Taste unter der rechten Siebensegmentanzeige niederhalten, muß der Apparat zyklisch von 1 bis 49 zählen, sofern wir die Taste lange genug gedrückt halten. Da wir nun wissen, daß der Apparat nicht falsch zählt oder schummelt, können wir den Takt auf 16 MHz ändern. Für die so gezogenen Lottozahlen können wir keine Gewähr übernehmen. Trotzdem viel Glück...

## Wie kommen wir zu weiteren Blättern im Schaltplan?

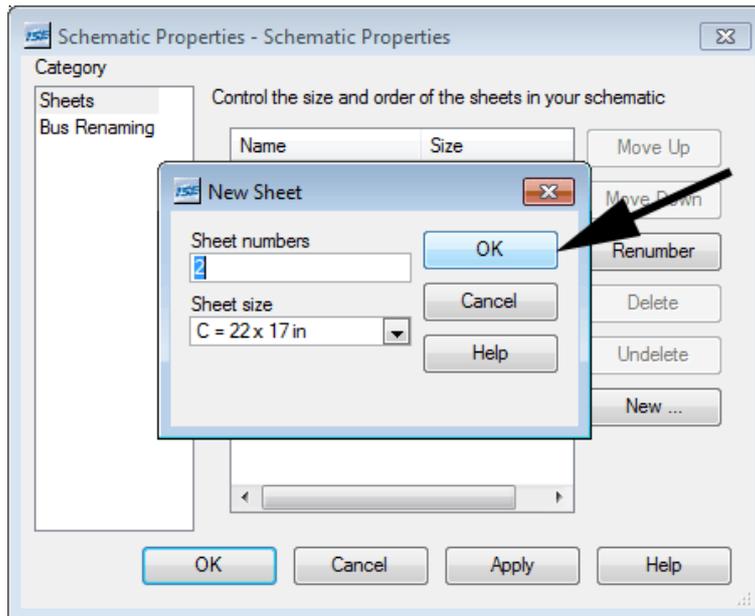
1. Mauscursor irgendwo auf die freie Zeichenfläche.
2. Rechte Maustaste. Dann *Object Properties*.
3. Dann *New* – *OK* – *OK*.



**Abb. 30** Anfordern eines neuen Blattes. Mauscursor auf die freie Zeichenfläche. Rechte Maustaste – “Object Properties”. 1 – 2 – 3: in dieser Reihenfolge klicken. Ggf. zuvor Papierformat auswählen (*Sheet Size*).

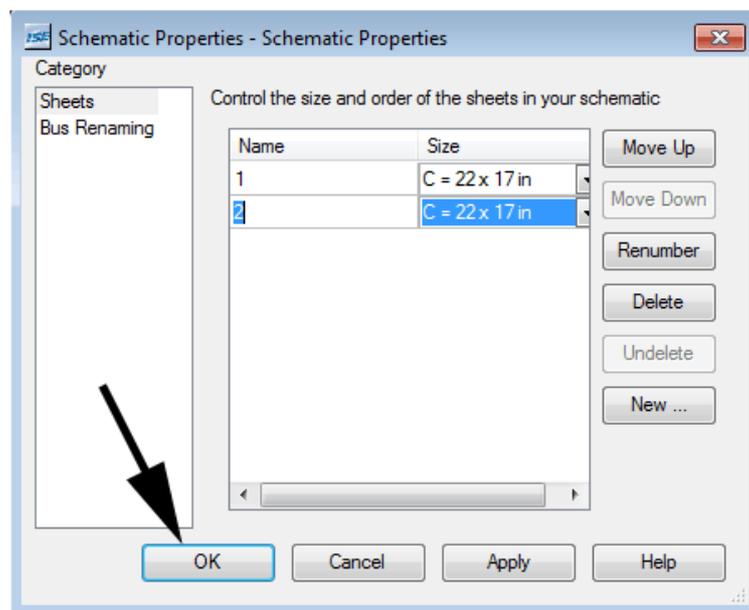


**Abb. 31** Wir brauchen eine neues Blatt. Also “New”...

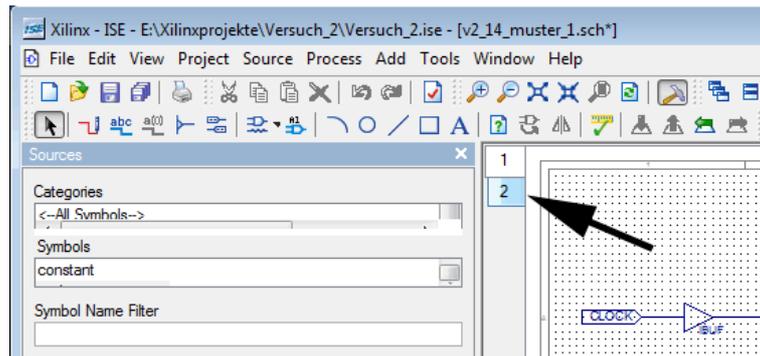


**Abb. 32** Bestätigen. Zuvor ggf. ein anderes Papierformat ("Sheet Size") wählen.

Praxistip: Wenn der Schaltplan gedruckt werden soll (Projekt, Abschlußarbeit usw.), auf Format A4 gehen und die Schaltung auf entsprechend viele Blätter aufteilen. Macht zwar etwas Arbeit, sieht aber VIEL besser aus...



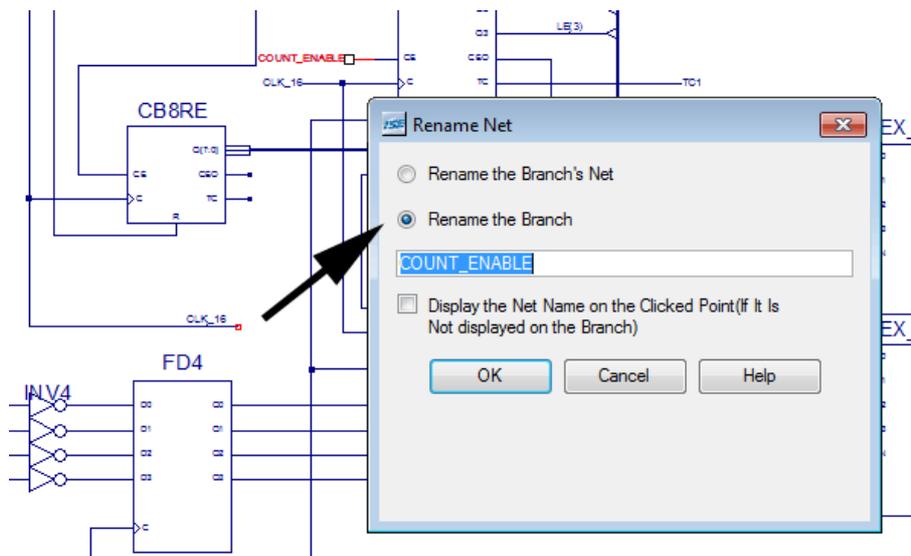
**Abb. 33** Das neue Blatt wurde angelegt. OK.



**Abb. 34** Das neue Blatt ist im Schaltplan verfügbar und kann dort angeklickt werden.

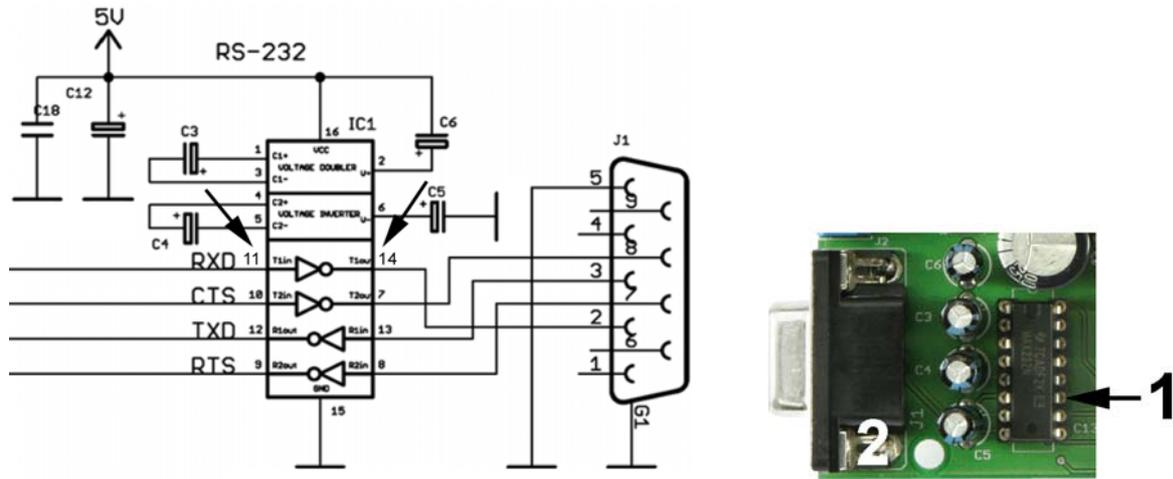
## Signale an Eingängen umbenennen

An einen einzelnen Eingang soll ein anderes Signal angeschlossen werden als bisher angeschlossen war. Alles andere soll so aber bleiben wie es ist. Leitung mit rechter Maustaste anklicken – *Rename Selected Net*. Jetzt unbedingt *Rename the Branch* auswählen – sonst wird das bisher angeschlossene Signal im gesamten Schaltplan umbenannt.



**Abb. 35** Aufpassen. Wenn nur der eine Eingang umbenannt werden soll, unbedingt diese Variante auswählen – sonst geht's schief...

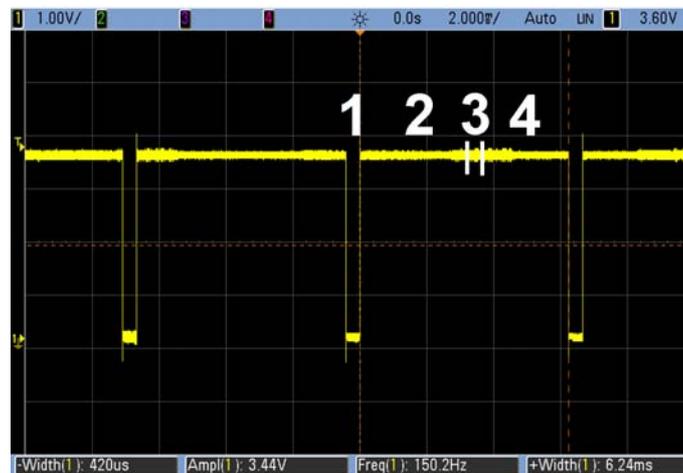
### Signalverlaufsbeispiele der seriellen Schnittstelle



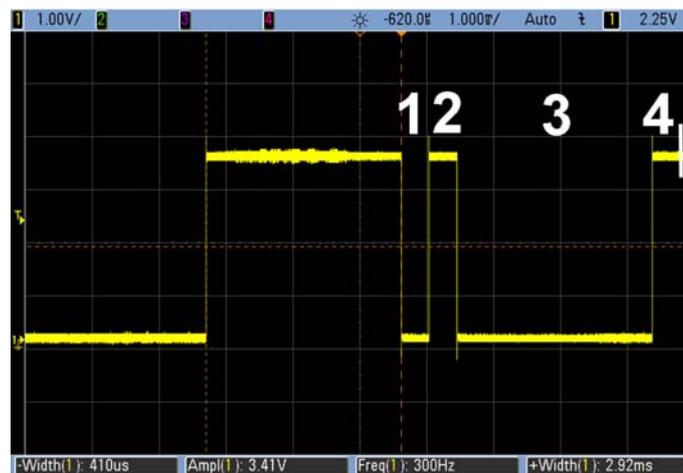
**Abb. 36** Hier messen. An Pin 11 des Schnittstellentreibers kommt das Signal TX\_OUT an. 1 - Tastkopfspitze; 2 - Masseklemme.



**Abb. 37** Zeichen 00H. 1 - Stopbit; 2 - Pause; 3 - Startbit; 4 - acht Datenbits (Nullen).



**Abb. 38** Zeichen FFH. 1 - Startbit; 2 - acht Datenbits (Einsen); 3 - Stopbit; 4 - Pause.

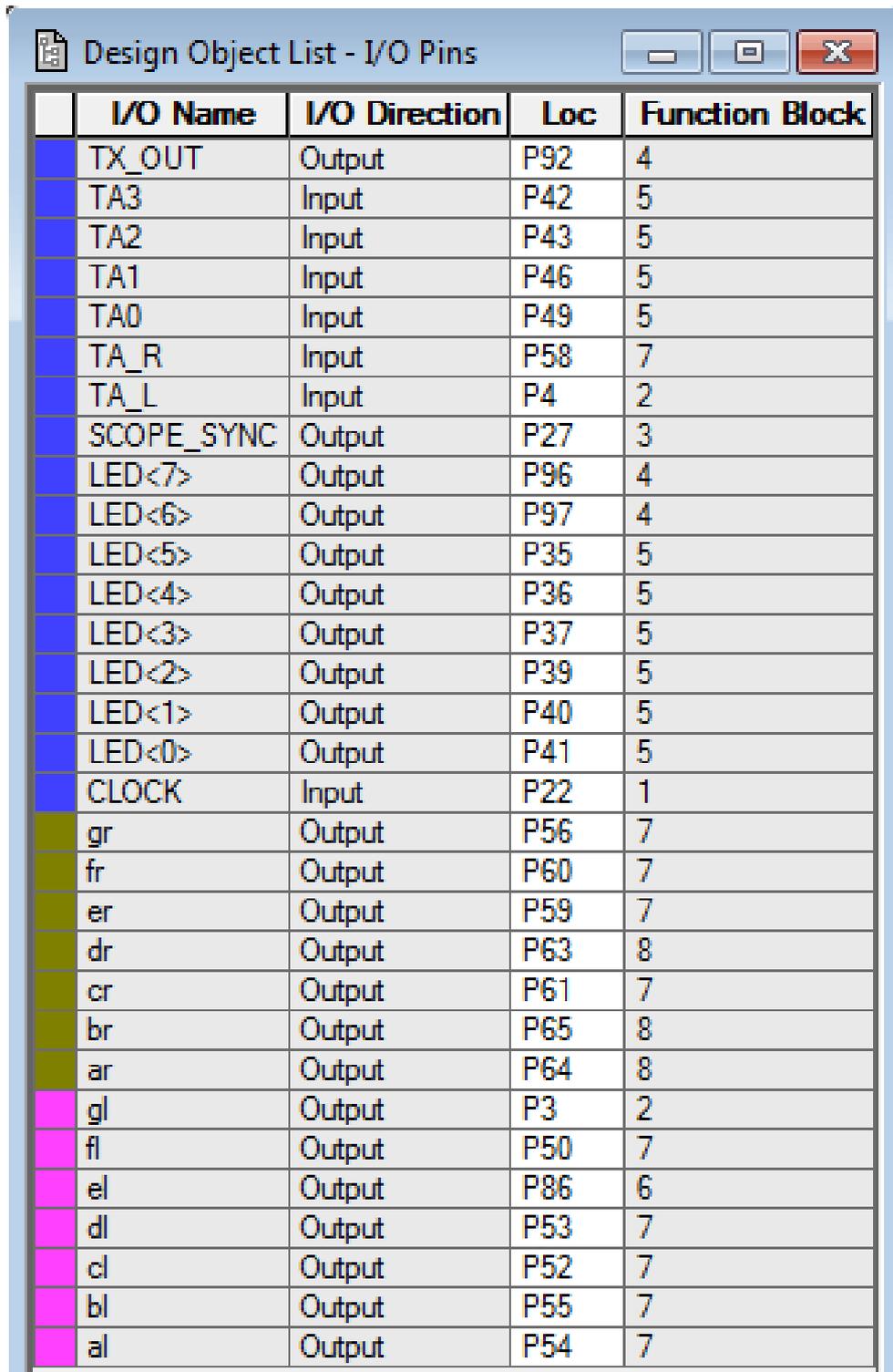


**Abb. 39** Zeichen 01H = 0000 0001. Übertragungsreihenfolge 1000 0000. 1 - Startbit; 2 - Bit 0 = 1; 3 - die weiteren sieben Datenbits (Nullen); 4 - Stopbit.



**Abb. 40** Zeichen 31H = 0011 0001. Übertragungsreihenfolge 1000 1100.

## Die Signalliste der Constraintsdatei



	I/O Name	I/O Direction	Loc	Function Block
	TX_OUT	Output	P92	4
	TA3	Input	P42	5
	TA2	Input	P43	5
	TA1	Input	P46	5
	TA0	Input	P49	5
	TA_R	Input	P58	7
	TA_L	Input	P4	2
	SCOPE_SYNC	Output	P27	3
	LED<7>	Output	P96	4
	LED<6>	Output	P97	4
	LED<5>	Output	P35	5
	LED<4>	Output	P36	5
	LED<3>	Output	P37	5
	LED<2>	Output	P39	5
	LED<1>	Output	P40	5
	LED<0>	Output	P41	5
	CLOCK	Input	P22	1
	gr	Output	P56	7
	fr	Output	P60	7
	er	Output	P59	7
	dr	Output	P63	8
	cr	Output	P61	7
	br	Output	P65	8
	ar	Output	P64	8
	gl	Output	P3	2
	fl	Output	P50	7
	el	Output	P86	6
	dl	Output	P53	7
	cl	Output	P52	7
	bl	Output	P55	7
	al	Output	P54	7