

Praktikum Digitaltechnik SS 2012

Versuch 3

Stand: 9. 7. 12

Versuchsplattform:

Lehrgerät 10a mit CPLD Evaluation Board (Pollin) und zwei Siebensegmentanzeigen 10a. Eingebaute Ein- und Ausgabemittel: vier Taster, acht LEDs, serielle Schnittstelle. Entwicklungssoftware: Xilinx ISE 9.x. Erprobung des Schnittstellentesters: am PC mittels Kommunikationsprogramm (Hyperterminal).

Aufgaben:

- Den Ausgangszustand herstellen.
- Schnittstellentester mit Dauerbetrieb.
- Schnittstellentester mit Einzelauslösung.
- Schnittstellentester mit Dauer- und Einzelbetrieb.
- Ziehen von Lottozahlen.
- Ringzähler.
- Johnsonzähler.
- Rückgekoppeltes Schieberegister.

– 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).

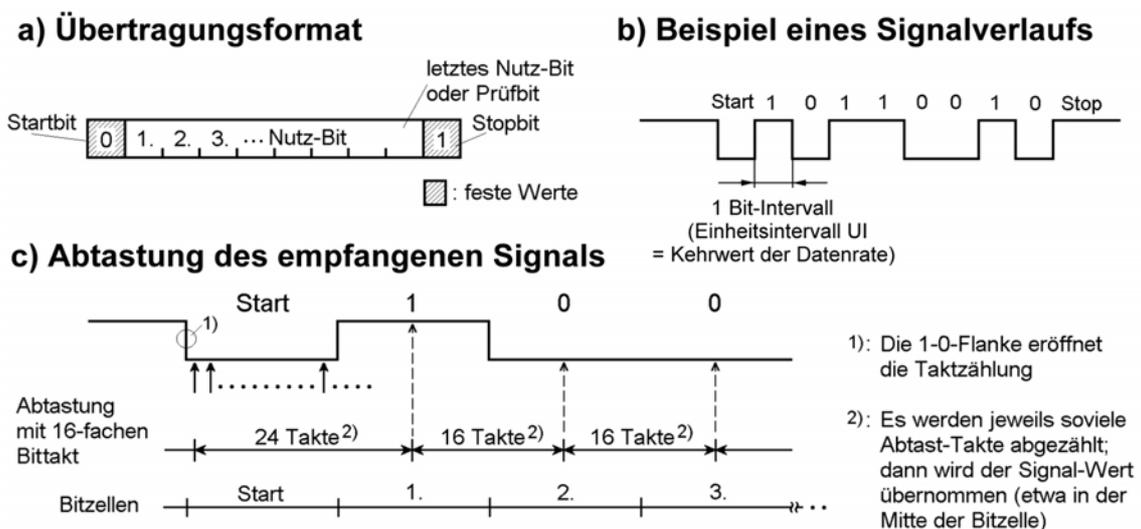


Abb. 1 Signalübertragung nach dem Start-Stop-Verfahren (1). Grundlagen.

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 1½ Stopbits ("1½ Bits" bedeutet, daß der High-Pegel wenigstens 1½ Bitzellen lang anliegen muß).

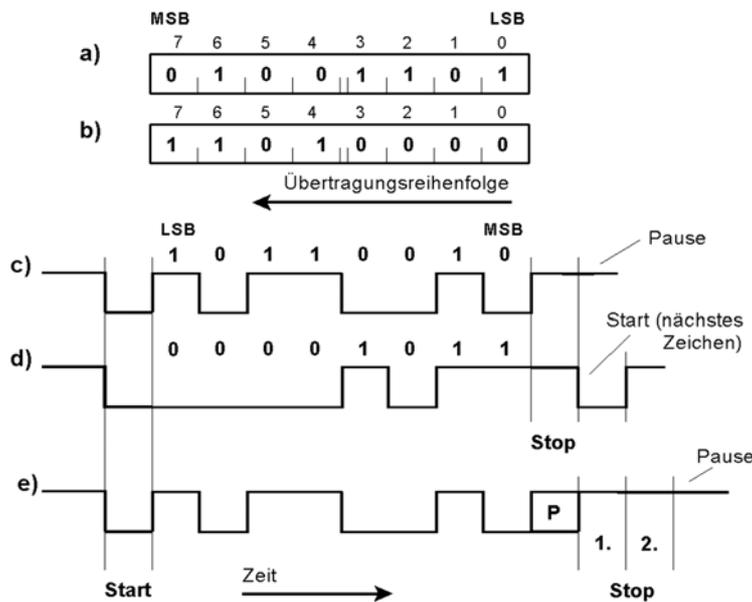


Abb. 2 Signalübertragung nach dem Start-Stop-Verfahren (2). Übertragungsbeispiele.

- 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 Grundsaltung wurde bereits im 2. 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. Xilinx-Umgebung aufrufen und ein neues Projekt anlegen, aber ohne eigene Quellen.
2. Die Quellen importieren.
3. Projekt durchlaufen lassen.
4. IMPACT aufrufen und Schaltkreis programmieren.
5. Funktionserprobung.

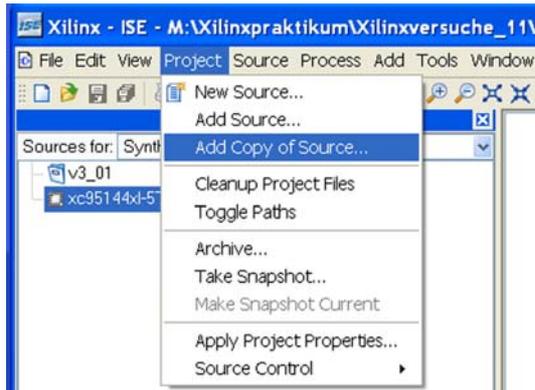


Abb. 3 So werden die Quellen importiert. Wir brauchen den Schaltplan, den Siebensegmentdecoder und die Constraintsdatei. Sie ist schon vorgefertigt und muß während des Versuchs nicht mehr abgeändert werden.

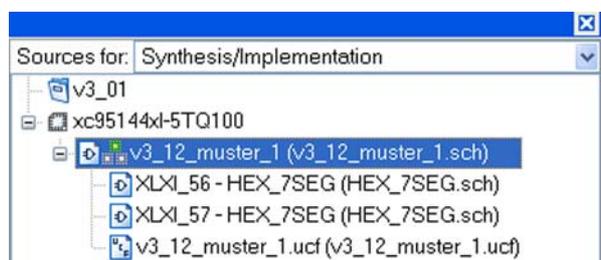


Abb. 4 Unser Projekt muß aus diesen Quellen bestehen.

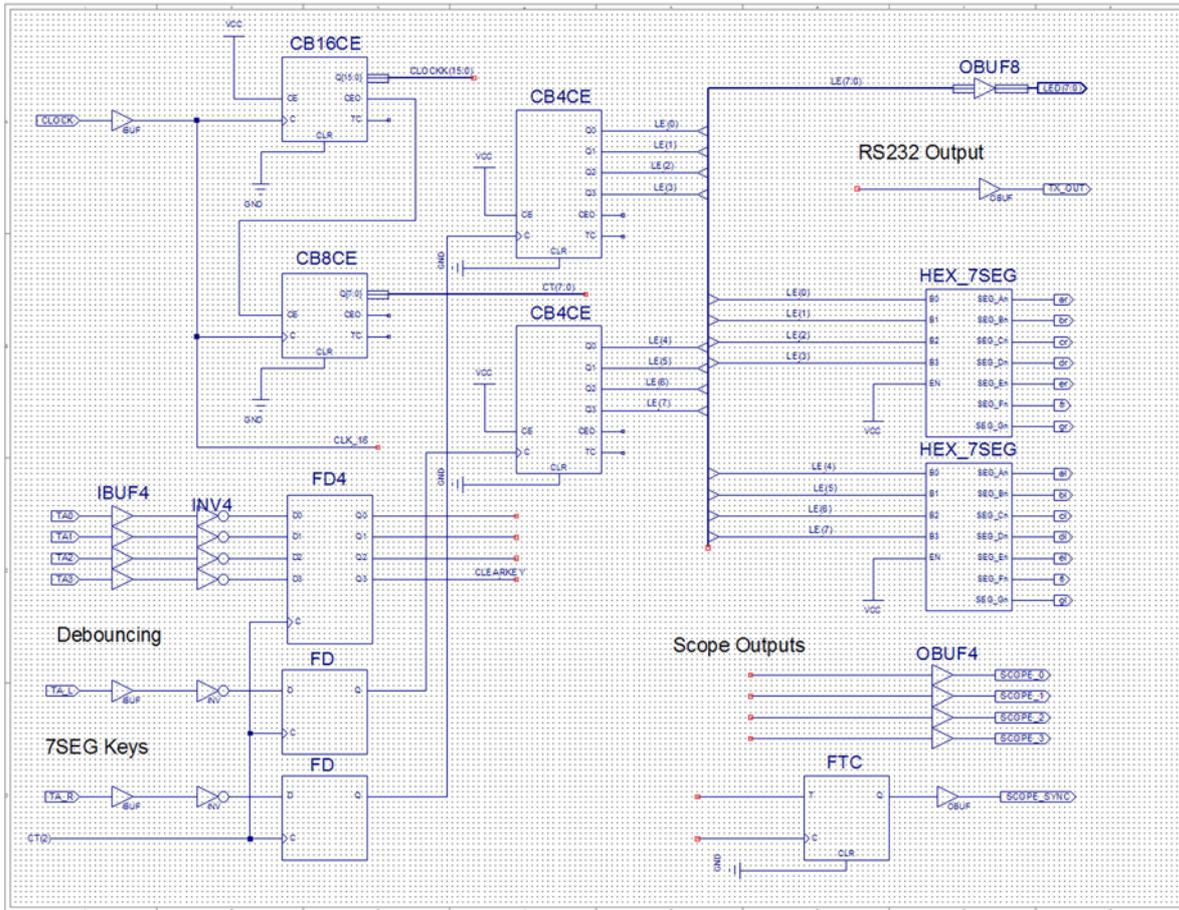


Abb. 5 Der ursprüngliche Schaltplan. Beim Betätigen der Tasten unter den Siebensegmentanzeigen muß die jeweils zugehörige Anzeige zyklisch vorwärts zählen.

Aufgabe 2: Schnittstellentester mit Dauerbetrieb

Die Aufgabe des Schnittstellentesters besteht darin, ein einziges Prüfzeichen zu senden. Der Zeichencode wird hexadezimal eingestellt (siehe Codetabelle). 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. Wir haben aber keine 16 Kippschalter. Deshalb machen wir nur die acht Bitpositionen des eigentlichen Prüfzeichens veränderlich und beschalten diese mit zwei Einstellzählern zu 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 S. 19).

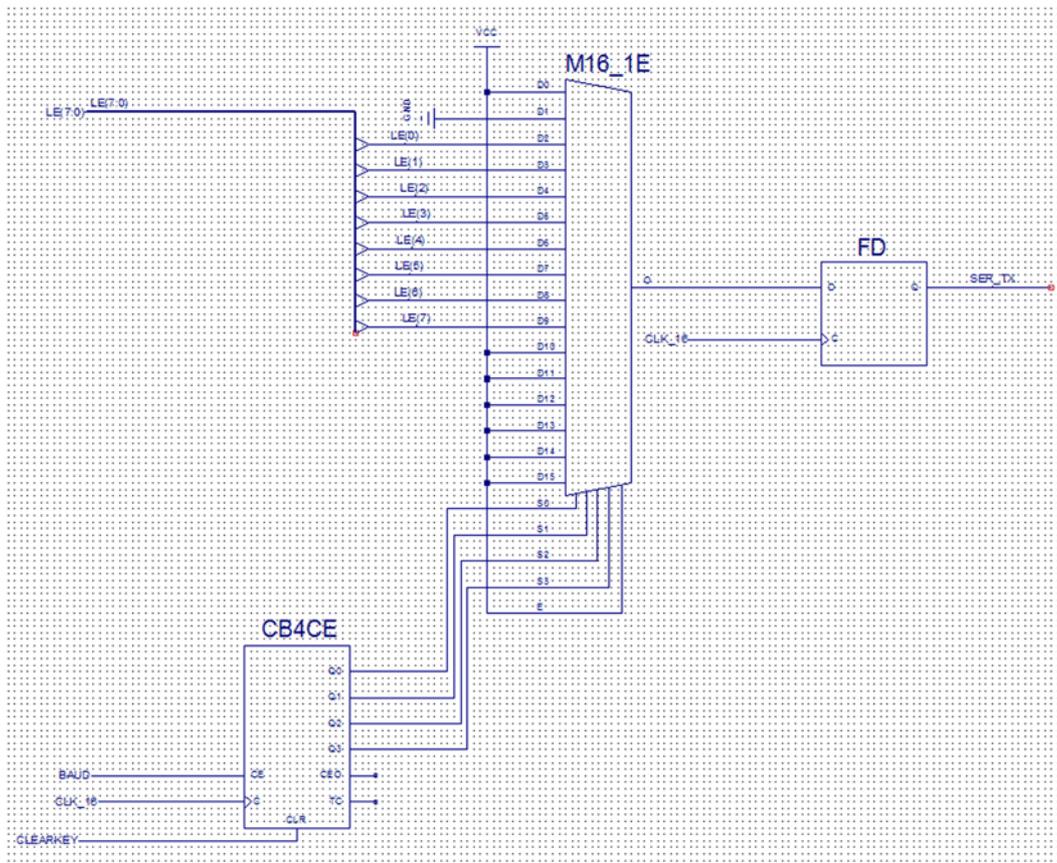


Abb. 6 Eine der Lösungen, um ein beliebiges serielles Bitmuster herzutsallen. Der Eingang des Output-Puffers TX_OUT (Blatt 1) muß als SER_TX benannt werden.

Wir müssen mit 2400 Bits/s ausgeben. $16\ 000\ 000 : 2400 = 6666,66\dots$

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 Beschtung:

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 den Eingang des OUBFs SCOPE_0 (Blatt 1) als SER_TX benennen. Zum Anschließen des Tastkopfs s. Seite 12. Signalverlaufsbeispiele auf Seite 20f.

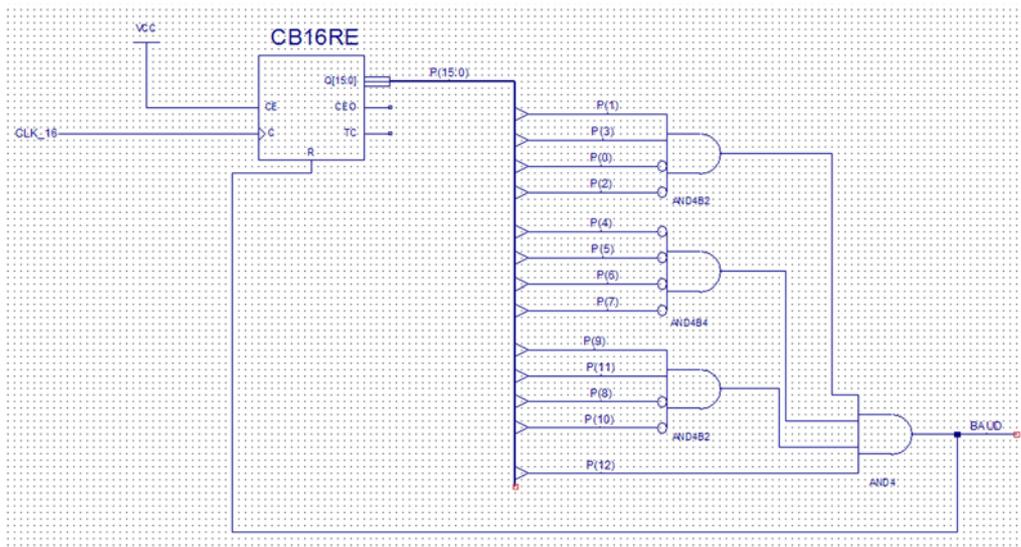


Abb. 7 Das Signal BAUD wird rund 2400mal je Sekunde einen Taktzyklus lang aktiv.

Erprobung:

1. Serielle Schnittstelle anschließen (Kabel mit 9poligem D-Sub-Stecker vom PC).
2. Hyperterminal aufrufen (Programme – Zubehör – Kommunikation).
3. Neue Verbindung einstellen. 2400 Baud, 8 Datenbits, kein Paritätsbit, 1 Stopbit, kein Handshaking.
4. Am Lehrgerät 10a einen passenden ASCII-Code einstellen, beispielsweise 41H = "A".

Die Bildschirmfläche des Hyperterminal muß mit dem gewählten Zeichen vollgeschrieben werden.

Es kann sein, daß es nicht richtig klappt. Ggf. die Taste 3 (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.

	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. 8 Der ASCII-Code (Codetabelle). Beispiel: Prüfzeichen 1 = 31H, Prüfzeichen A = 41H.

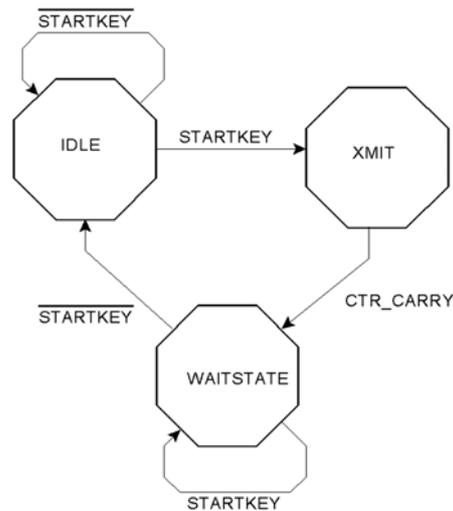


Abb. 11 Das Zustandsdiagramm.

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, wenn 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).

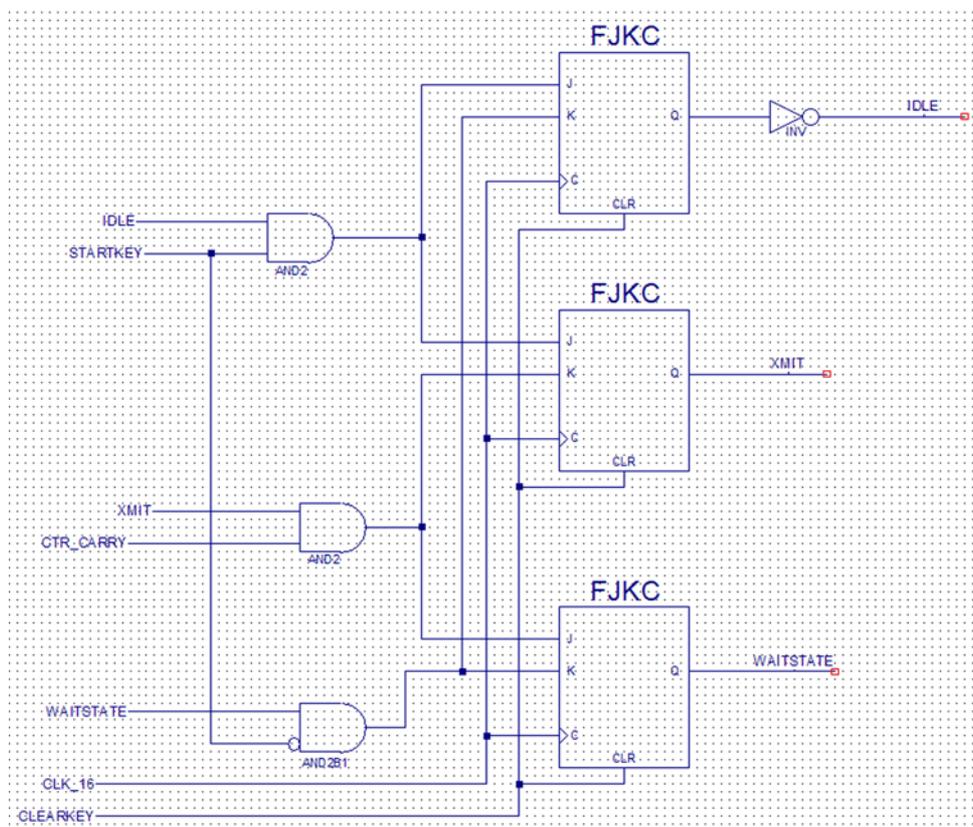


Abb. 12 Der Zustandsautomat. Der Sendezustand XMIT wird erst dann, verlassen, nachdem der Abfragezähler durchgelaufen ist (CTR_CARRY).

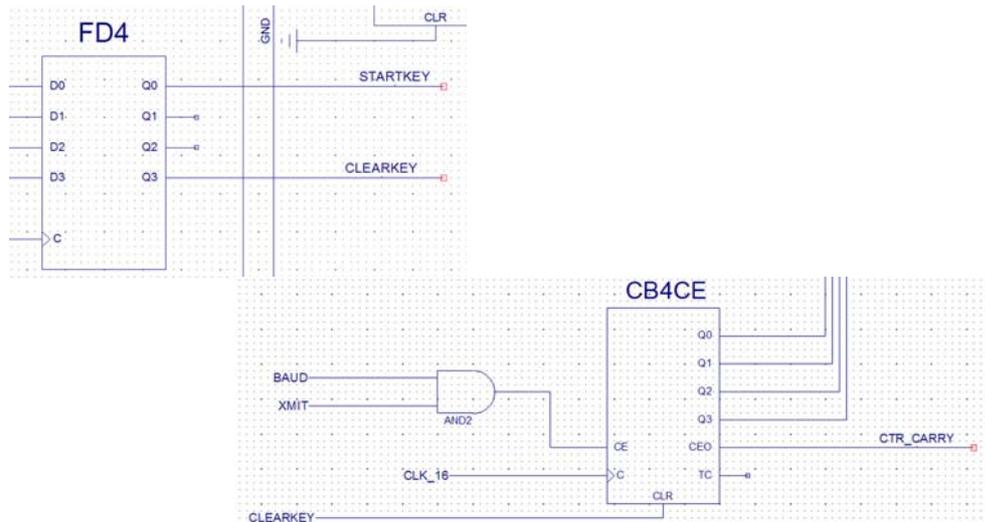


Abb. 13 Weitere Änderungen.

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

Aufgabe 4: Schnittstellentester mit Dauer- und Einzelbetrieb

Jetzt sollen noch zwei Tasten hinzukommen. 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.

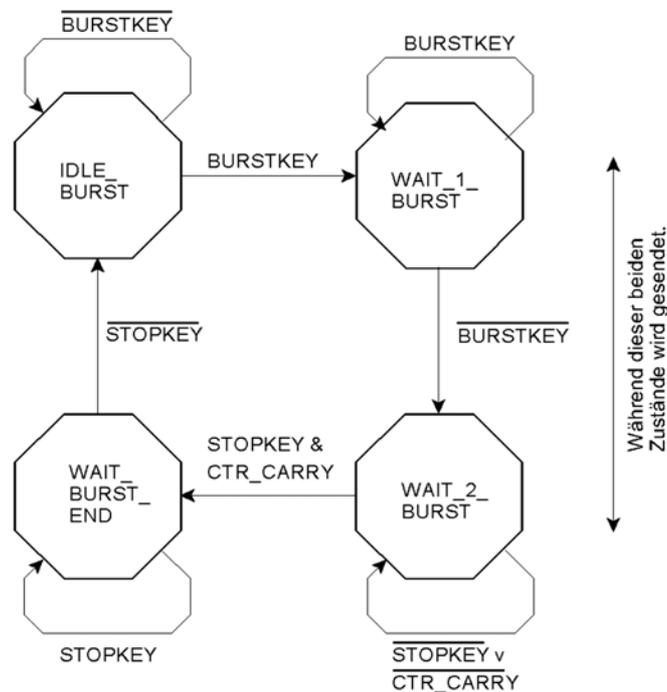


Abb. 14 Der neue Zustandsautomat. Es sind zwei Tasten. Das Drücken der ersten (BURSTKEY) löst das Senden aus. Nun muß darauf gewartet werden, bis sie wieder losgelassen wird. Dann wird aber immer noch gesendet, und zwar solange, bis die zweite Taste (STOPKEY) betätigt wird und der Abfragezähler durchgelaufen ist. Jetzt ist noch zu warten, bis auch diese Taste losgelassen wird.

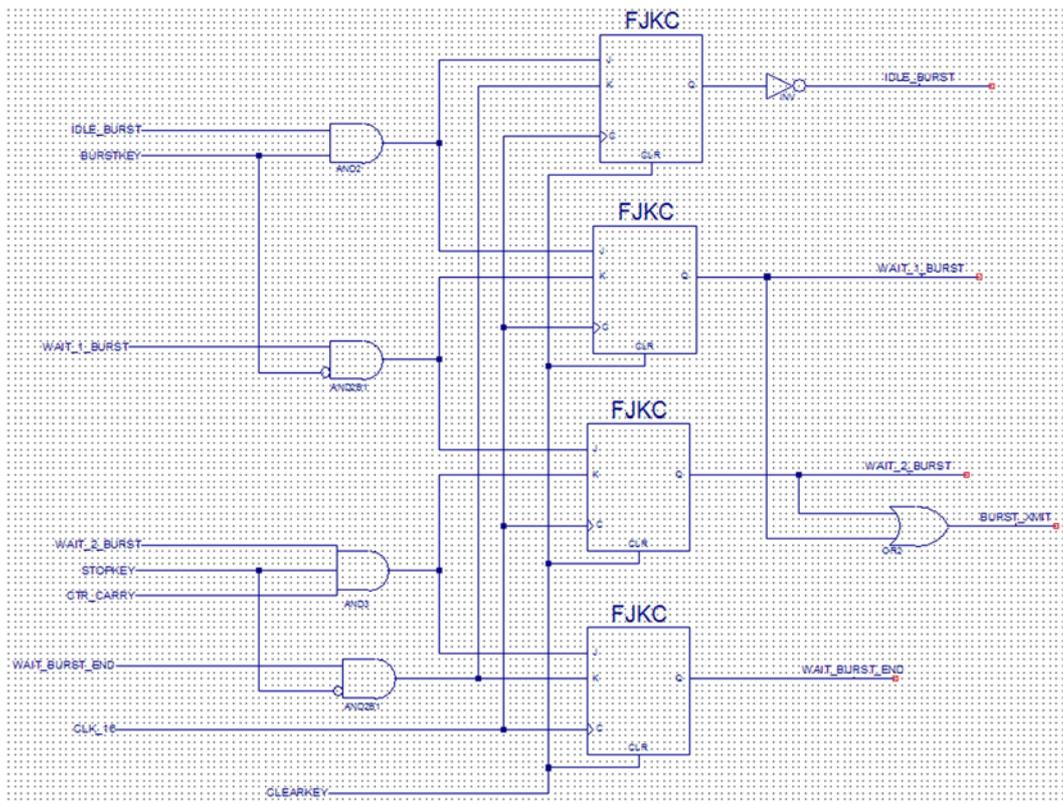


Abb. 15 Der zweite Zustandsautomat.

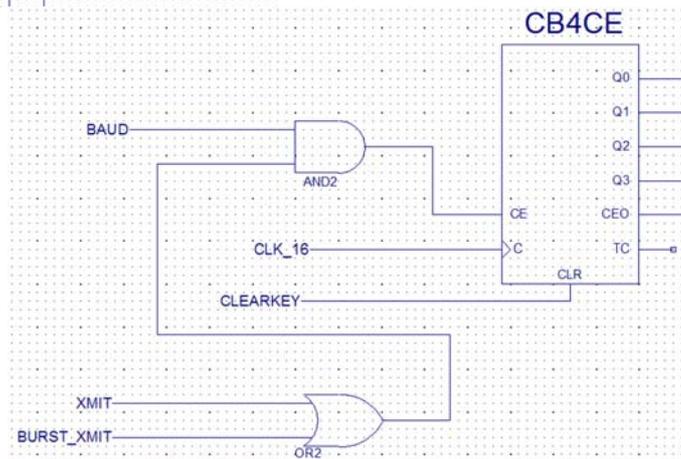
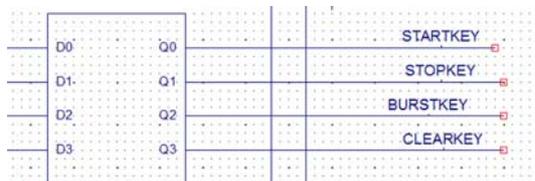


Abb. 16 Änderungen, Änderungen...

Aufgabe 5: Ziehen von Lottozahlen

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

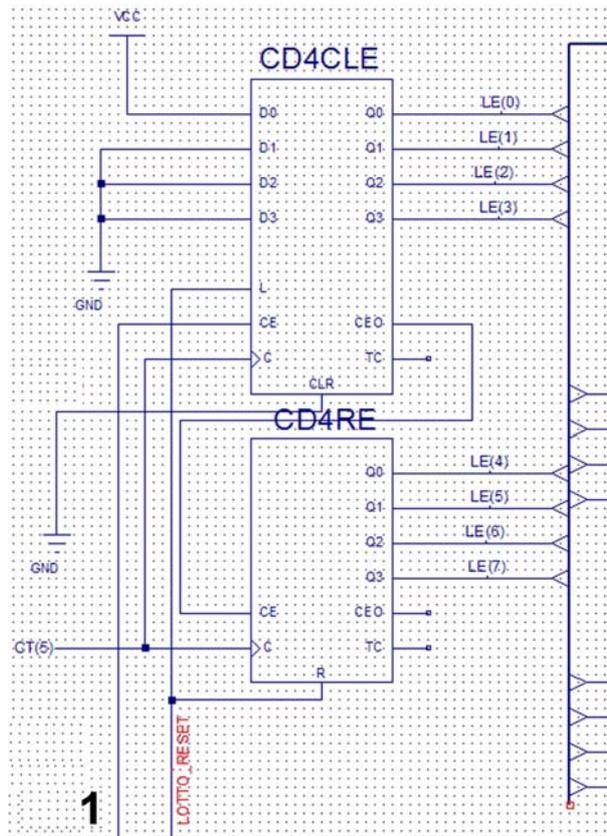


Abb. 17 Um fertig zu werden, wird einfach und schmucklos geändert. 1 kommt von der Taste unter der rechten Siebensegmentanzeige.

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

Wir brauchen 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.

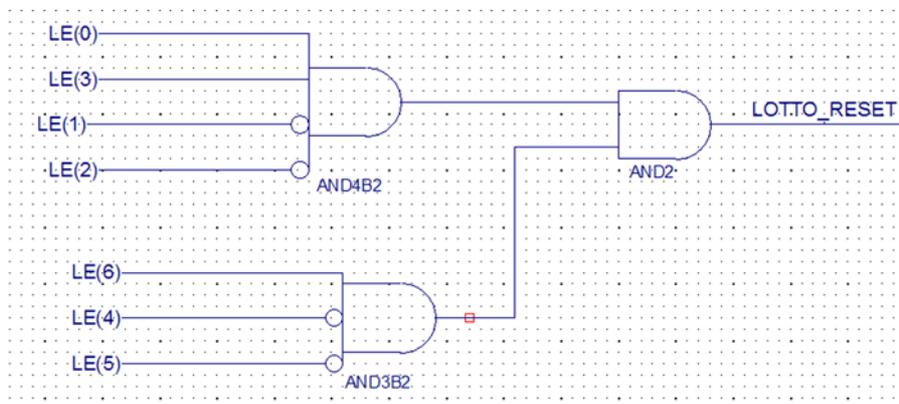


Abb. 18 Der Lottozahlendecoder.

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...

Aufgabe 6: Ringzähler

Wir brauchen wieder ein neues Blatt, um den Schaltplan zu zeichnen. Den OBUF8 der LEDs löschen und den zugehörigen Marker auf das neue Blatt schaffen. Dort einen neuen OBUF8 andocken. Jetzt stehen die LEDs zur Anzeige zur Verfügung. Der Ringzähler hat vier Flipflops. Beim ersten wenden wir den uns schon bekannten Trick an, um nach dem Programmieren oder Einschalten automatisch den Anfangswert 1 zu haben.

Erprobung:

1. Mit Takt von Hand (Taste 0) und Beobachten der LEDs.
2. Mit Oszilloskop. Takt auf CLOKK(4) ändern (1 MHz). Die Scope Outputs auf Blatt 1 passend verbinden. Oszilloskop über Tastköpfe anschließen. Masseklemme ggf. an Befestigung des D-Sub-Steckverbinders der seriellen Schnittstelle. Kanal 1 an SCOPE_SYNC. Über Kanal 2 die einzelnen Signale nacheinander anzeigen.

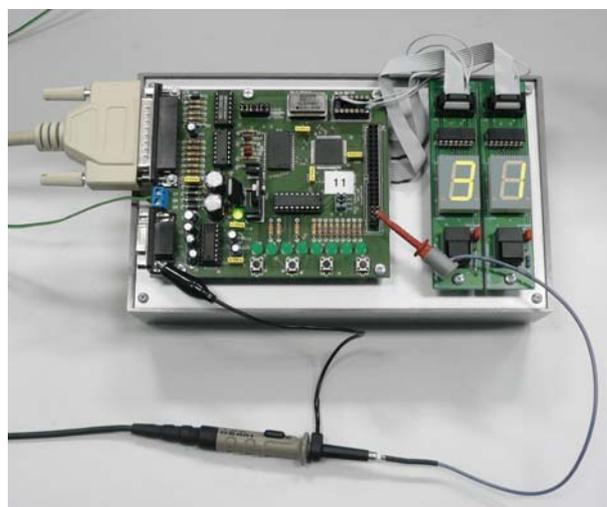
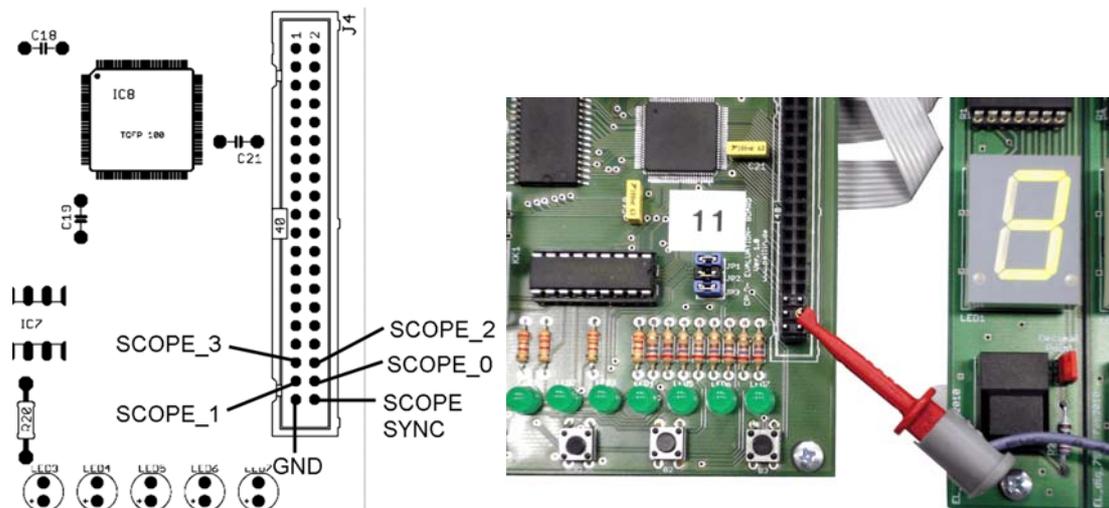


Abb. 19 Die Oszilloskopanschlüsse.

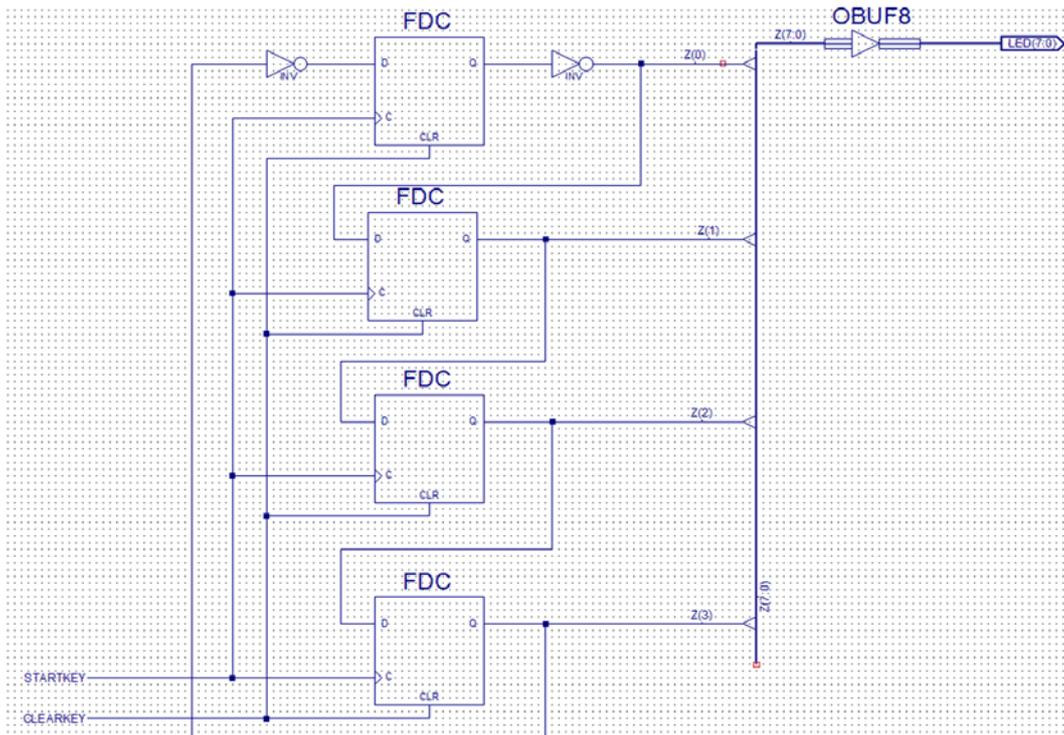


Abb. 20 Der Ringzähler. Wenn es mit der LED-Anzeige funktioniert, STARTKEY durch CLOK(4) ersetzen.

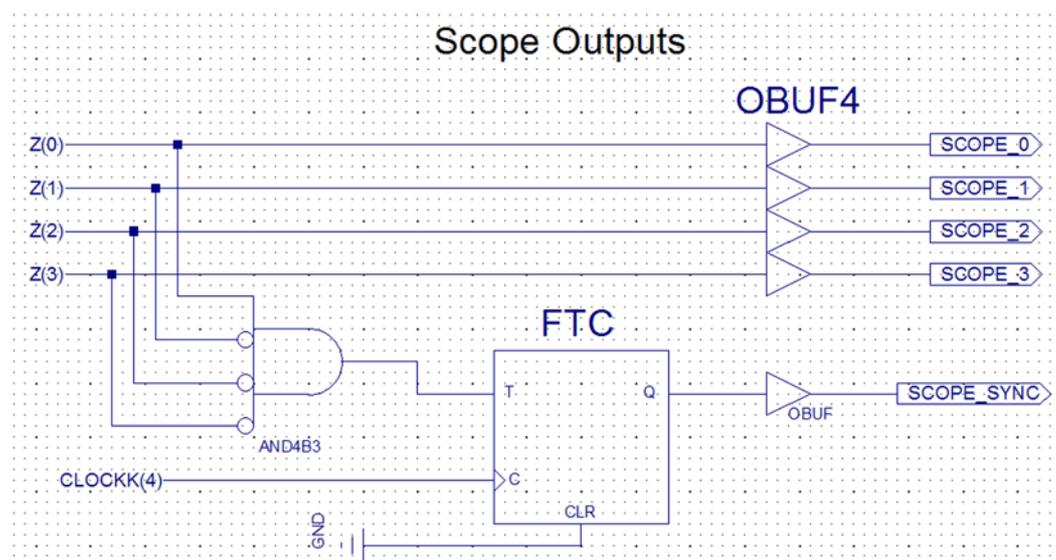


Abb. 21 Die Oszilloskopanschlüsse. Das UND-Gatter ANDB3 decodiert die Anfangsstellung 0001B..

Aufgabe 7: Johnsonzähler

Die Zählerschaltung wird einfach und schmucklos abgeändert. Für die Synchronisation brauchen wir aber ein AND4B4 (alle vier Eingänge negiert, um den Anfangswert 0000B zu erkennen). Erprobung wie Ringzähler.

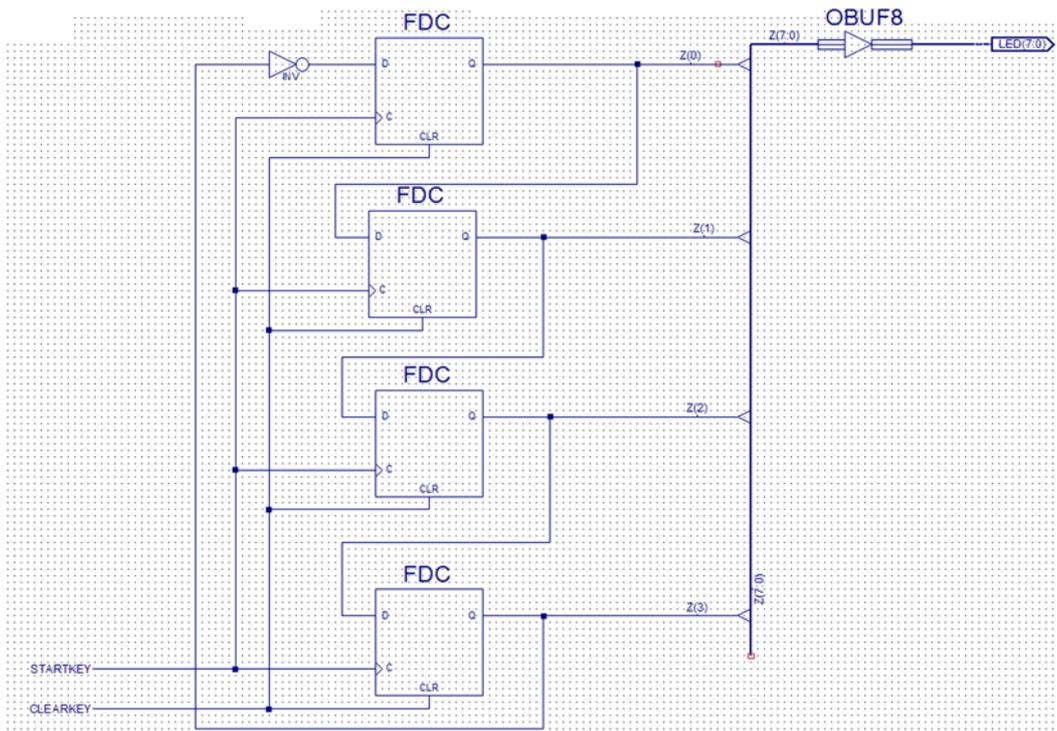


Abb. 22 Der Johnsonzähler. Wenn es mit der LED-Anzeige funktioniert, STARTKEY durch CLOKK(4) ersetzen.

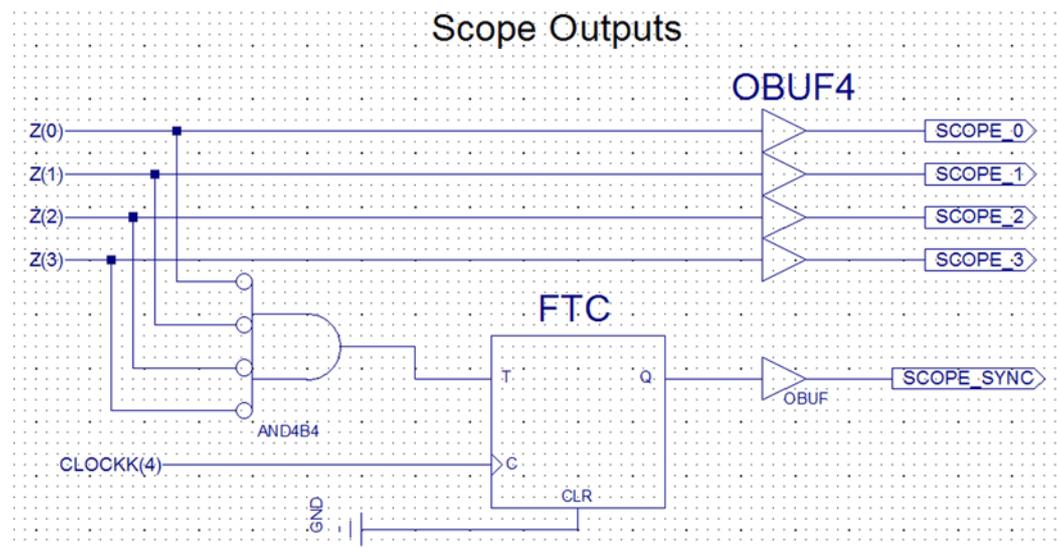


Abb. 23 Nur eine Kleinigkeit ändern...

Aufgabe 8: Rückgekoppeltes Schieberegister

Eine nochmalige kleine Änderung mit großer Wirkung. Siehe auch die Anleitung zu Versuch 1. Es genügt, eines der Ausgangssignale zu beobachten, da alle gleich aussehen (nur Verzögerung). Das UND-Gatter AND4B3 muß wieder rein (s. Ringzähler).

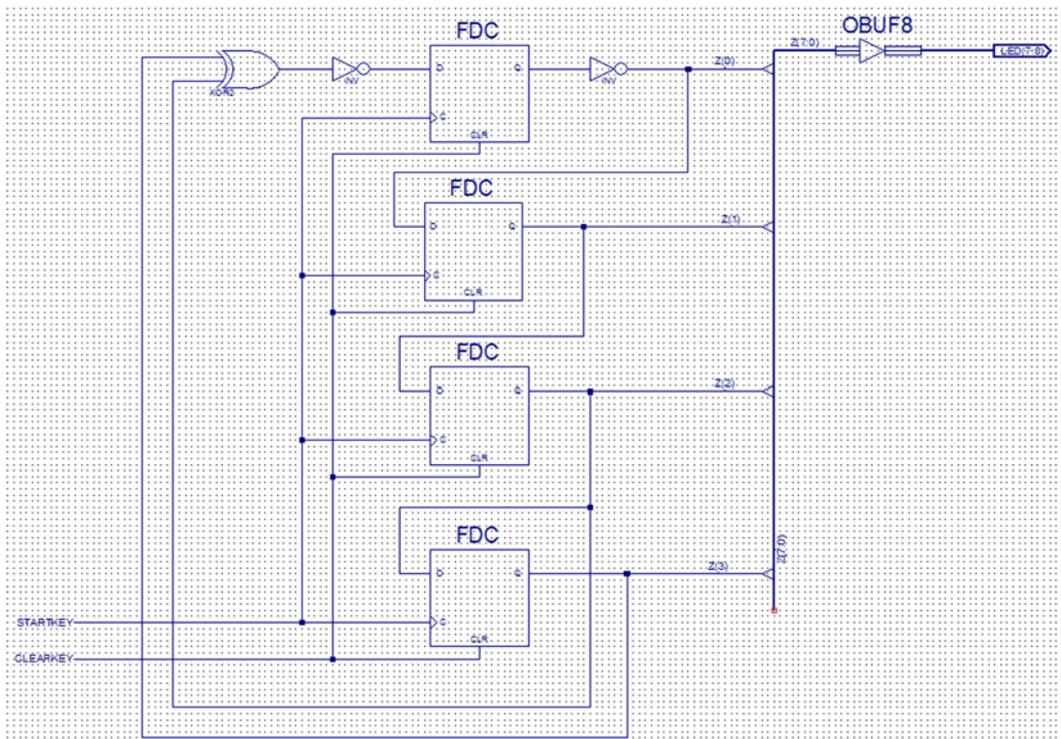


Abb. 24 Das rückgekoppelte Schieberegister.

Zum Schluß wird ein längeres Schieberegister gebaut. Es besteht aus 2 Schieberegistern zu je 4 Bits. Wir brauchen einen Anfangswert, gleichgültig welchen, nur nicht Null. Er soll nicht viel Arbeit machen. Deshalb wird das erste Schieberegister mit Einsen geladen, das zweite gelöscht. Dieses Bitmuster (0FH) wird mit einer UND-Verknüpfung decodiert. Deren Ausgang wird genau einmal in jedem Umlauf (Periode, Zyklus) aktiv. Die Anzahl der Takte in jedem Umlauf (Zykluslänge) wird gezählt und headezimal angezeigt.

Erprobung:

1. Prüfung mit einem langsamen Takt anhand der LEDs auf der Pollin-Platine.
2. Takt des Schieberegisters und des Zählers auf CLOCKK(4) ändern. Beobachten der Zykluslängenzählung und des erzeugten seriellen Bitmusters (Oszilloskop).
3. Die Rückführung ändern. Hierzu das XOR-Gatter mit Z(6) und Z(7) beschalten. Die freien Eingänge an GND. Alternativ dazu die Luxusvariante gemäß Abbildung 28 bauen und Taste 0 (STARTKEY) niederhalten. Welche Zykluslänge ergibt sich jetzt?

Das Schieberegister läuft nach dem Programmieren oder Einschalten nicht mehr automatisch an. Korrekte Funktion erst nach Betätigen der Taste 3 (CLEARKEY). Solange die betätigt wird, müssen Schieberegister und Zykluslängenzähler auf ihren Anfangswerten (0FH, 00H) stehenbleiben. Signalverlaufsbeispiele s. Seite 21.

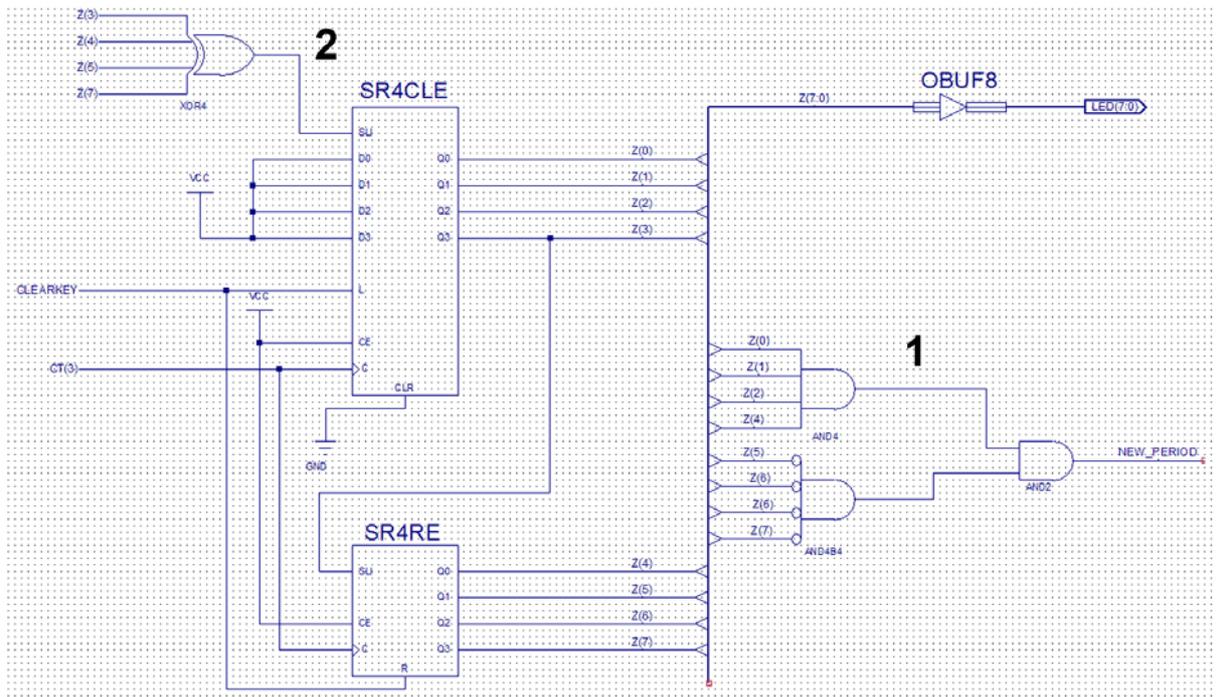


Abb. 25 Das längere rückgekoppelte Schieberegister. Anfangswert 0FH. 1 - Decodieren des Anfangswertes; 2 - Rückkopplung. Hiermit muß die Zykluslänge = 255 (FFH) sein (maximale Zykluslänge).

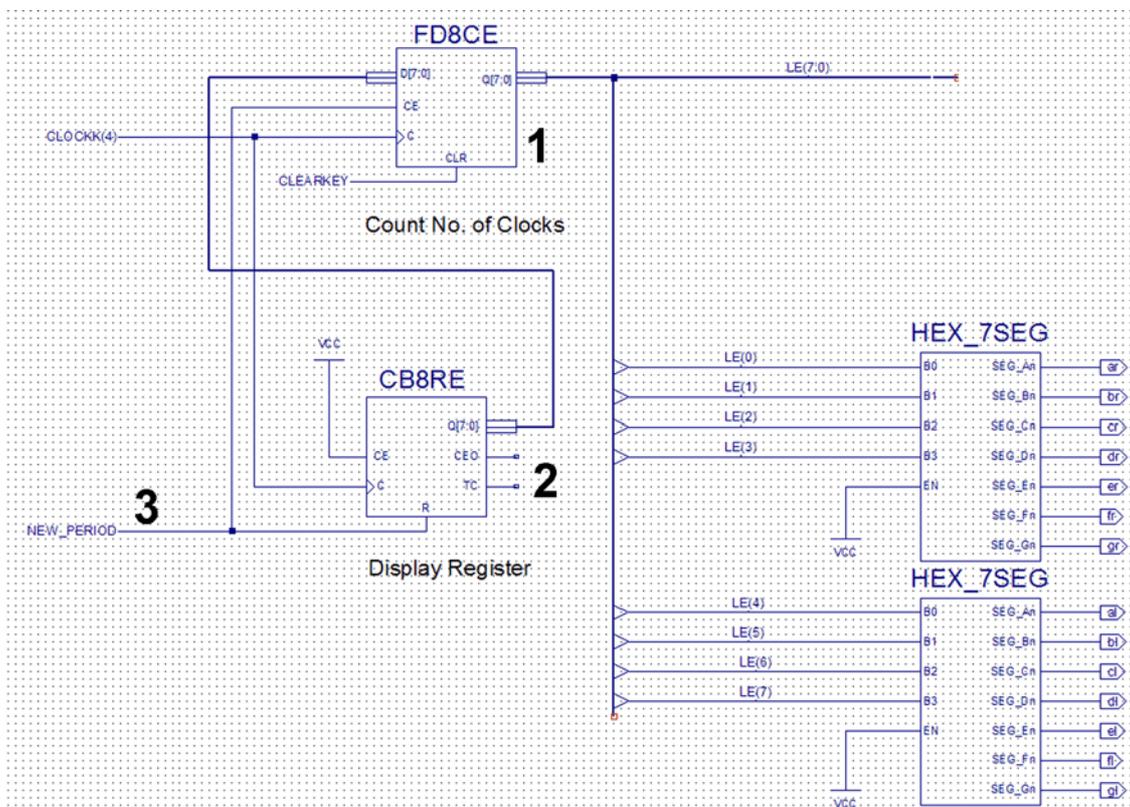


Abb. 26 Zählen der Zykluslänge. 1 - 8-Bit-Binärzähler; 2 - Anzeigeregister. Der Zähler 1 zählt die Taktimpulse. Zu Beginn eines neuen Umlaufs wird er gelöscht (3). Zuvor aber wird der aktuelle Inhalt in das Anzeigeregister 2 übernommen. Angezeigter Wert = Zykluslänge - 1 (in Hex). In den Erprobungsbeispielen ergeben sich die Anzeigen FEH und 3EH.

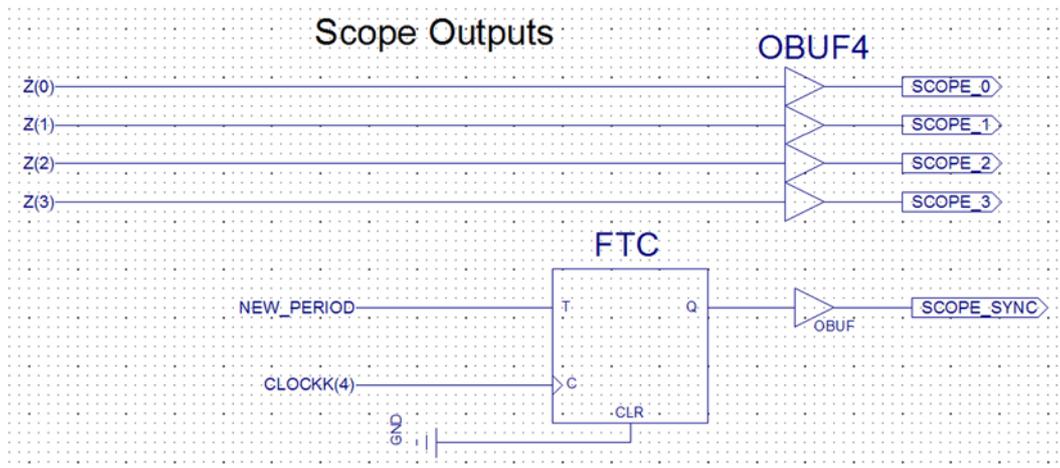


Abb. 27 Das Anfangswertsignal wird zum Triggern des Oszilloskops genutzt.

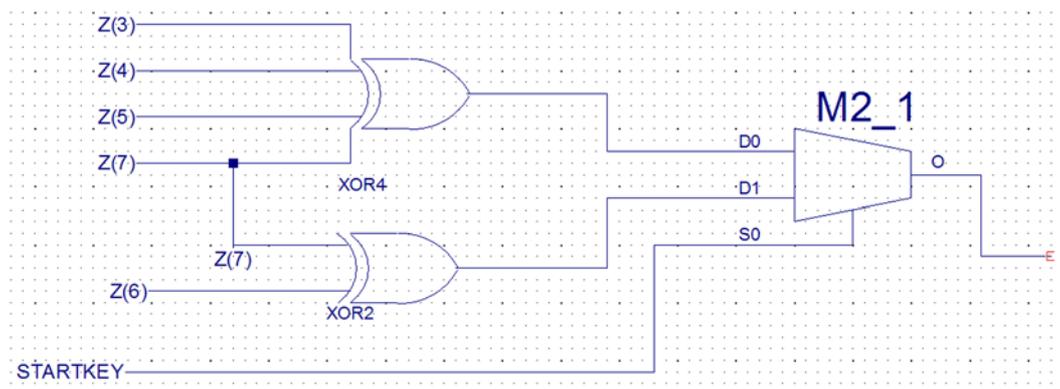


Abb. 28 Eine Luxusvariante. Umschaltung der Zykluslänge mittels Taste 0 (STARTKEY). Der Multiplexerausgang ist an den Schieberegistereingang anzuschließen (s. Pos. 2 in Abb. 25). Wenn das dauernde Niederhalten der Taste zu un bequem ist, kann ein Toggle-Flipflop zwischenschalten. Nach jedem Ändern der Zykluslänge Taste 3 betätigen (CLEARKEY; zum Rücksetzen).

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.

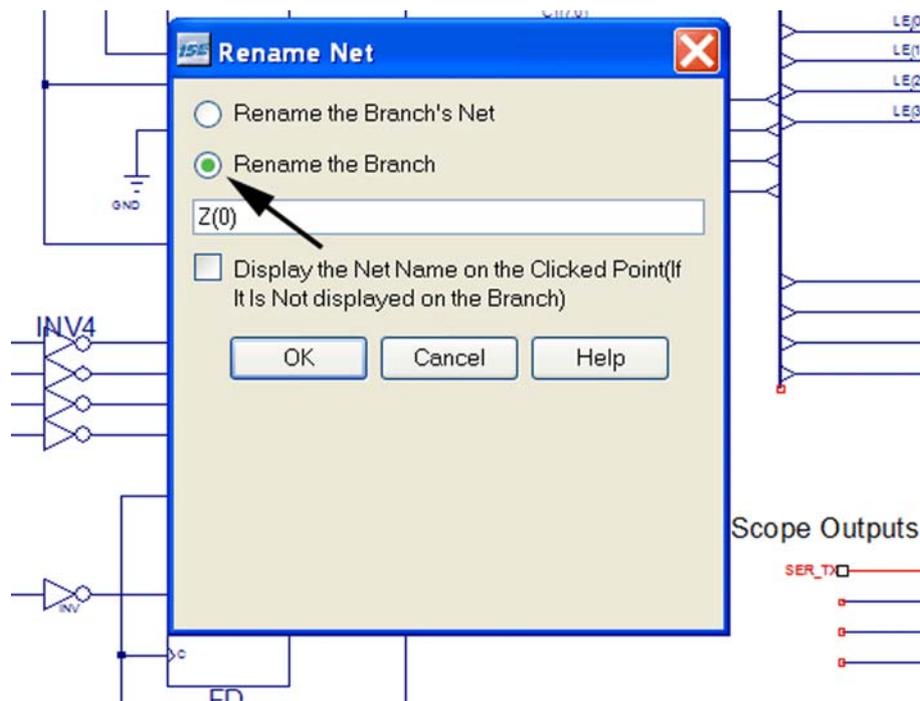


Abb. 29 Aufpassen. Wenn nur der eine Eingang umbenannt werden soll, unbedingt diese Variante auswählen – sonst geht's schief...

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*.

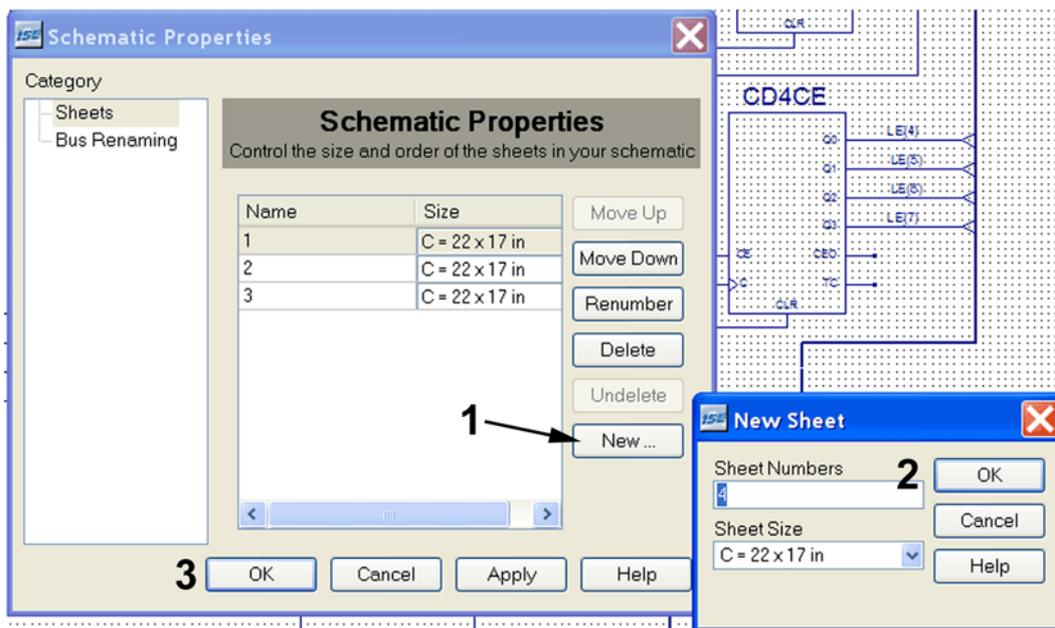


Abb. 30 Anfordern eines neuen Blattes. 1 – 2 – 3: in dieser Reihenfolge klicken. Ggf. zuvor Papierformat auswählen (*Sheet Size*).

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

Signalverlaufsbeispiele



Abb. 31 Zeichen 00H. 1 - Stopbit; 2 - Pause; 3 - Startbit; 4 - acht Datenbits (Nullen).

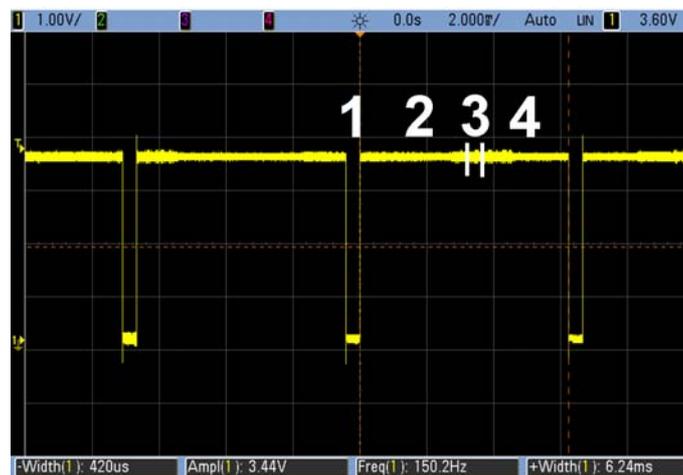


Abb. 32 Zeichen FFH. 1 - Startbit; 2 - acht Datenbits (Einsen); 3 - Stopbit; 4 - Pause.

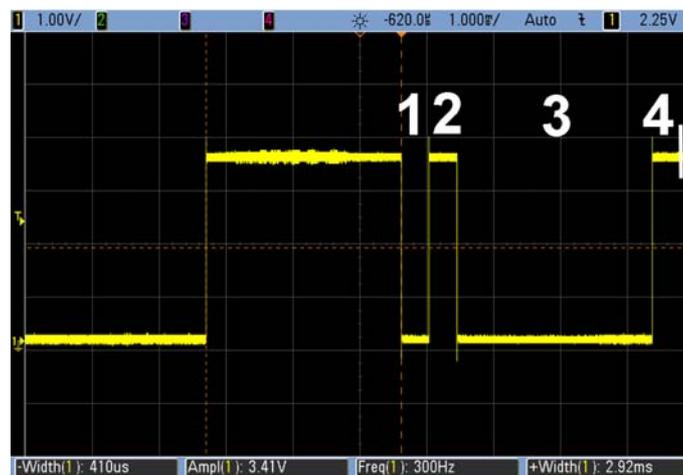


Abb. 33 Zeichen 01H = 0000 0001. Übertragungsreihenfolge 1000 0000. 1 - Startbit; 2 - Bit 0 = 1; 3 - die weiteren sieben Datenbits (Nullen); 4 - Stopbit.

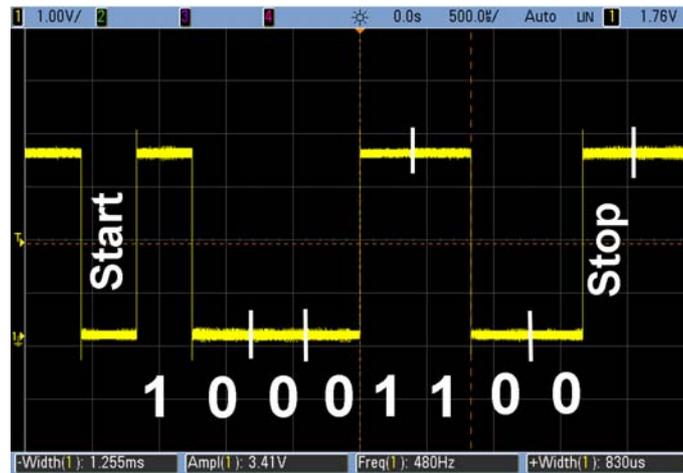


Abb. 34 Zeichen 31H = 0011 0001. Übertragungsreihenfolge 1000 1100.

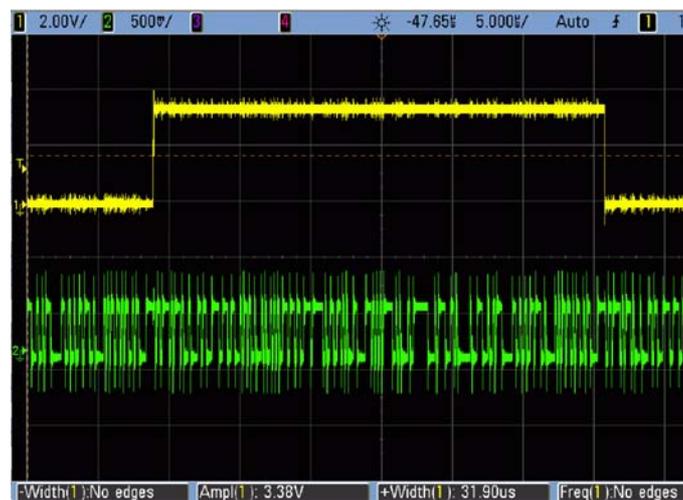


Abb. 35 Lineares rückgekoppeltes Schieberegister (LFSR). 8 Bits, maximale Zykluslänge. Oben das Synchronisationssignal (SCOPE_SYNC).



Abb. 36 Lineares rückgekoppeltes Schieberegister (LFSR). 8 Bits, verringerte Zykluslänge.