

# Praktikum Digitaltechnik SS 2014

## Versuch 3

Stand: 24. 6. 2015

### Versuchsplattform:

CPLD-Lehrgerät 12 mit CPLD Evaluation Board (Pollin). Eingebaute Ein- und Ausgabemittel: vier Taster, acht LEDs, serielle Schnittstelle. Zusätzlich Siebensegmentanzeige 09/13, Universaladapter 10b, Lüftertrainer 14a, XY-Adapter 09a (bedarfsweise). Entwicklungssoftware: Xilinx ISE 10.1.

### Aufgaben:

- Tageszeituhr.
- Stoppuhr.
- Universalzähler.

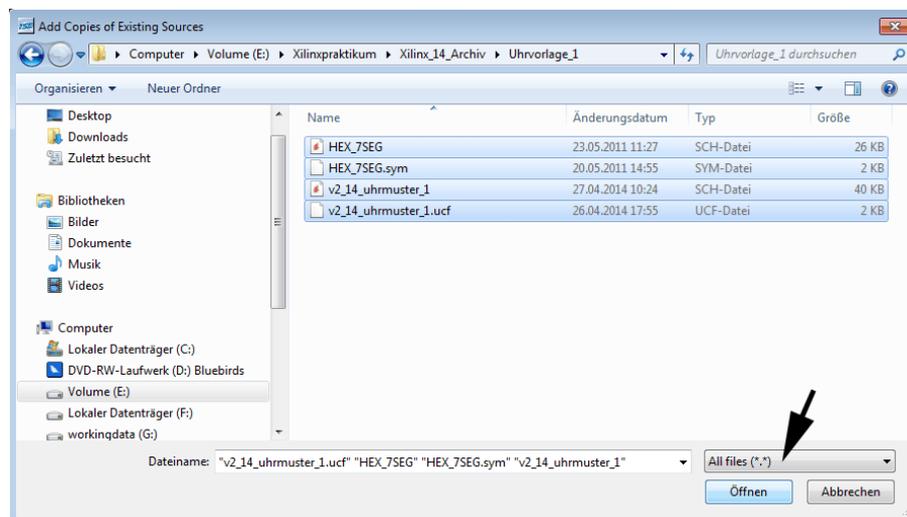
### Für Fortgeschrittene zusätzlich:

- Vorwärts und rückwärts zählen.
- Pulsweitenmodulation (PWM).
- Dreieckssignale erzeugen.

– Was nicht fertig wird, bleibt liegen. –

## Aufgabe 1: Tageszeituhr

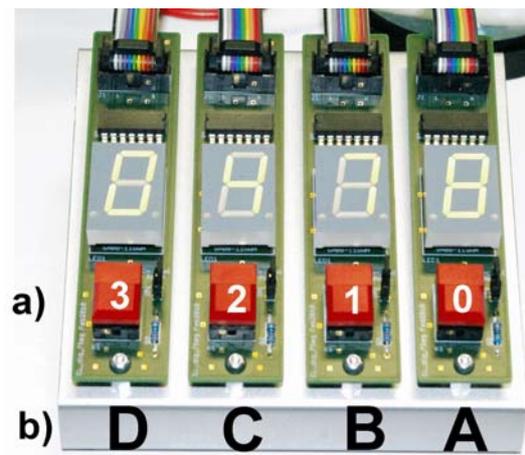
Wir nutzen alle vier Siebensegmentanzeigen aus. Unser Ziel: ein Uhr, die Stunden und Minuten anzeigt, von 00:00 bis 23:59. Die Erprobung würde allerdings etwas lange dauern. Deshalb steuern wir die Schaltung mit schnelleren Taktsignalen an. Wir beginnen ein neues Projekt. Es stehen Vorlagen zur Verfügung, die wir als Kopien übernehmen können (“Add Copy of Source”). Jetzt wollen wir etwas (für uns) Neues ausprobieren, nämlich eine vollsynchronen Zählerschaltung mit Dezimalzählern.



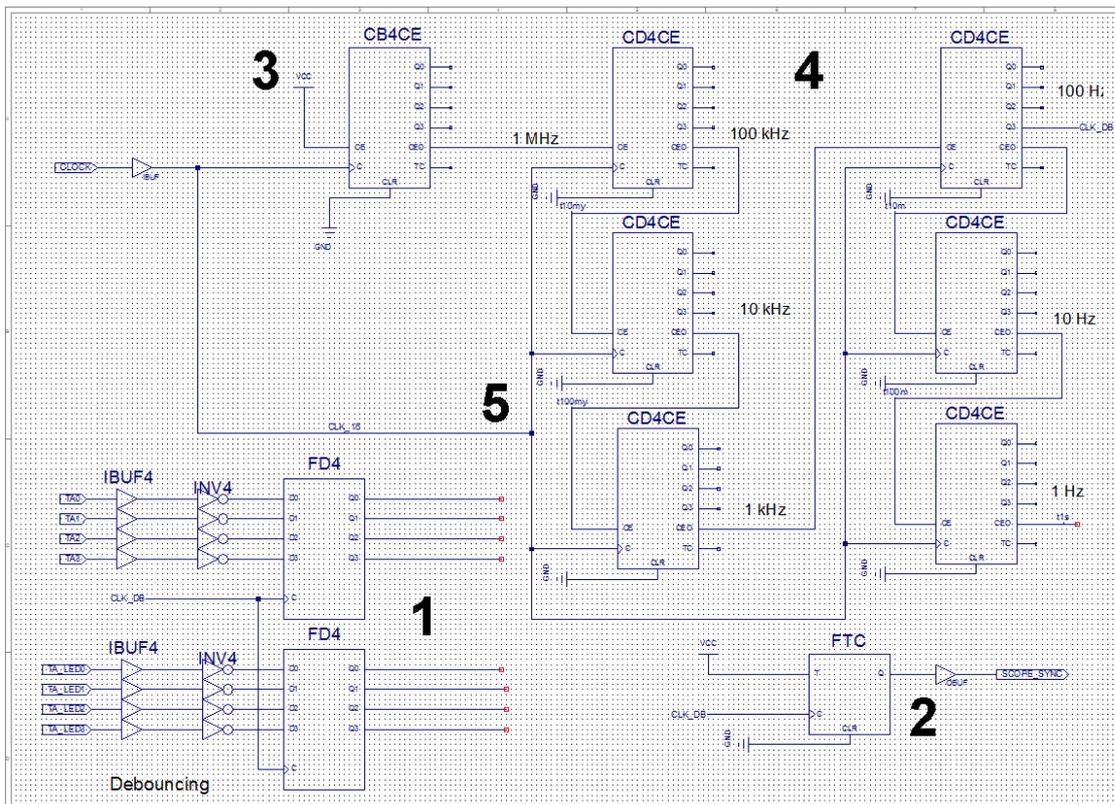
**Abb. 1** Neues Projekt. Die Quellen werden importiert. Zum Verzeichnis (auf der Netzfestplatte) s. die Anleitung des Versuchs 1. Wie das Unterverzeichnis heißen könnte, liegt nahe (z. B. “Uhrvorlage\_1”). Wir brauchen den Schaltplan, den Siebensegmentdecoder und die (vorgefertigte) Constraintsdatei. Hierzu müssen wir **alle** Dateien zu sehen bekommen (Pfeil).

### Hinweis:

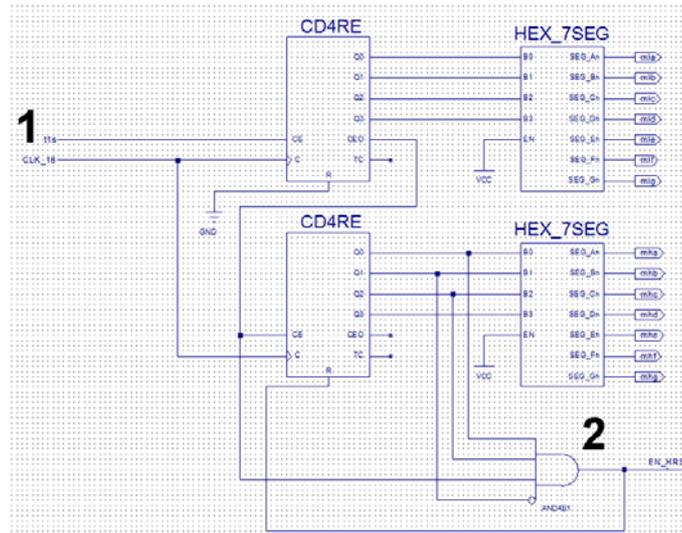
Zu den Belegungen der CPLD-Pins siehe die Constraintstabellen am Ende dieser Versuchsanleitung.



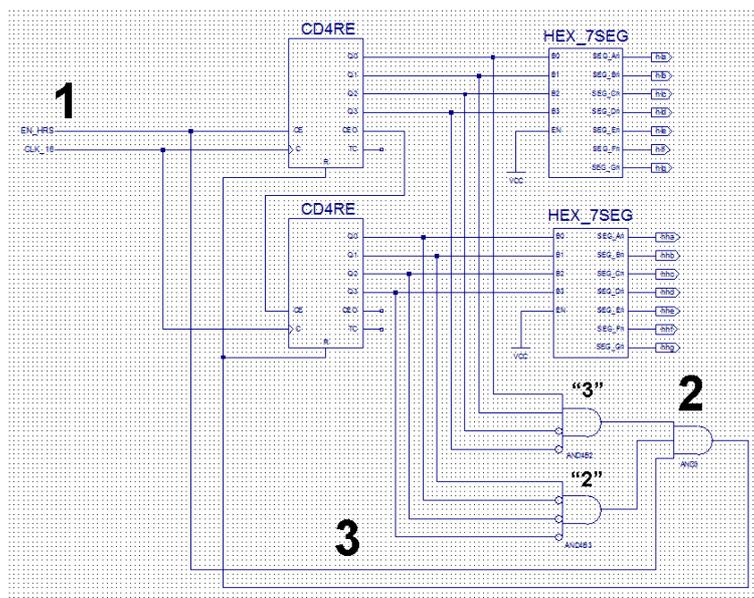
**Abb. 2** Anschließen der Siebensegmentanzeige 09/13. a) die laufenden Nummern der LED-Tasten (TA\_LED0 usw.). b) die Ports des CPLD-Lehrgerätes 12.



**Abb. 3** Die Grundschaltung. Blatt 1 = Frequenzbereitung und Tastenentprellung. 1 - Entprellen aller 8 Tasten der Versuchsanordnung. Oben die der Pollin-Platine, darunter die der Siebensegmentanzeige. 2 - Hilfsschaltung zur Signalanzeige mittels Oszilloskop. 3 - Binärzähler. Teilt die 16 MHz auf 1 MHz herunter. 4 - dezimale Teilerkette. Jeder Übertragsausgang (CEO) ergibt eine Frequenz, die jeweils 1/10 der Eingangsfrequenz (am Zählerlaubniseingang CE) beträgt. An der Teilerkette können so Impulsfrequenzen von 100 kHz bis zu 1 Hz (Sekundenimpulse) abgegriffen werden. 5 - der Grundtakt (16 MHz) ist an alle Zählerstufen geführt. Das Zählen wird lediglich über die Zählerlaubniseingänge (CE) gesteuert (vollsynchrone Betriebsweise).

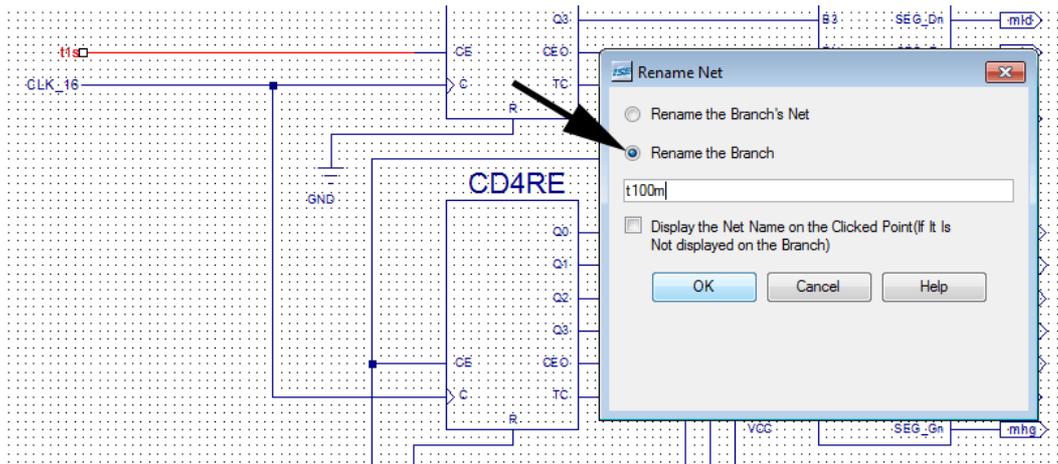


**Abb. 4** Die Grundsaltung. Blatt 2 = Minutenzähler. Die niederwertige Stelle zählt von 0 bis 9, die höherwertige von 0 bis 5. 1 - zur Probe zählen wir zunächst im Rhythmus von 1 s. 2 - Zählerlaubnissignal für den Stundenzähler. Es wird aktiv, wenn der Minutenzähler die Stellung "59" erreicht hat.



**Abb. 5** Die Grundsaltung. Blatt 3 = Stundenzähler. Er zählt dezimal von 0 bis 23. 1 - Zählerlaubnis vom Minutenzähler. 2 - Decodierung des Zählerstandes "23". Ist dieser Stand erreicht und aktiviert der Minutenzähler erneut die Zählerlaubnis, wird der Stundenzähler auf "00" zurückgesetzt. 3 - hierzu wird das Zählerlaubnissignal im Decoder konjunktiv verknüpft.

Beim Zählen im Sekundenrhythmus können wir mit etwas Geduld erkennen, wie die Minuten abgezählt werden und der Stundenzähler weiterschaltet. Um zu sehen, wie es in den Stundenstellen aussieht, ändern wir den Zählrhythmus auf 100 ms.



**Abb. 6** Das Zählerlaubnissignal des Minutenzählers wird umbenannt (Blatt 2). Aufpassen – "Rename the Branch"...

a)

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
73/144 (51%)	180/720 (25%)	45/144 (32%)	30/81 (38%)	140/432 (33%)

b)

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
69/144 (48%)	172/720 (24%)	41/144 (29%)	30/81 (38%)	124/432 (29%)

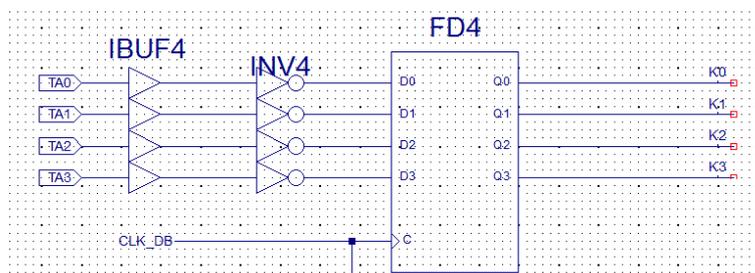
**Abb. 7** Ein lehrreicher Vergleich. Wir betrachten die "Fitting Reports" beider Varianten. a) mit Zählung alle Sekunde, b) mit Zählung alle 100 ms. Jetzt sind es 4 Zellen weniger. Weshalb? – Weil die Teilerstufe von 100 ms auf 1 s gar nicht mehr benötigt wird. Das Programm hat dies erkannt und die nutzlose Stufe einfach weggelassen...

**Durchlaufzeiten:**

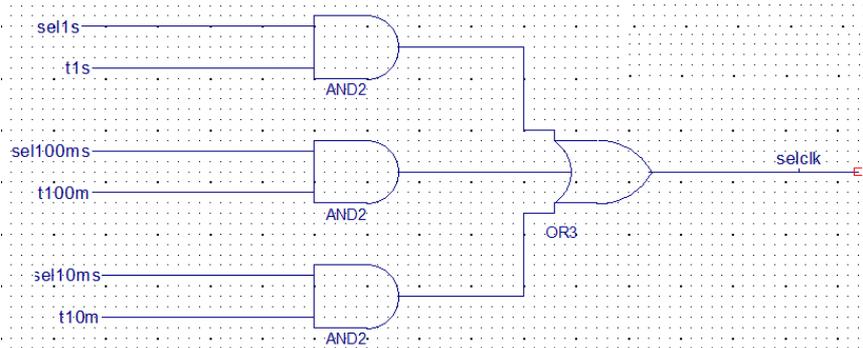
24 Stunden = 86 400 s. 1 min = 60 s.  $86\ 400 : 60 = 1440$ . Somit dauert der Durchlauf bei einem Zähltakt von

- 1 s: 1440 s = 24 min,
- 100 ms: 144 s = 2 min 24 s,
- 10 ms: 14,4 s.

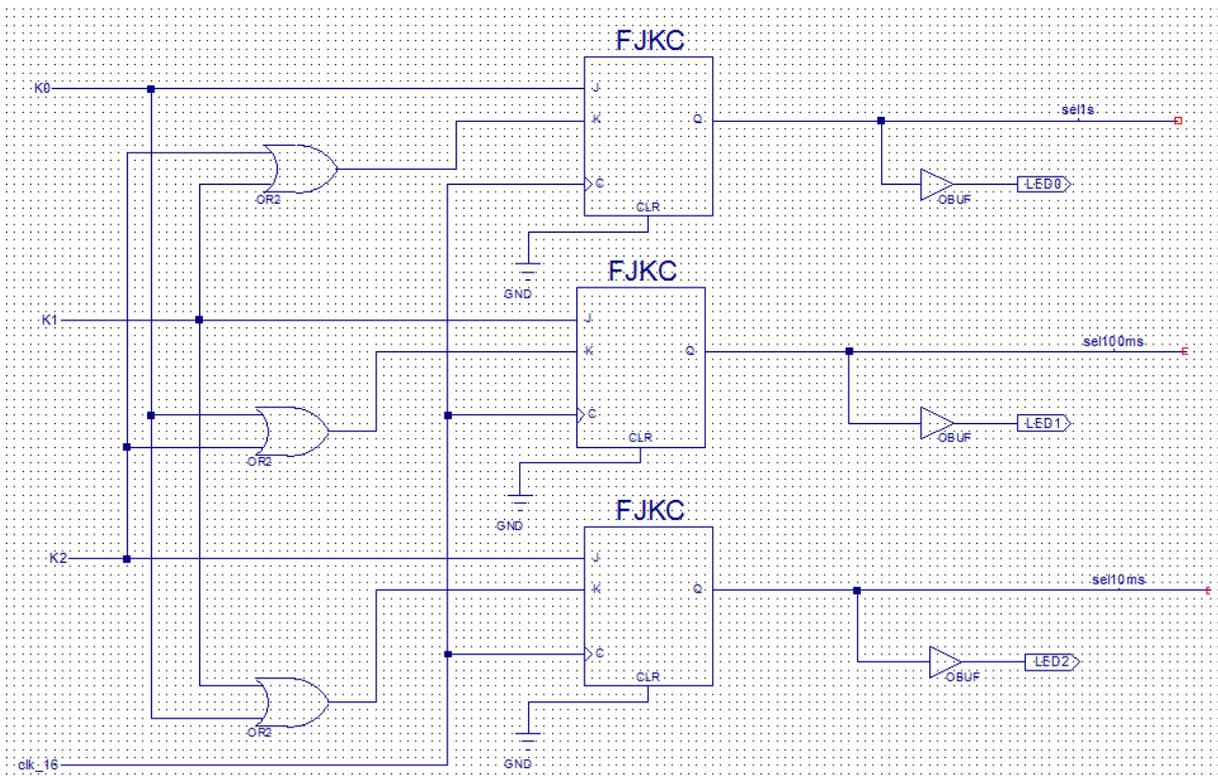
Jetzt soll die Uhr so umgebaut werden, daß sich die drei genannten Zählakte über Tasten auswählen lassen. Wir verwenden die Tasten 2, 1 und 0 auf der Pollin-Platine.



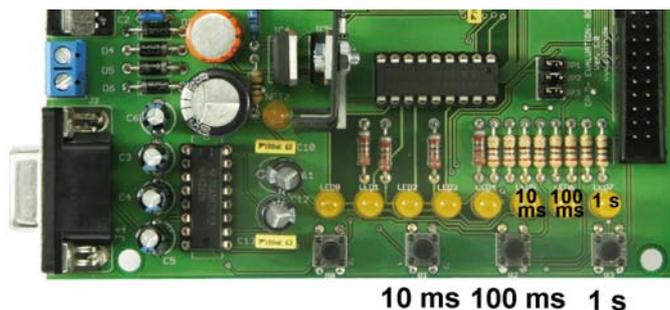
**Abb. 8** Die entprellten Tastensignale erhalten Namen (Blatt 1).



**Abb. 9** Über einen Datenselektor werden die Zählakte ausgewählt (findet auf Blatt 2 Platz).

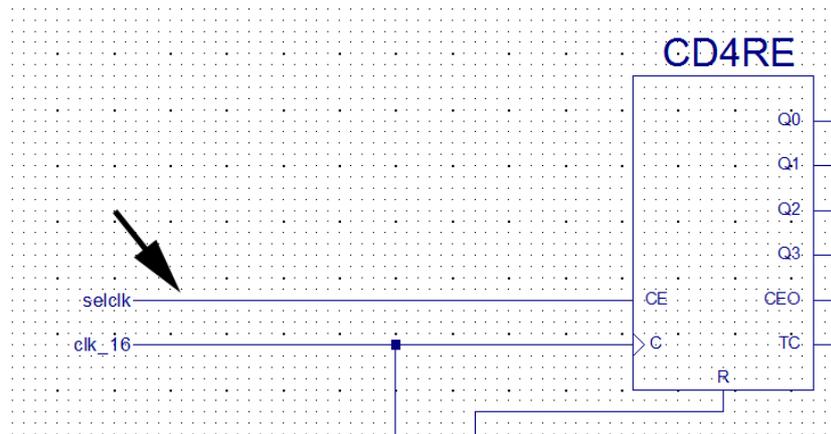


**Abb. 10** Wir wollen die Tasten nicht dauernd niederhalten. Deshalb werden die Tastenbetätigungen in Merklipflops gefangen. Das zur jeweiligen Taste gehörende Flipflop wird gesetzt, die anderen werden zurückgesetzt. Dabei verzichten wir auf die Berücksichtigung von Spitzfindigkeiten (z. B. daß mehr als eine Taste gleichzeitig gedrückt sein kann). Schaltung ggf. auf einem neuen Blatt. Für die LED-Anzeige muß die Constraintsdatei entsprechend ergänzt werden. LED0 = P41, LED1 = P40, LED2 = P39.



10 ms 100 ms 1 s

**Abb. 11** Tasten und Anzeigen auf der Pollin-Platine.



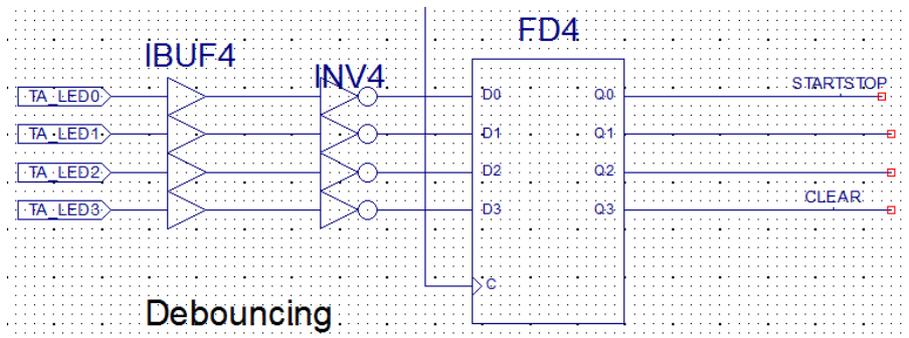
**Abb. 12** Der Zählerlaubniseingang des Minutenzählers wird passend umbenannt (Blatt 2).

## Aufgabe 2: Stoppuhr

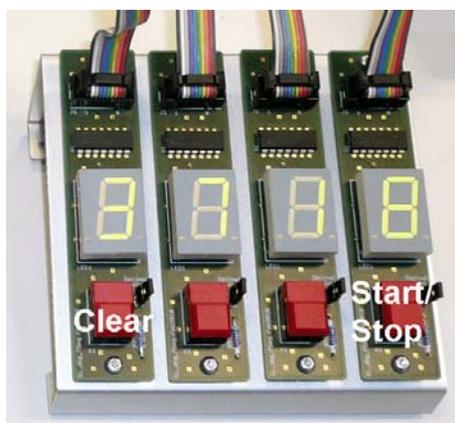
Die Stoppuhr soll in vier Ziffernstellen bis 99,99 s zählen, und zwar mit einer Auflösung von 1/100 s. Hierzu wird die Uhr im Schnellverfahren umgebaut. Alle Minuten- und Stundenzähler werden zu einfachen, durchlaufenden Dezimalzählern ohne zusätzliche Decodierung.

Bedienung:

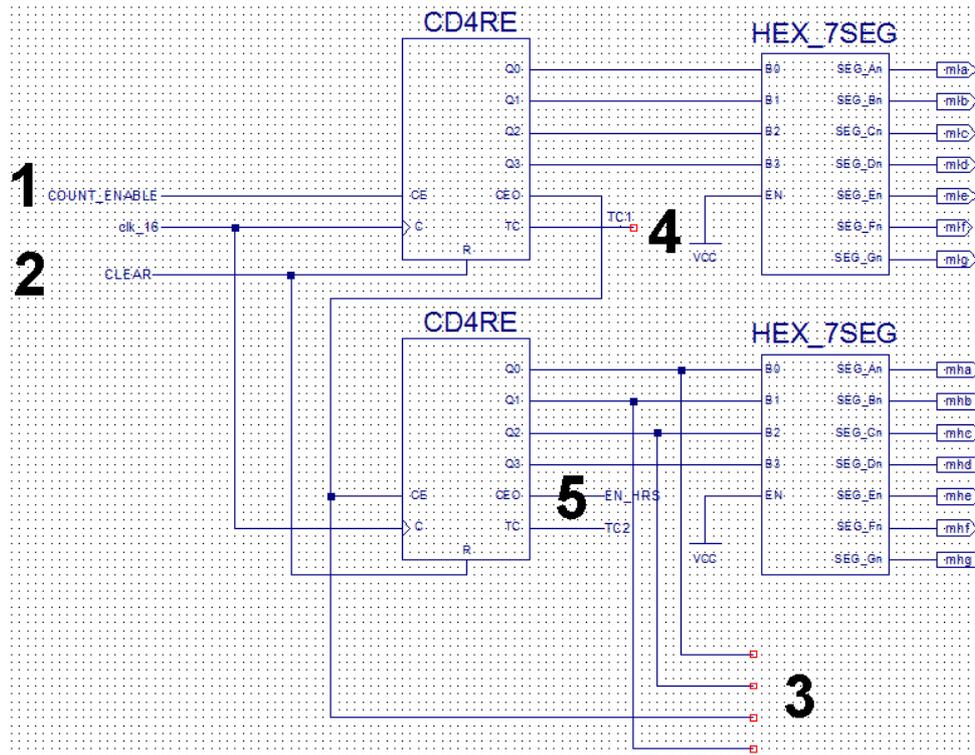
- Start und Stop: LED-Taste 0 (rechts außen).
- Löschen: LED-Taste 3 (links außen).



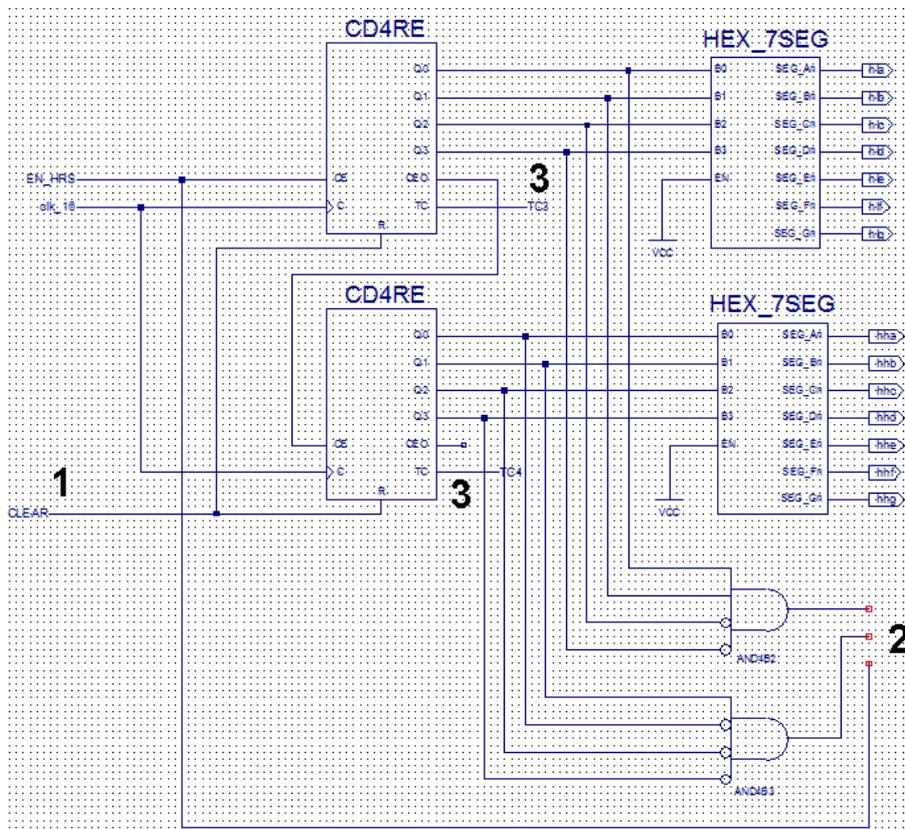
**Abb. 13** Benennung der LED-Tasten.



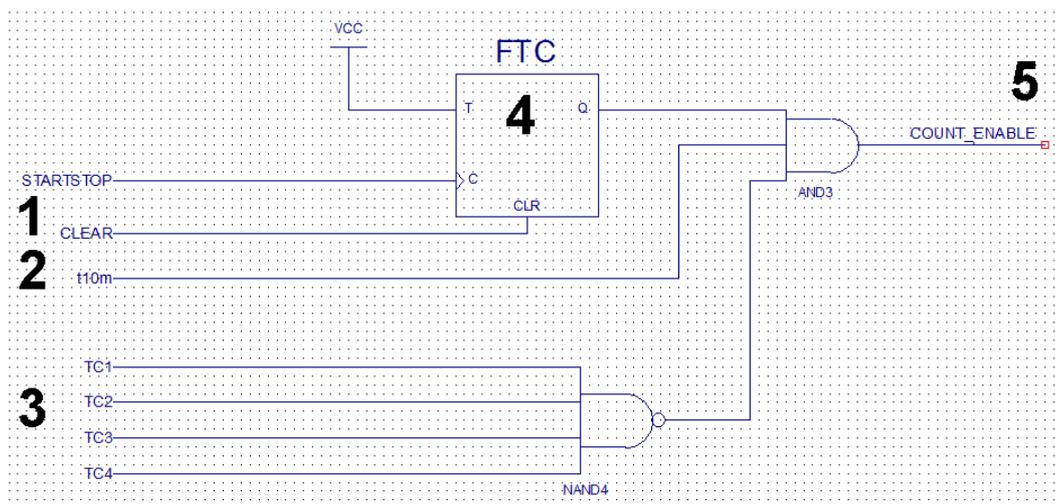
**Abb. 14** Die Bedientasten.



**Abb. 15** Umbau der Minutenzähler. 1 - neue Zählerlaubnis. 2 - Löshsignal. 3 - der Decoder wurde entfernt (einfach und schmucklos). 4 - die TC-Ausgänge der Zähler erhalten Namen. 5 - das Zählerlaubnissignal für den Stundenzähler (jetzt Sekundenzähler).



**Abb. 16** Umbau der Stundenzähler. 1 - Löshsignal. 2 - der Decoder wurde entfernt (einfach und schmucklos). 3 - die TC-Ausgänge der Zähler erhalten Namen.

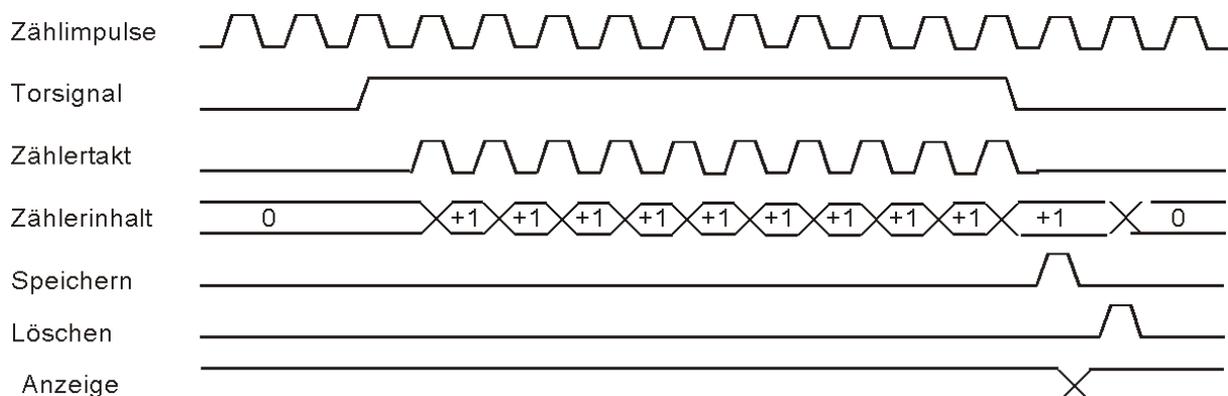


**Abb. 17** Die Steuerschaltung (ggf. neues Blatt). Gleiches Prinzip wie bei der Stoppuhr im Versuch 2.. Nur werden die Zählsignale von dezimalen Teilerstufen geliefert, so daß nichts zu decodieren ist. 1 - Löschsingal. 2 - wir zählen mit 100 Hz bzw. im Rhythmus von 10 ms. 3 - Zählendesignale (TC-Ausgänge) der vier Zähler. 4 - Start-Stop-Umschaltung mittels Toggle-Flipflop. 5 - Zählerlaubnissingal.

*Hinweis:* Die Schaltung von Abb. 10 kann bleiben. Sie wird nur nicht genutzt. Wird sie gelöscht, müssen die Anschlüsse der LEDs aus der Constraintsdatei entfernt werden. 1. Möglichkeit: die Constraintsdatei aus dem Projekt entfernen (Source – Remove) und die Datei der Vorlage erneut einbauen (Add Copy of Source). 2. Möglichkeit: die Constraintdatei als Textdatei aufrufen (Edit – Preferences – Text Editor) und die Einträge löschen.

### Aufgabe 3: Universalzähler

Universalzähler im eigentlichen Sinne sind Meßgeräte, die Frequenzen, Periodendauern usw. messen können. Solche Meßaufgaben sind aber auch im Rahmen von Anwendungslösungen zu erledigen. Wenn es um höhere Frequenzen geht und es womöglich exakt auf den einzelnen Takt ankommt, muß die Meßaufgabe schaltungstechnisch gelöst werden; Mikrocontroller sind dann nicht geeignet.



**Abb. 18** Zum Wirkprinzip des Universalzählers. Ein Torsingal bestimmt, wann gezählt wird. In dieser Zeit gelangen Zähltake zu einem Meßzähler. Nachdem das Torsingal inaktiv geworden ist, wird der Zählerinhalt zwecks Anzeige (oder weiterer Auswertung) gespeichert. Dann wird der Zähler gelöscht, und das Zählen kann von neuem beginnen.

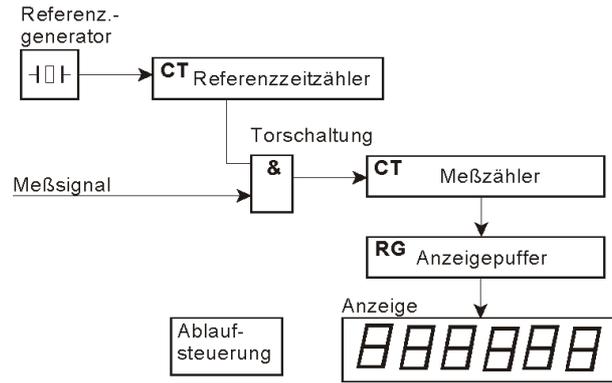


Abb. 19 Universalzähler im Blockschaltbild.

Wir haben nur eine vierstellige Anzeige, also bauen wir auch nur einen vierstelligen Zähler. Hierzu wird die bisherige Schaltungslösung (der Uhr) entsprechend umgebaut. Damit es übersichtlich bleibt, beginnen wir aber besser mit einem neuen Projekt. Die ursprünglichen Dateien stehen im Verzeichnis "Zaehlvorlage\_1". Neues Projekt anlegen und Quellen hineinkopieren ("Add Copy of Source").

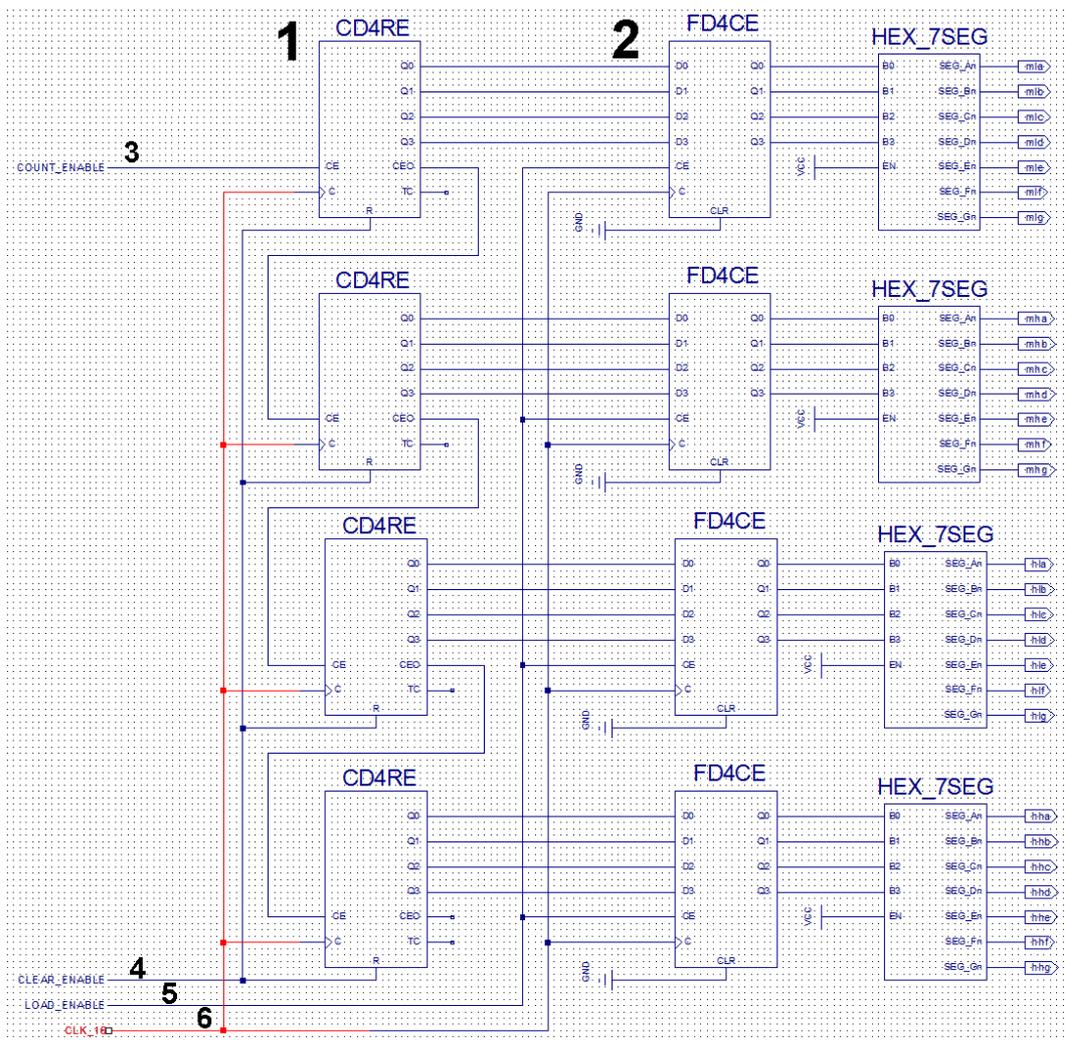
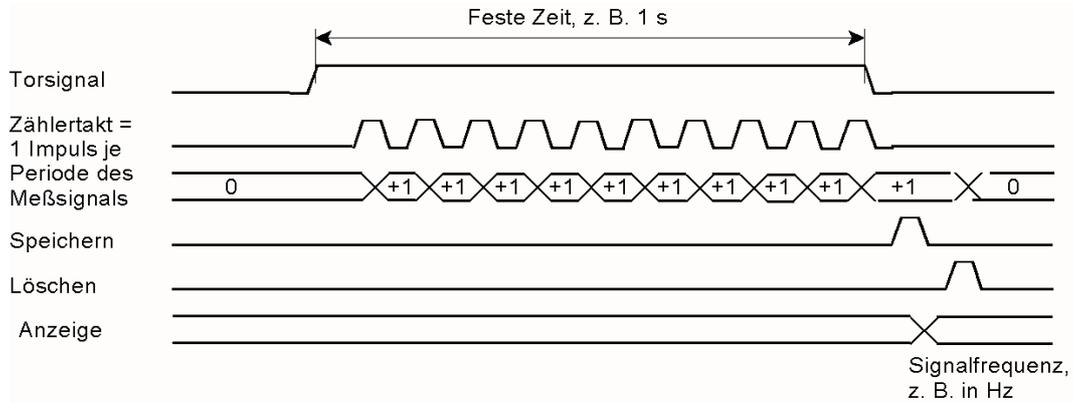


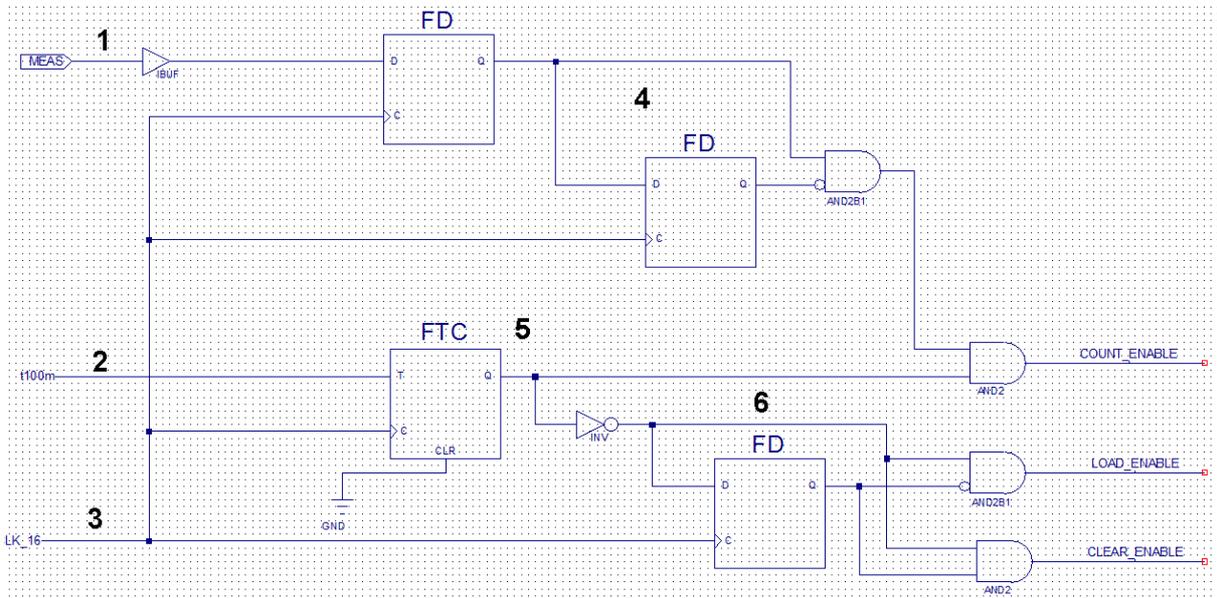
Abb. 20 Die gesamte Meßzähleranordnung ist auf einem Blatt dargestellt. 1 - Meßzähler; 2 - Anzeigepuffer; 3- Zählerlaubnis; 4 - Löscherlaubnis; 5 - Übernahmeerlaubnis; 6 - Grundtakt. Es soll eine vollsynchronen Schaltung werden. Alle Flipflops schalten mit dem Grundtakt von 16 MHz.

### Der Frequenzzähler

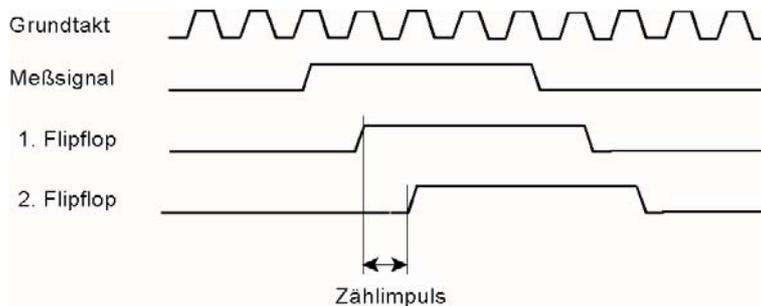
Wenn wir die Torschaltung eine Sekunde lang offenhalten und währenddessen das Eingangssignal auf den Zähler wirken lassen, zählt er alle Eingangsimpulse (= Signalperioden) je Sekunde, zeigt also die Signalfrequenz direkt in Hz an. Die Meßzeit (1 s) ist allerdings recht lang. Deshalb verkürzen wir sie auf  $1/10 \text{ s} = 100 \text{ ms}$ .



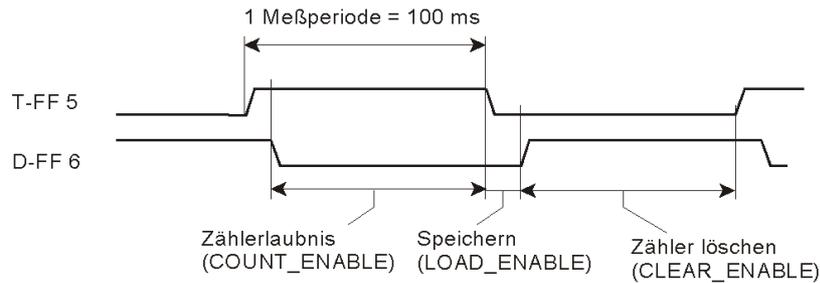
**Abb. 21** Das Prinzip der Frequenzmessung. Zählertakt = Meßsignal, Torsignal = feste Öffnungszeit.



**Abb. 22** Die Steuerschaltung. Die Ablaufsteuerschaltungen der professionellen Universalzähler haben es in sich. Hier geht es aber nur ums Prinzip. 1 - Meßsignal; 2 - Torsignal (100 ms); 3 - Grundtakt; 4 - Zählimpulserzeugung; 5, 6 - Ablaufsteuerkette.



**Abb. 23** Aus jeder Meßsignalperiode ist ein Zählimpuls zu gewinnen. Eine einfache Schiebekette (die Flipflops 4 in Abb. 20) leistet das Gewünschte. Es ist eine bewährte Grundschaltung zur Flankenerkennung und als elementarer Sequencer (Single-Shot).

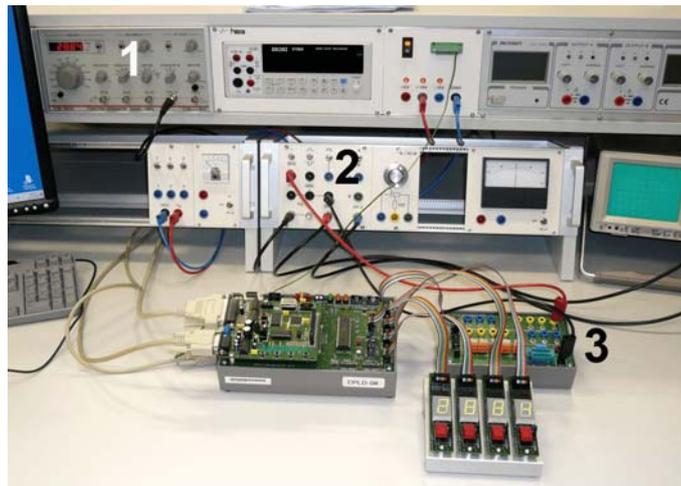


**Abb. 24** Zur Wirkungsweise der Ablaufsteuerung. Auf Leistung kommt es hier nicht an. Deshalb zählen wir nur in jedem zweiten Zeitintervall. 100 ms zählen, dann im Anzeigepuffer speichern, dann während des Rests der zweiten 100 ms den Zähler löschen. Die Flipflops 6 und 7 (abb. 20) wirken als Schiebekette ähnlich wie der Flipflops 4.

*Hinweis:* Das Meßsignal MEAS liegt am CPLD-Pin 42 an. Dieses Pin ist auch mit der Taste KEY3 verbunden (eine Eigenheit des CPLD-Moduls). Diese Taste wird deshalb hier nicht genutzt. Sie läßt sich auch nicht in die Constraintsdatei eintragen. Nicht betätigen! (Denn sonst wird MEAS auf Masse gezogen.)

### Funktionsprüfung:

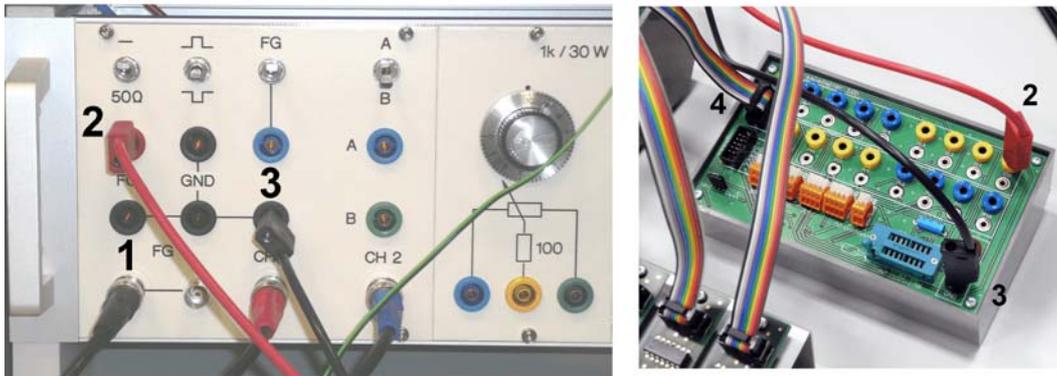
Das Meßsignal kommt vom Toellner-Funktionsgenerator des Laborplatzes. Es wird über den Meßadapter 09b und den Universaladapter 10b eingespeist.



**Abb. 25** Die Versuchsanordnung. 1 - Funktionsgenerator; 2 - Meßadapter 09b; 3 - Universaladapter 10b (genannt die Bunte Kuh).



**Abb. 26** Die Impulse kommen vom Synchronsignalausgang des Funktionsgenerators. Sie haben eine passende Amplitude (TTL-Pegel).



**Abb. 27** So anschließen: 1 - vom Funktionsgenerator; 2 - Signalkabel; 3 - Massekabel. 4 - Bandkabel von Port B Universaladapter nach Port E CPLD-Lehrgerät.

**Der Frequenzbereich:**

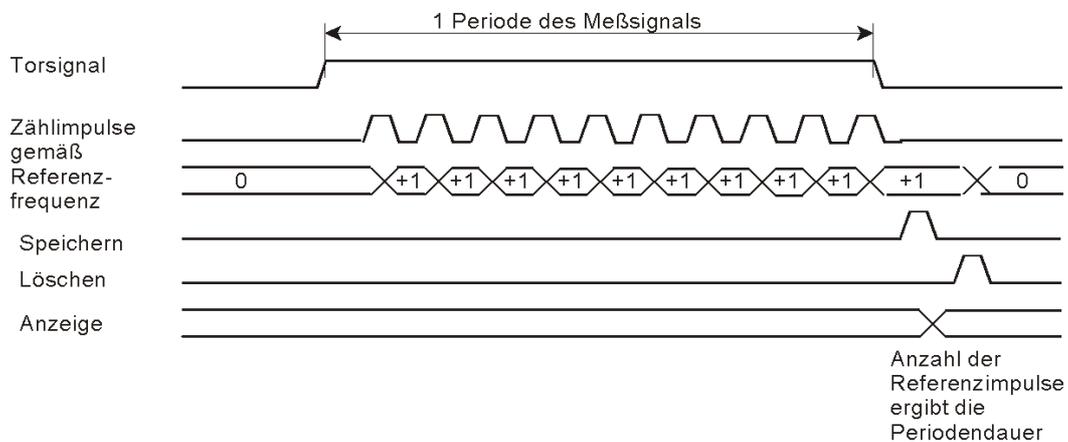
In 100 ms kann bis 9999 gezählt werden. Also ca. 10 000 Impulse / 100 ms oder 100 000 Impulse / s. Die höchste Signalfrequenz, die wir messen können, ist somit 99,99 kHz. Die niedrigste meßbare Signalfrequenz liegt vor, wenn in 100 ms ein einziger Impuls auftritt. Das entspricht 10 Impulsen/s = 10 Hz.



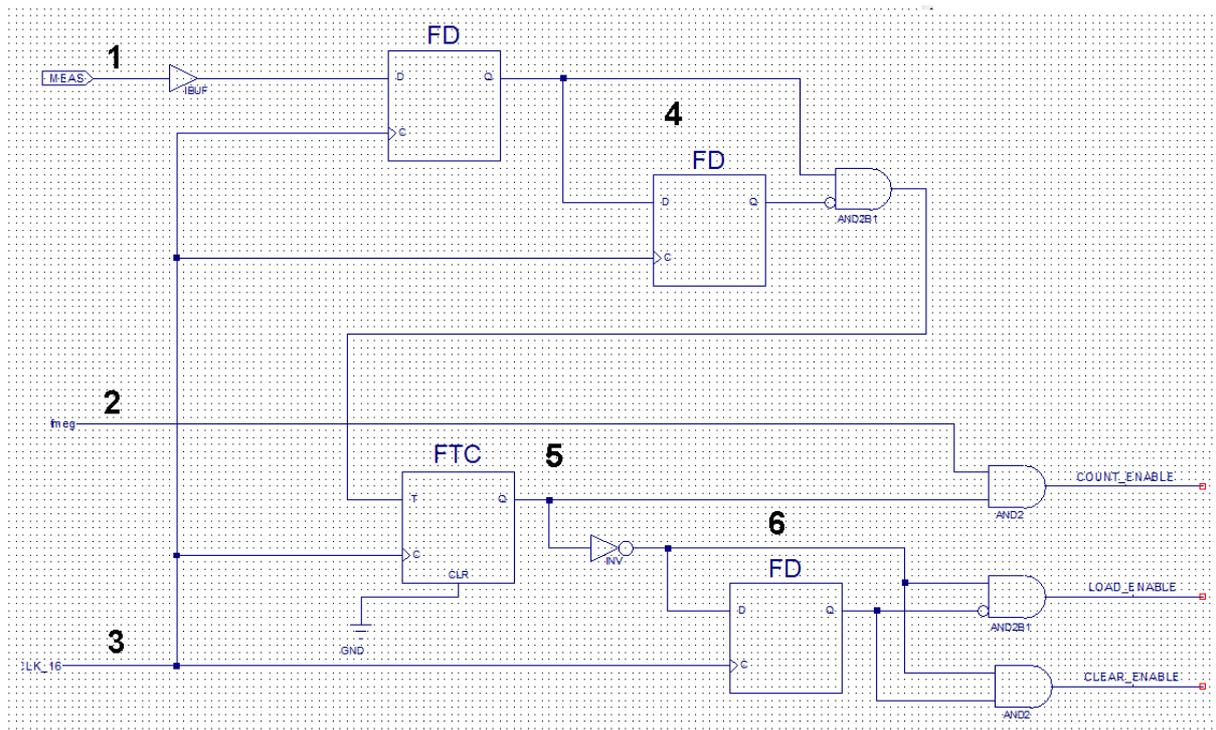
**Abb. 28** Die Frequenzmessung funktioniert offensichtlich...

**Die Periodendauermessung**

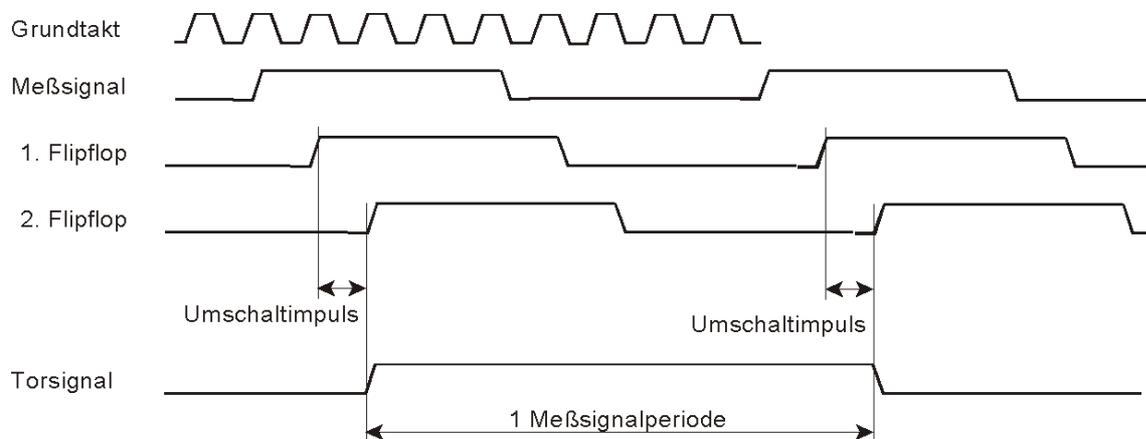
Das Meßsignal öffnet die Torschaltung und läßt Zählimpulse einer bestimmten Referenzfrequenz zum Zähler gelangen. Wenn die Referenzfrequenz 1 MHz beträgt und wenn der Meßimpuls 5 ms lang aktiv ist, dann gelangen 5000 Zähltake zum Zähler.



**Abb. 29** Das Prinzip der Periodendauermessung. Wir messen mit 1 MHz. Das Meßsignal hält die Torschaltung für jeweils eine Signalperiode offen, so daß alle  $\mu$ s ein Impuls auf den Zähler gegeben wird.



**Abb. 30** Die Steuerschaltung<sup>1</sup>. 1 - Meßsignal; 2 - Zählimpulse (1 MHz); 3 - Grundtakt; 4 - Steuerimpulserzeugung für die Torschaltung; 5, 6 - Ablaufsteuerkette.



**Abb. 31** Zur Wirkungsweise der Ablaufsteuerung. Auch hier zählen wir nur in jeder zweiten Periode des Meßsignals. In der jeweils anderen Periode wird (am Anfang) der Zählwert gespeichert, anschließend wird der Zähler gelöscht. Die Ablaufsteuerkette wirkt genauso wie bei der Frequenzmessung, nur wird sie nicht vom Referenzzeitähler, sondern vom Meßsignal geschaltet. Der Zähltakt selbst kommt von innen, ist also bereits synchron.

### Der Meßbereich:

Die Schaltung kann 9999  $\mu\text{s}$  abzählen. Eine so lange Periode entspricht einer Meßfrequenz  $> 100$  Hz. Die kürzeste meßbare Periodendauer liegt vor, wenn nur ein einziger Impuls gezählt wird. Das entspricht einer Meßfrequenz von 1 MHz. Wegen der Toleranzen beim Synchronisieren sind so kleine Zählwerte aber praktisch unbrauchbar.

1: Praxistip: Verbindungen löschen und alles neu zeichnen.

### Das Kombinationsgerät

Es ist eigentlich nur eine Frage der Umschaltung. Bei der Frequenzmessung kommen die Zählimpulse vom Meßsignal und das Torsignal von innen, bei der Periodendauermessung ist es umgekehrt. Deshalb sollen jetzt zwei Tasten (der Pollin-Platine) genutzt werden, um zwischen den Betriebsarten umzuschalten. Die Umschaltung erfolgt über Multiplexer. Die jeweilige Betriebsart wird mit je einer LED angezeigt. Nach dem Einschalten ist zunächst die Periodendauermessung eingestellt.

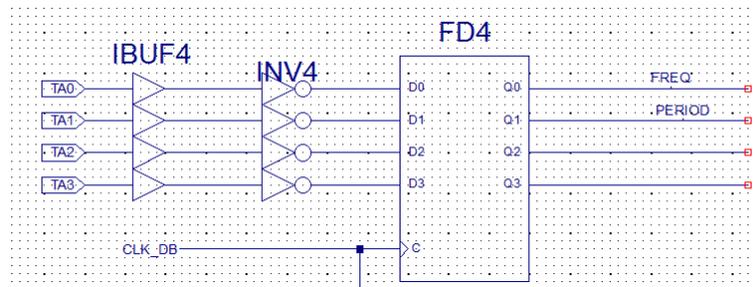
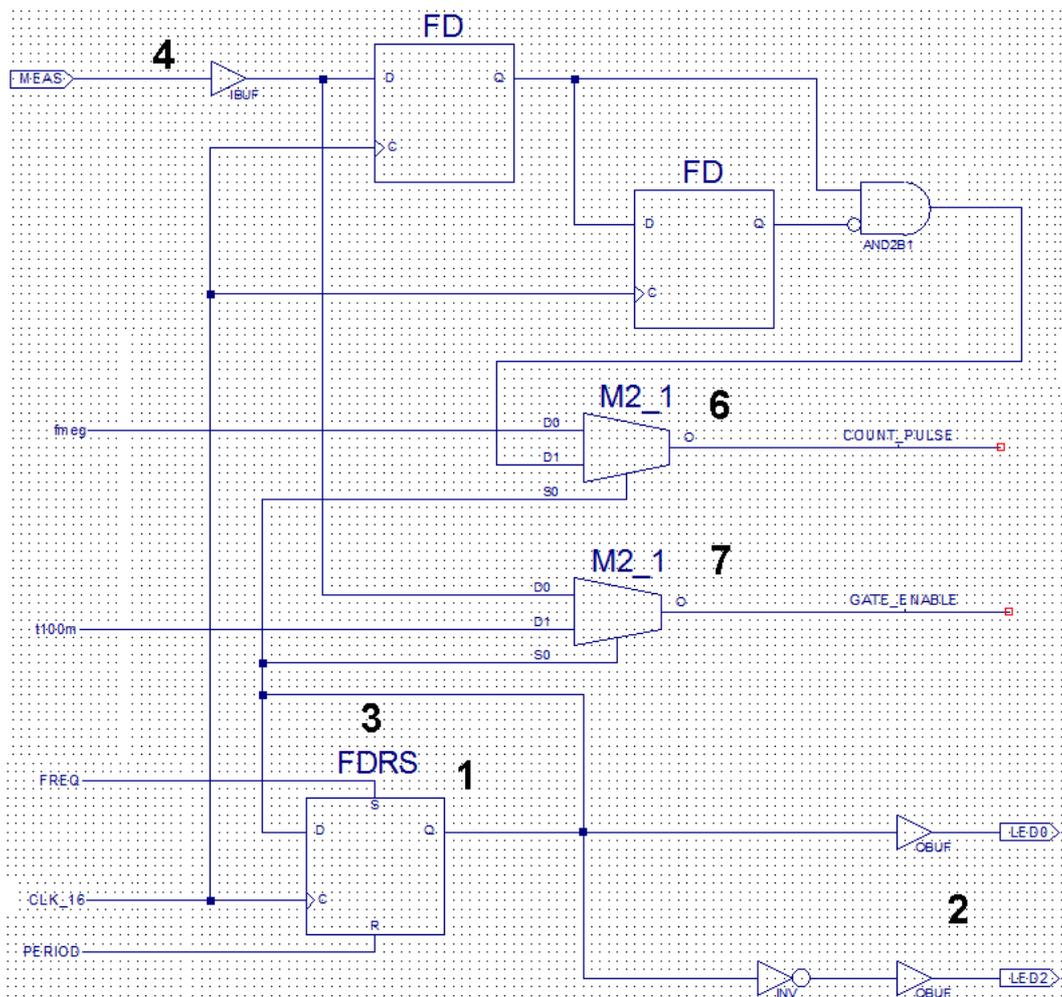


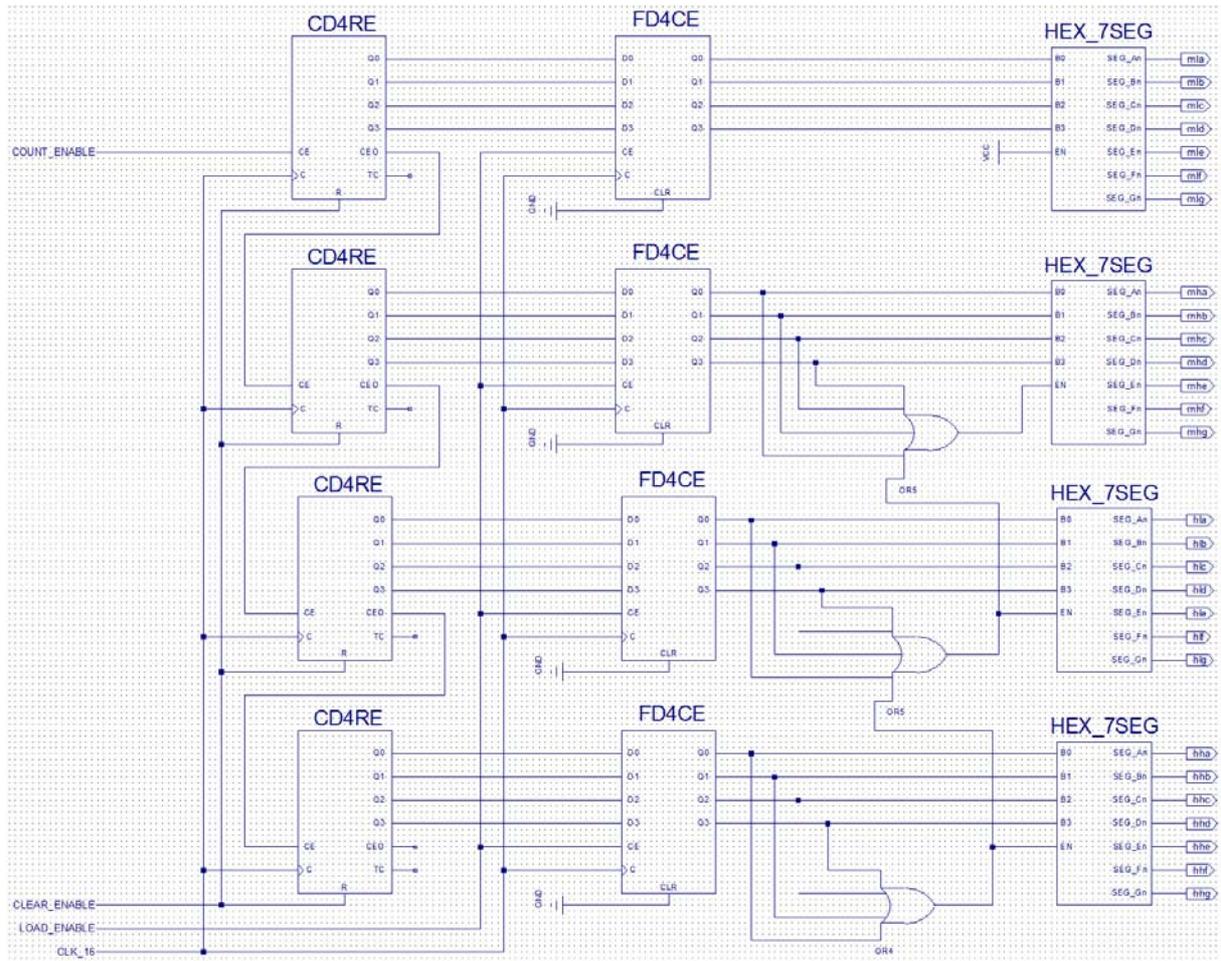
Abb. 32 Die Umschaltsignale.



(Neues Blatt)

**Abb. 33** Die Betriebsartenumschaltung. 1 - Merkflipflop für Betriebsart; 2 - Betriebsartenanzeige; 3 - Selbsthaltung; 4 - Meßsignal; 5 - Meßimpulsbildung (Flankenerkennung; Single-Shot); 6 - Zählimpulsauswahl (Meßsignalimpulse oder 1 MHz); 7 - Torsignalauswahl (Meßsignal oder 100 ms). Für die LED-Anzeige muß die Constraintsdatei entsprechend ergänzt werden. LED0 = P41, LED2 = P39.

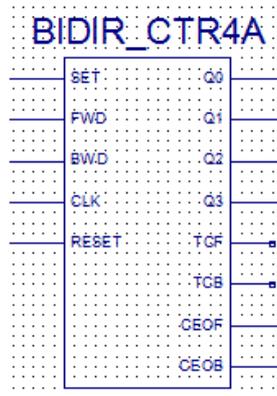




**Abb. 36** So lassen sich führende Nullen unterdrücken.

## Aufgabe 4: Vorwärts und rückwärts zählen

Das ist die Vorübung für die weiteren Aufgaben. Wir brauchen einen Zähler, der vorwärts und rückwärts zählen kann. Ein entsprechender 4-Bit-Zähler ist bereits vorgefertigt. Eine neues Projekt anlegen und die Quellen hineinkopieren ("Add Copy of Source"). Die ursprünglichen Dateien stehen im Verzeichnis "FWD\_BWD\_Vorlage\_1".

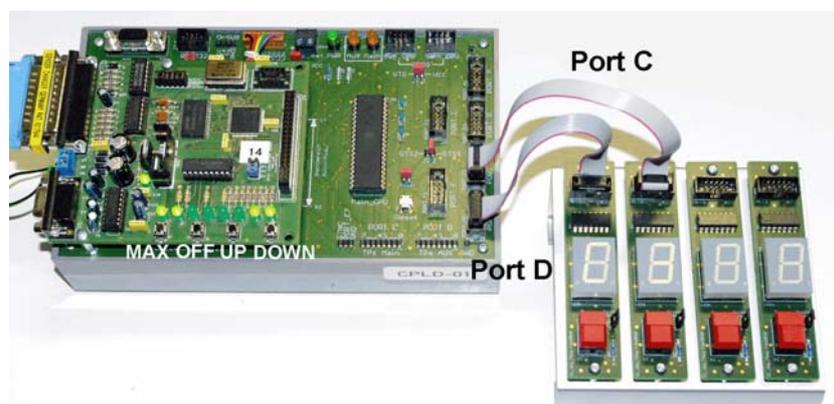


SET:	Alle Flipflops setzen.	Q0...Q3:	Zählausgänge.
FWD:	Vorwärts zählen.	TCF:	Endstellung (FH) beim Vorwärtszählen.
BWD:	Rückwärts zählen.	TCB:	Endstellung (0H) beim Rückwärtszählen.
CLK:	Takt	CEOF:	Zählübertrag beim Vorwärtszählen.
RESET:	Alle Flipflops löschen.	CEOB:	Zählübertrag beim Rückwärtszählen.

**Abb. 37** Der Vorwärts-Rückwärts-Zähler. Alle Eingänge wirken synchron zum Takt.

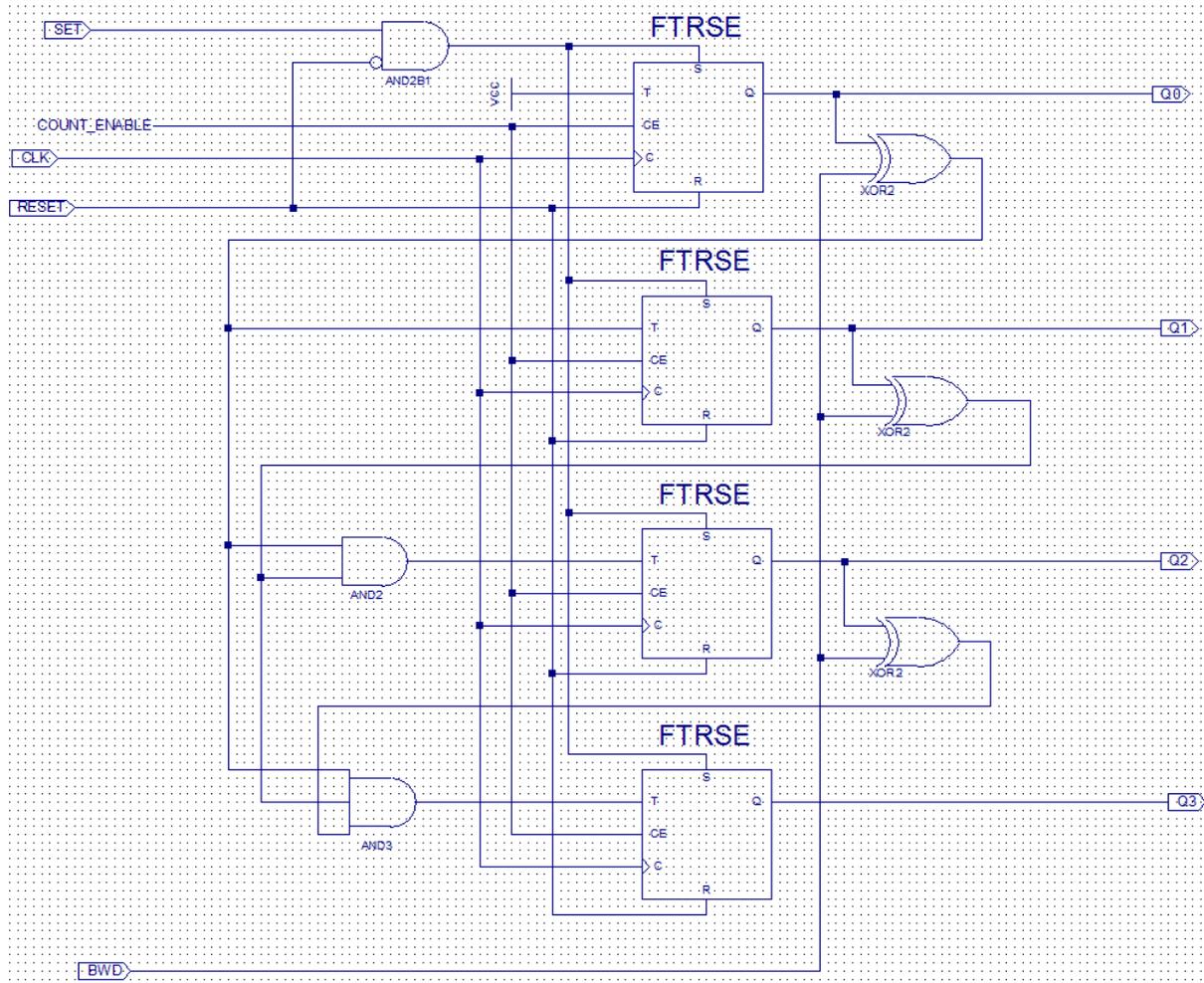
### Erprobung:

Mit einem langsamen Zähltakt und den folgenden Tastenwirkungen. Zwei Siebensegmentanzeigen anschließen. Das Zählen an den LEDs der Pollin-Platine und an den Siebensegmentanzeigen beobachten.

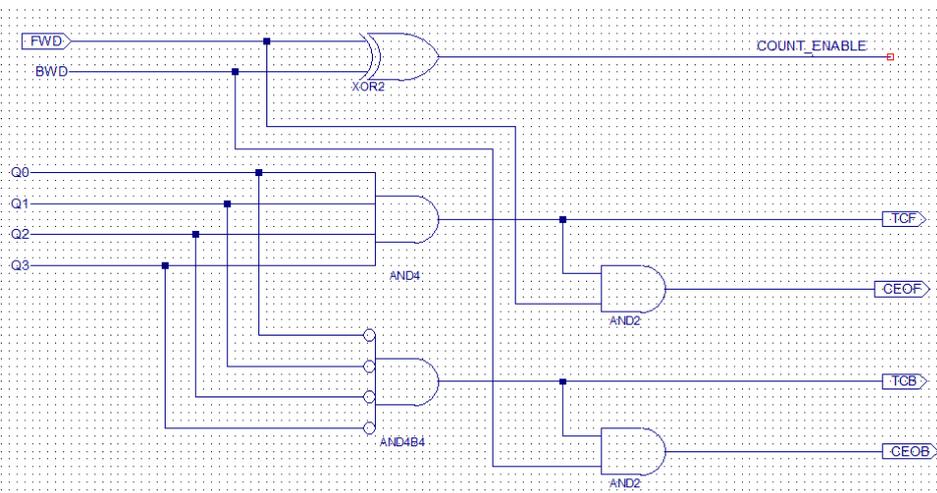


- Taste 0: rückwärts zählen (DOWN).
- Taste 1: vorwärts zählen (UP).
- Taste 2: löschen (alles Null = Aus; OFF).
- Taste 3: setzen (alles Eins = volle Leistung; MAX).

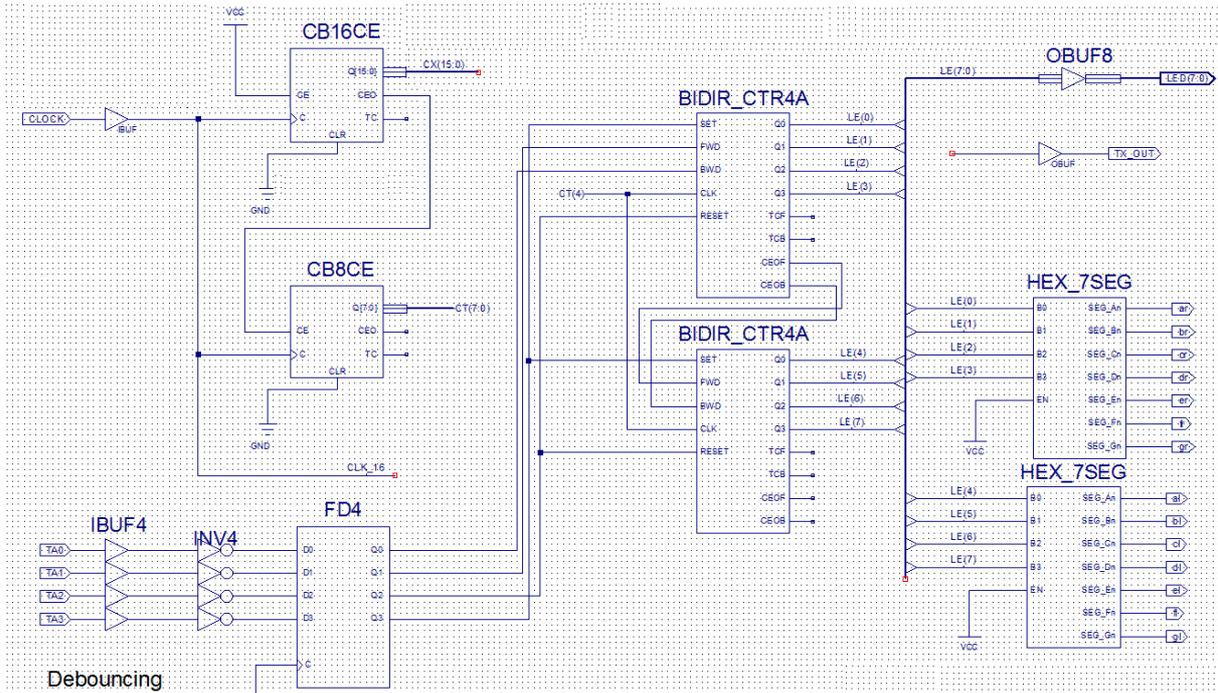
**Abb. 38** Der Versuchsaufbau. Die Tastenbezeichnungen beziehen sich teils schon auf die nächste Aufgabe.



**Abb. 39** Ein Blick ins Innere. Die synchrone Zähler-schaltung. Mit T-Flipflops wird sie besonders einfach.



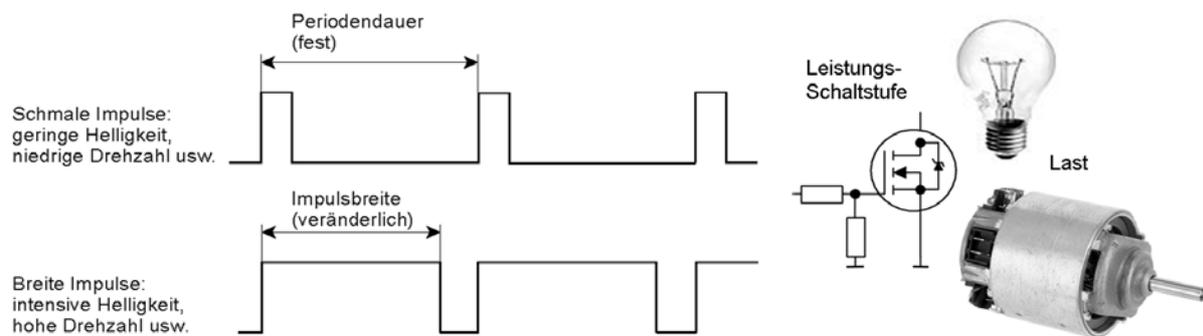
**Abb. 40** Die Zählersteuerung.



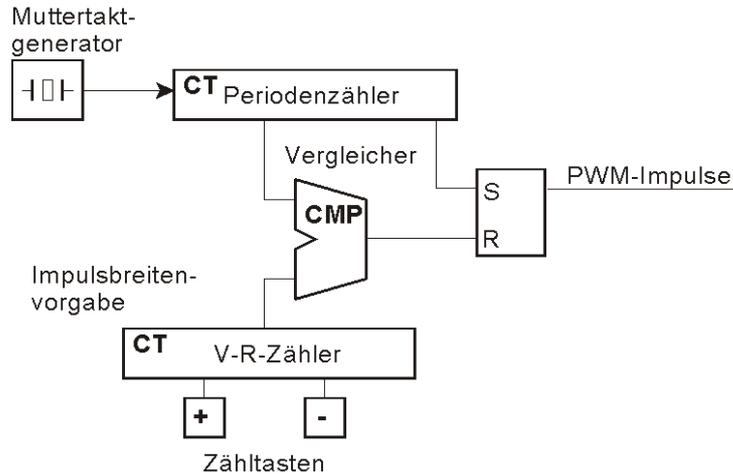
**Abb. 41** Zum Erproben nutzen wir die Grundschialtung des Versuchs 2, aber mit zwei eingebauten Vorwärts-Rückwärts-Zählern. Sie sind als 8-Bit-Zähler verschaltet.

### Aufgabe 5: Pulsweitenmodulation (PWM)

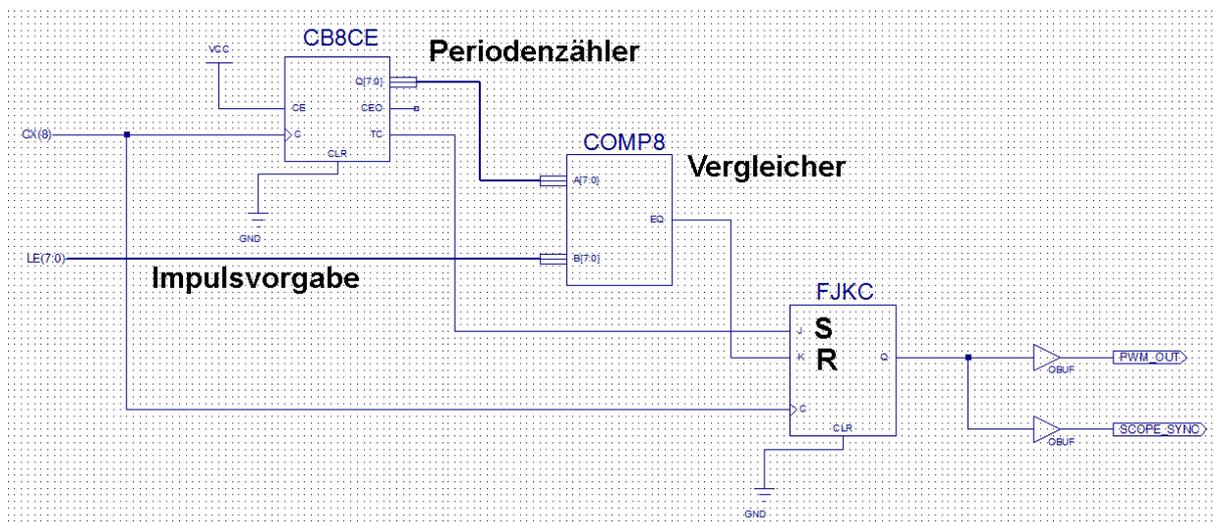
Die Pulsweitenmodulation dient dazu, kontinuierliche Größen, wie die Helligkeit einer Lichtquelle, die Drehzahl eines Motors oder die Amplitude einer Spannung, auf rein digitale Weise einzustellen, also durch einfaches Ein- und Ausschalten. Hierzu werden Impulsfolgen mit fester Periodendauer, aber veränderlicher Breite abgegeben. Je breiter der Impuls, desto höher die Helligkeit, Drehzahl usw. Gar kein Impuls ergibt den Aus-Zustand. Eine ständige Erregung (Dauersignal) ergibt den jeweiligen Maximalwert der Helligkeit, Drehzahl usw. (“Vollgas”). Die kontinuierliche Wirkung bei impulsweiser Erregung ergibt sich infolge von Integrationswirkungen. Bei Lichtquellen integriert das Auge des Betrachters (eine entsprechend angesteuerte LED flimmert im Takt der Erregung, wir können das aber nicht wahrnehmen), beim Motor ist es die mechanische Trägheit. Unsere Versuchsschialtung soll solche Impulse abgeben. Die Impulsbreite stellen wir mit dem Vorwärts-Rückwärts-Zähler ein.



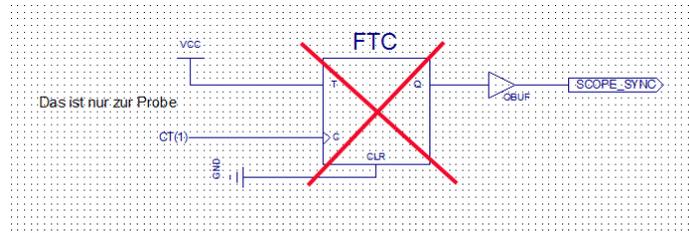
**Abb. 42** Zum Prinzip der Pulsweitenmodulation (PWM).



**Abb. 43** Das Prinzip der Impulserzeugung. Der Periodenzähler läuft ständig um. Läuft er über das Zählende hinaus wieder in die erste Stellung ein, setzt er das Flipflop. Der PWM-Impuls wird aktiv. Entspricht der Zählerstand dem Wert der vorgegebenen Impulsbreite, so setzt der Vergleicher das Flipflop wieder zurück. Der Impuls ist damit zu Ende.



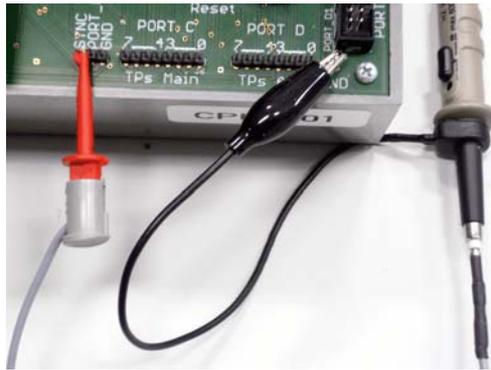
Auf Blatt 1:



**Abb. 44** Den über Tasten einstellbaren Vorwärts-Rückwärts-Zähler haben wir schon. Wir ergänzen die Schaltung (auf einem neuen Blatt) um den Periodenzähler, den Vergleicher und das Flipflop. Das Signal SCOPE SYNC von Blatt 1 wird hierher verlegt und angeschlossen.

**Der Versuchsaufbau:**

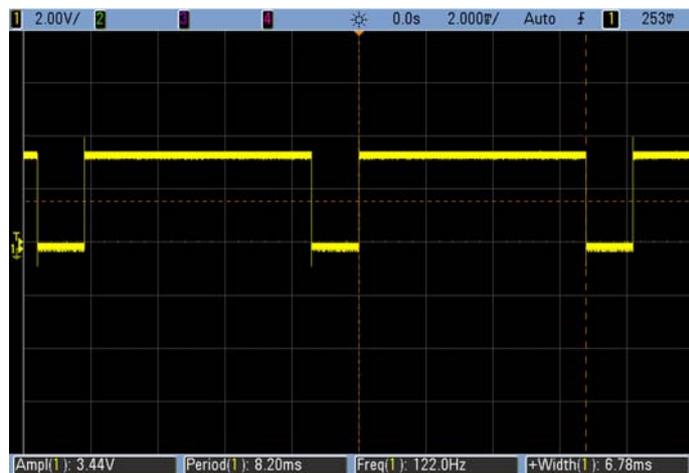
Die Wirkungsweise der Schaltung veranschaulichen wir uns mittels Oszilloskop, die Funktion der Pulsweitenmodulation durch Anschließen des Lüftertrainers 14a. Zunächst nur das Oszilloskop anschließen und die Schaltung ausprobieren.



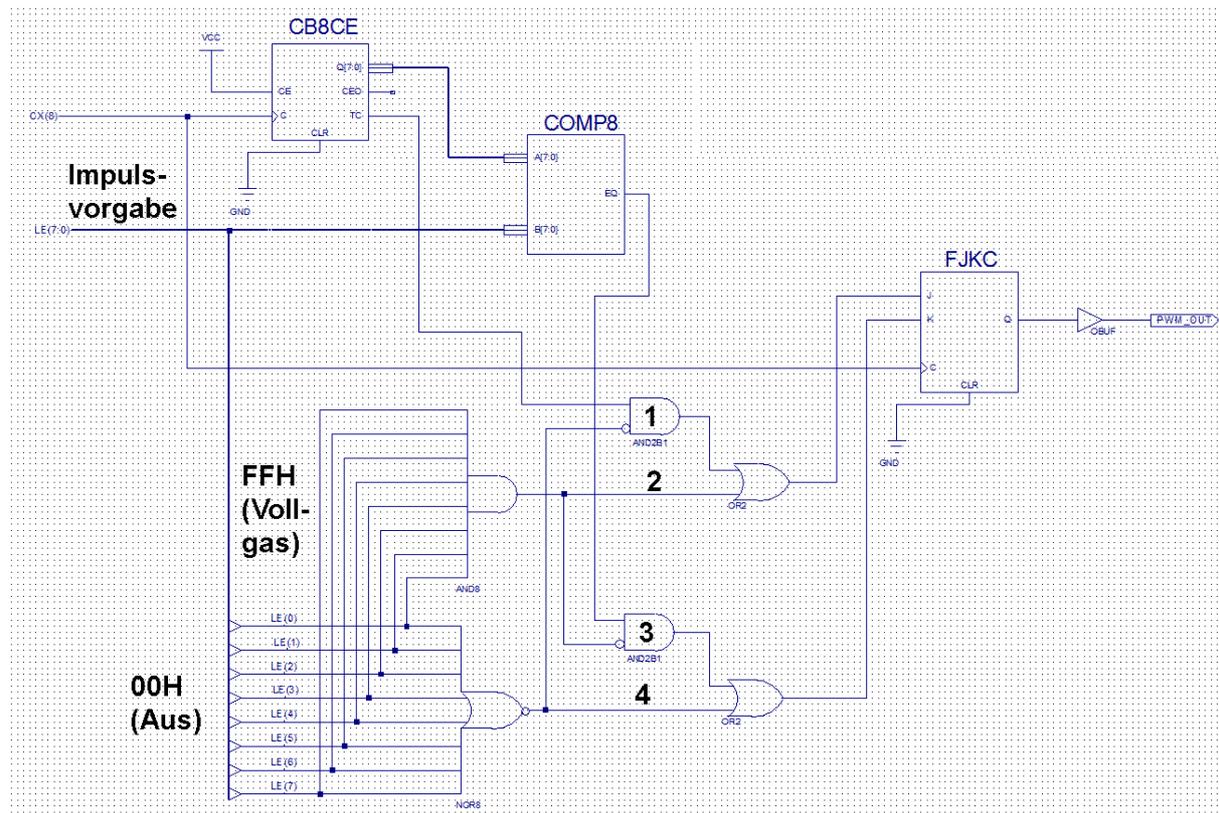
**Abb. 45** So wird das Oszilloskop angeschlossen: der Haken des Tastkopfes an den Meßpunkt SYNC, die Masseklemme an GND.

Die Schaltung scheint zu funktionieren. Sie verhält sich aber irgendwie merkwürdig. Es gibt weder absolute Ruhe noch wirkliches Vollgas. In Stellung 0 des Vorwärts-Rückwärts-Zählers wird immer noch ein Impuls abgegeben. Er ist zwar nur eine einzige Taktperiode breit, manchmal stört das aber trotzdem. In Stellung FFH ergibt sich ein symmetrischer Impulszug. Deshalb wollen wir die Schaltung umbauen:

- Das Flipflop darf nicht gesetzt werden, wenn die Impulsvorgabe eine Null liefert (Aus-Zustand).
- Das Flipflop darf nicht zurückgesetzt werden, wenn die Impulsvorgabe den Endwert (FFH) liefert (Vollgas).



**Abb. 46** Typische PWM-Impulse. Hier sind auch die Zeitparameter zu erkennen.

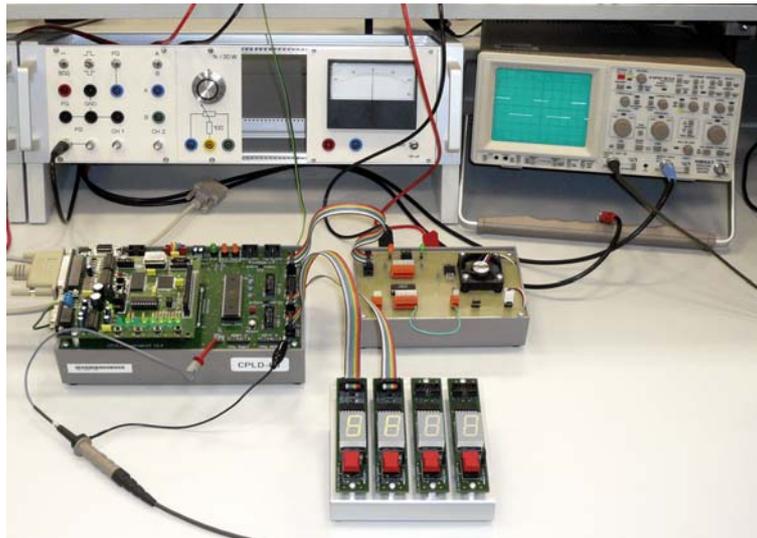


**Abb. 47** Die erweiterte Impulsschaltung. 1 - das Flipflop wird vom Periodenzähler nur gesetzt, wenn die Impulsvorgabe nicht Null ist. 2 - das Flipflop wird immer dann gesetzt, wenn die Impulsvorgabe der Endwert (FFH) ist. 3 - das Flipflop wird vom Vergleichler nur zurückgesetzt, wenn die Impulsvorgabe nicht der Endwert (FFH) ist. 4 - das Flipflop wird immer dann zurückgesetzt, wenn die Impulsvorgabe Null ist.

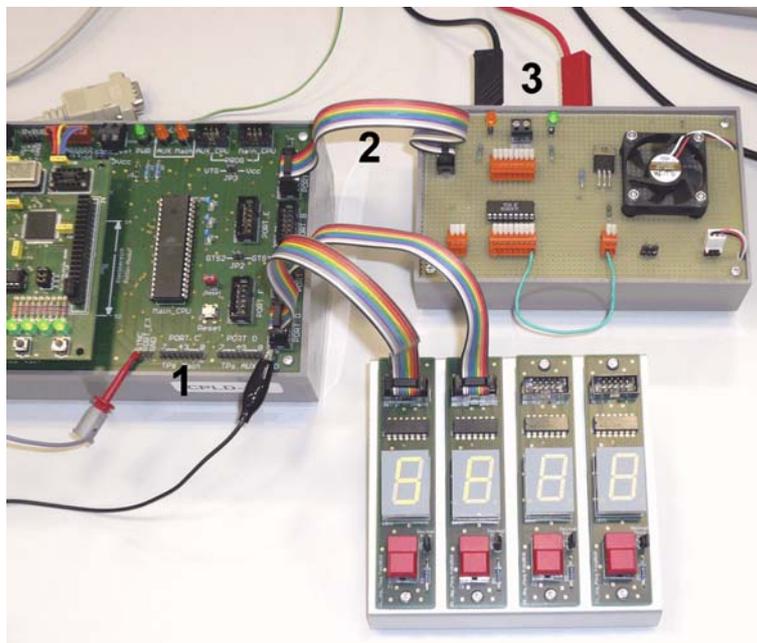
Wenn alles funktioniert, nehmen wir der Lüftertrainer in Betrieb.

#### Aufbau- und Inbetriebnahmevorschrift:

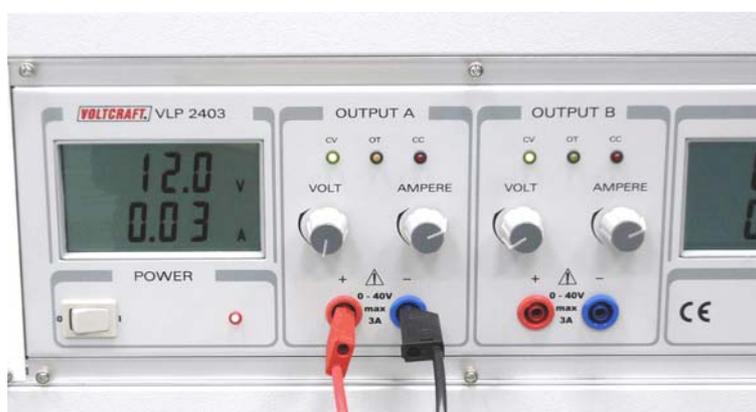
1. Stromversorgung (Festspannungsnetzgerät) aus.
3. Voltcraft-Netzgerät ein. Auf 12 V stellen. Wieder aus.
2. Lüftertrainer (Rückseite) an Voltcraft-Netzgerät anschließen. Schwarze Buchse an – (blau), blaue Buchse an + (rot).
3. Lüftertrainer an Port B des CPLD-Lehrgerätes anschließen.
4. Oszilloskop anschließen.
5. Oszilloskop ein.
6. Festspannungsnetzgerät ein.
7. Voltcraft ein.



**Abb. 48** Der Versuchsaufbau im Überblick.



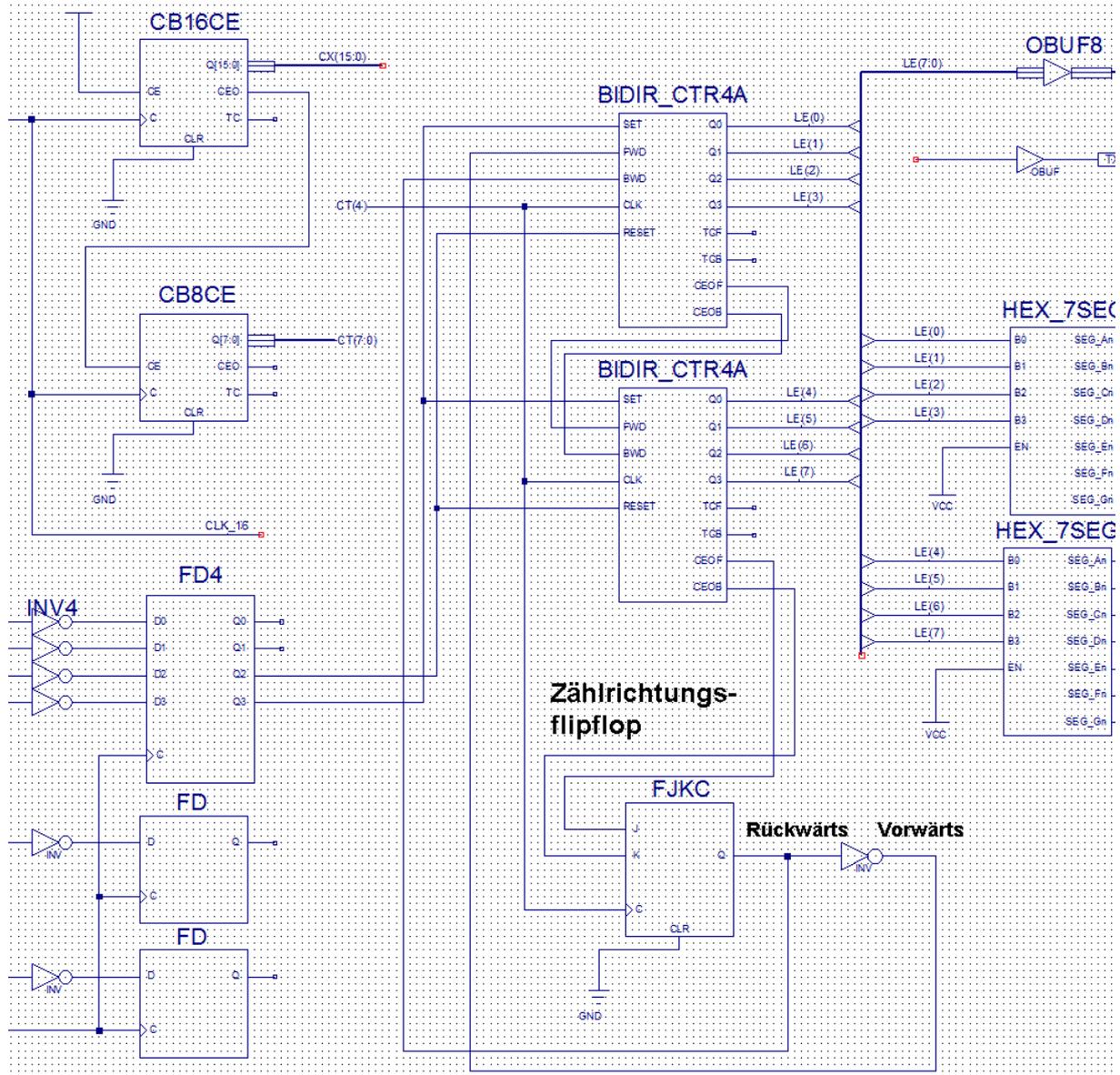
**Abb. 49** Der Versuchsaufbau. 1 - Oszilloskopanschluß. 2 - Lüftertrainer an Port A. 3 - Lüfterbetriebsspannung vom Voltcraft-Netzgerät.



**Abb. 50** So wird das Voltcraft-Netzgerät eingestellt und angeschlossen.

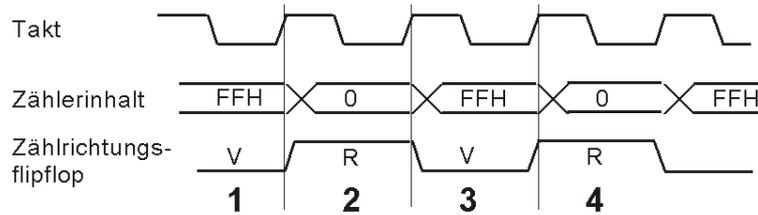
**Den Bereich der Impulsbreite automatisch durchfahren (Wobbeln)**

Die Impulsbreite und damit die Drehzahl des Lüfters soll von Null an langsam bis zum Maximalwert wachsen und dann ebenso langsam bis auf Null zurückgehen. Es liegt nahe, eine entsprechend langsamen Takt an der Vorwärts-Rückwärts-Zähler anzulegen und über ein Flipflop die Zählrichtung zyklisch umzuschalten: erst vorwärts zählen, nach Erreichen der Endstellung FFH auf Rückwärtszählen umschalten, nach Erreichen der Endstellung 00H wieder auf Vorwärtszählen usw.



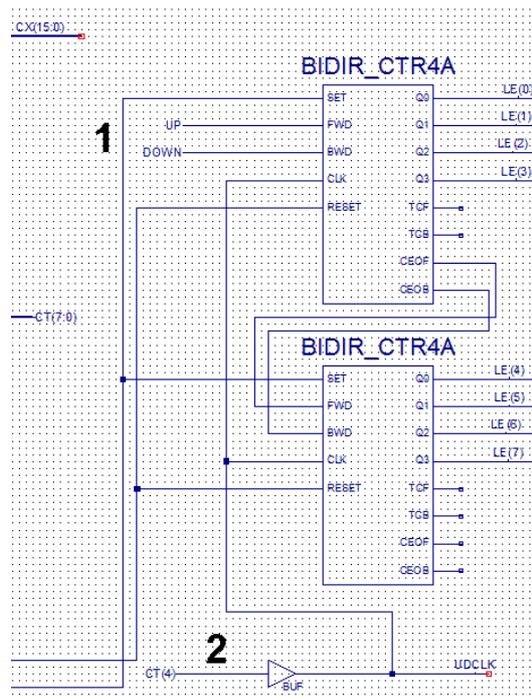
**Abb. 51** Die Schaltung ist schnell geändert...

Sie funktioniert aber offensichtlich nicht richtig. Weshalb? Es ist eine Eigenheit der vollsynchronen Betriebsweise. Jedes Signal kommt erst im nachfolgenden Takt zur Wirkung, jedes Flipflop verzögert um eine Taktperiode.

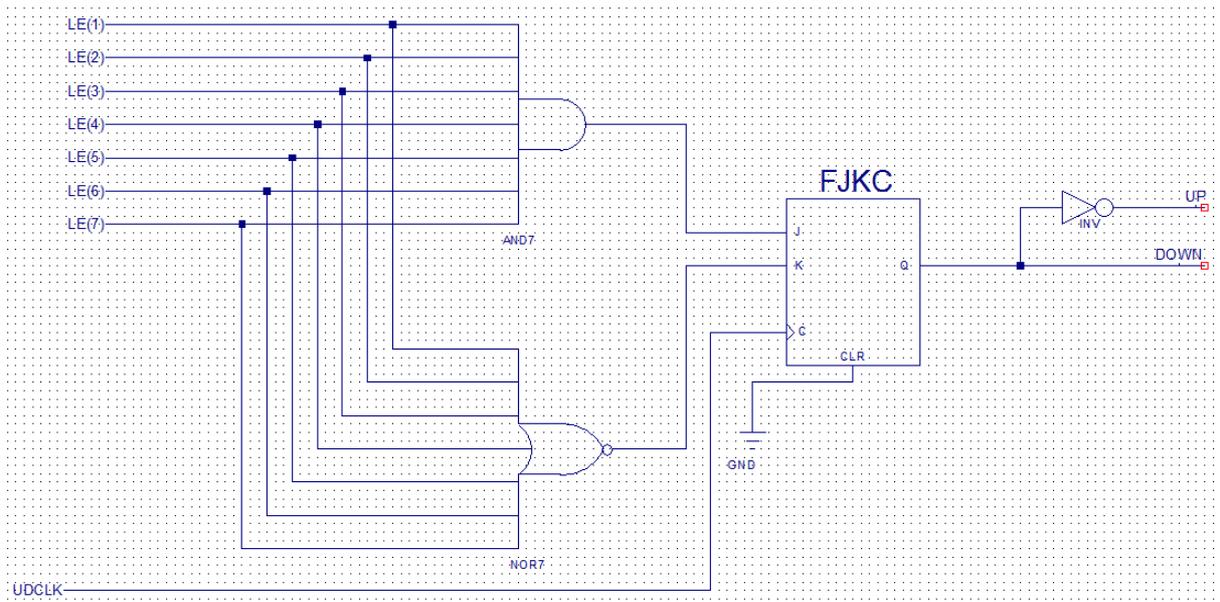


**Abb. 52** Wenn der Zähler das erste Mal hochgezählt hat (bis FFH), passiert folgendes: 1 - Zählerstand = FFH. 2 - mit dem nachfolgenden Takt schaltet das Flipflop um. Der Zähler zählt noch einmal vorwärts und gelangt so in Stellung 0. 3 - In Rückwärtsrichtung ist 0 die Endstellung. Also wird das Flipflop wieder umgeschaltet. Der Zähler zählt noch einmal rückwärts und gelangt so in Stellung FFH. 4 - Jetzt gilt wieder die Vorwärtsrichtung. Da ist FFH die Endstellung. Also wird erneut umgeschaltet usw. Die Schaltung springt so zyklisch zwischen den Stellungen 0 und FFH hin und her.

Die Lösung: wir müssen das Umschalten jeweils eine Zählerstellung vorher auslösen; von vorwärts auf rückwärts in Stellung F7H, von rückwärts auf vorwärts in Stellung 01H. Wenn man die Zählvorgänge genauer betrachtet, sieht man sofort, daß in beiden Fällen die Bitposition 0 im Grunde als Don't Care behandelt werden kann.



**Abb. 53** Schaltungsänderungen. 1 - Richtungssteuersignale; 2 - gemeinsamer Takt für Zähler und Zählrichtungsflop.



**Abb. 54** Die neue Richtungssteuerung.

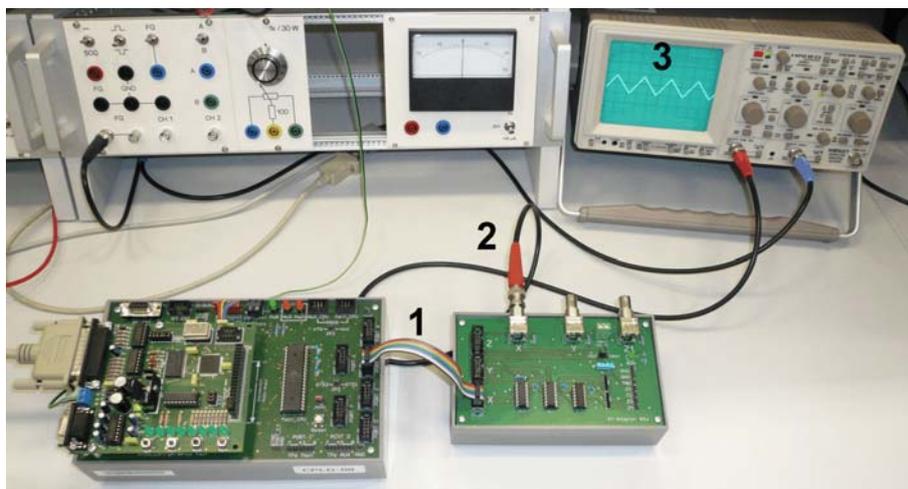
## Aufgabe 6: Dreieckssignale

Abschließen soll gezeigt werden, wie aus Bitmustern analoge Signalverläufe erzeugt werden können.

Der Versuchsaufbau:

1. Alles aus (Netzgeräte).
2. Lüftertrainer abbauen.
3. Den XY-Adapter 09a an Port B des Pollin-Lehrgerätes und an das Oszilloskop anschließen. Dieser Adapter ist mit Analog-Digital-Wandlern bestückt.

Wir nutzen die zuvor aufgebaute Schaltung, betreiben aber den Vorwärt-Rückwärts-Zähler mit einem schnelleren Takt und ändern die Zählerausgänge von den LEDs auf Port B (neue Constraints-Datei). Hierzu können wir ein neues Projekt anlegen und die Quellen aus dem Verzeichnis "XY\_Vorlage\_1" kopieren ("Add Copy of Source").



**Abb. 55** Der Versuchsaufbau. 1 - Port B CLD-Lehrgerät an Steckverbinder X des XY-Adapters. 2 - Oszilloskopanschluß an BNC-Buchse X. 3 - dieser Signalverlauf wird erwartet.

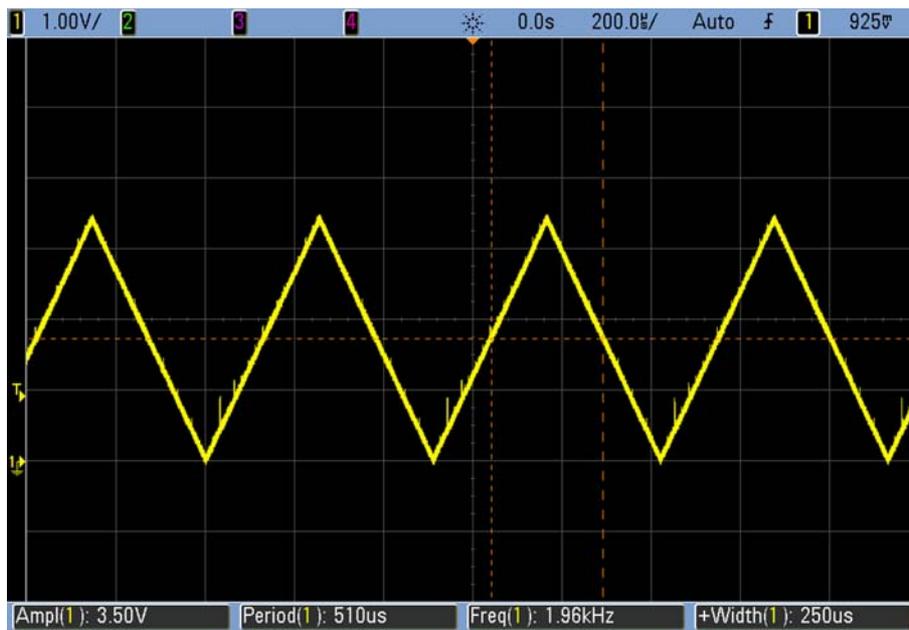


Abb. 56 Hieraus sind die Signalkennwerte ersichtlich.

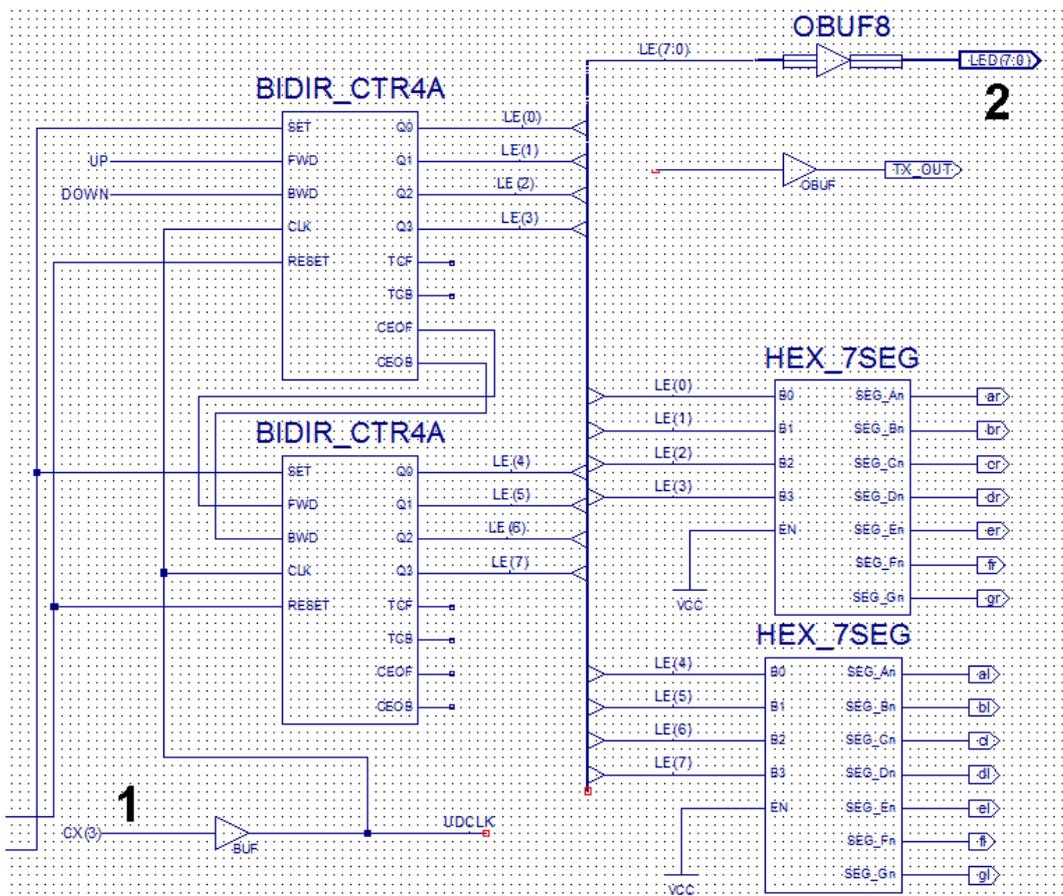


Abb. 57 Änderungen gegenüber der vorhergehenden Schaltungslösung. 1 - anderer Takt (höhere Frequenz); 2 - Ausgabe nicht an die LEDs, sondern an Port B des CPLD-Lehrgeräts (geänderte Constraintsdatei).

## Constraintstabellen

Uhrmuster_1	Uhrmuster_2:
NET "CLOCK" LOC = "P22" ;	NET "CLOCK" LOC = "P22" ;
NET "hha" LOC = "P54" ;	NET "hha" LOC = "P54" ;
NET "hhb" LOC = "P55" ;	NET "hhb" LOC = "P55" ;
NET "hhc" LOC = "P52" ;	NET "hhc" LOC = "P52" ;
NET "hhd" LOC = "P53" ;	NET "hhd" LOC = "P53" ;
NET "hhe" LOC = "P86" ;	NET "hhe" LOC = "P86" ;
NET "hhf" LOC = "P50" ;	NET "hhf" LOC = "P50" ;
NET "hhg" LOC = "P3" ;	NET "hhg" LOC = "P3" ;
NET "hla" LOC = "P64" ;	NET "hla" LOC = "P64" ;
NET "hlb" LOC = "P65" ;	NET "hlb" LOC = "P65" ;
NET "hlc" LOC = "P61" ;	NET "hlc" LOC = "P61" ;
NET "hld" LOC = "P63" ;	NET "hld" LOC = "P63" ;
NET "hle" LOC = "P59" ;	NET "hle" LOC = "P59" ;
NET "hlf" LOC = "P60" ;	NET "hlf" LOC = "P60" ;
NET "hlg" LOC = "P56" ;	NET "hlg" LOC = "P56" ;
NET "mha" LOC = "P73" ;	NET "mha" LOC = "P73" ;
NET "mhb" LOC = "P74" ;	NET "mhb" LOC = "P74" ;
NET "mhc" LOC = "P71" ;	NET "mhc" LOC = "P71" ;
NET "mhd" LOC = "P72" ;	NET "mhd" LOC = "P72" ;
NET "mhe" LOC = "P68" ;	NET "mhe" LOC = "P68" ;
NET "mhf" LOC = "P70" ;	NET "mhf" LOC = "P70" ;
NET "mhg" LOC = "P66" ;	NET "mhg" LOC = "P66" ;
NET "mla" LOC = "P82" ;	NET "mla" LOC = "P82" ;
NET "mlb" LOC = "P85" ;	NET "mlb" LOC = "P85" ;
NET "mlc" LOC = "P80" ;	NET "mlc" LOC = "P80" ;
NET "mld" LOC = "P81" ;	NET "mld" LOC = "P81" ;
NET "mle" LOC = "P78" ;	NET "mle" LOC = "P78" ;
NET "mlf" LOC = "P79" ;	NET "mlf" LOC = "P79" ;
NET "mlg" LOC = "P76" ;	NET "mlg" LOC = "P76" ;
NET "SCOPE_SYNC" LOC = "P27" ;	NET "SCOPE_SYNC" LOC = "P27" ;
NET "TA0" LOC = "P49" ;	NET "TA0" LOC = "P49" ;
NET "TA1" LOC = "P46" ;	NET "TA1" LOC = "P46" ;
NET "TA2" LOC = "P43" ;	NET "TA2" LOC = "P43" ;
NET "TA3" LOC = "P42" ;	NET "TA3" LOC = "P42" ;
NET "TA_LED0" LOC = "P77" ;	NET "TA_LED0" LOC = "P77" ;
NET "TA_LED1" LOC = "P67" ;	NET "TA_LED1" LOC = "P67" ;
NET "TA_LED2" LOC = "P58" ;	NET "TA_LED2" LOC = "P58" ;
NET "TA_LED3" LOC = "P4" ;	NET "TA_LED3" LOC = "P4" ;

Stoppuhrmuster_1	Zaehlermuster_1
NET "CLOCK" LOC = "P22" ;	NET "CLOCK" LOC = "P22" ;
NET "hha" LOC = "P54" ;	NET "hha" LOC = "P54" ;
NET "hhb" LOC = "P55" ;	NET "hhb" LOC = "P55" ;
NET "hhc" LOC = "P52" ;	NET "hhc" LOC = "P52" ;
NET "hhd" LOC = "P53" ;	NET "hhd" LOC = "P53" ;
NET "hhe" LOC = "P86" ;	NET "hhe" LOC = "P86" ;
NET "hhf" LOC = "P50" ;	NET "hhf" LOC = "P50" ;
NET "hhg" LOC = "P3" ;	NET "hhg" LOC = "P3" ;
NET "hla" LOC = "P64" ;	NET "hla" LOC = "P64" ;
NET "hlb" LOC = "P65" ;	NET "hlb" LOC = "P65" ;
NET "hlc" LOC = "P61" ;	NET "hlc" LOC = "P61" ;
NET "hld" LOC = "P63" ;	NET "hld" LOC = "P63" ;
NET "hle" LOC = "P59" ;	NET "hle" LOC = "P59" ;
NET "hlf" LOC = "P60" ;	NET "hlf" LOC = "P60" ;
NET "hlg" LOC = "P56" ;	NET "hlg" LOC = "P56" ;
NET "mha" LOC = "P73" ;	NET "MEAS" LOC = "P42" ;
NET "mhb" LOC = "P74" ;	NET "mha" LOC = "P73" ;
NET "mhc" LOC = "P71" ;	NET "mhb" LOC = "P74" ;
NET "mhd" LOC = "P72" ;	NET "mhc" LOC = "P71" ;
NET "mhe" LOC = "P68" ;	NET "mhd" LOC = "P72" ;
NET "mhf" LOC = "P70" ;	NET "mhe" LOC = "P68" ;
NET "mhg" LOC = "P66" ;	NET "mhf" LOC = "P70" ;
NET "mla" LOC = "P82" ;	NET "mhg" LOC = "P66" ;
NET "mlb" LOC = "P85" ;	NET "mla" LOC = "P82" ;
NET "mlc" LOC = "P80" ;	NET "mlb" LOC = "P85" ;
NET "mld" LOC = "P81" ;	NET "mlc" LOC = "P80" ;
NET "mle" LOC = "P78" ;	NET "mld" LOC = "P81" ;
NET "mlf" LOC = "P79" ;	NET "mle" LOC = "P78" ;
NET "mlg" LOC = "P76" ;	NET "mlf" LOC = "P79" ;
NET "SCOPE_SYNC" LOC = "P27" ;	NET "mlg" LOC = "P76" ;
NET "TA0" LOC = "P49" ;	NET "SCOPE_SYNC" LOC = "P27" ;
NET "TA1" LOC = "P46" ;	NET "TA0" LOC = "P49" ;
NET "TA2" LOC = "P43" ;	NET "TA1" LOC = "P46" ;
NET "TA3" LOC = "P42" ;	NET "TA2" LOC = "P43" ;
NET "TA_LED0" LOC = "P77" ;	NET "TA_LED0" LOC = "P77" ;
NET "TA_LED1" LOC = "P67" ;	NET "TA_LED1" LOC = "P67" ;
NET "TA_LED2" LOC = "P58" ;	NET "TA_LED2" LOC = "P58" ;
NET "TA_LED3" LOC = "P4" ;	NET "TA_LED3" LOC = "P4" ;

Zaehlermuster_2	Zaehlermuster_3 und _4
NET "CLOCK" LOC = "P22" ;	NET "CLOCK" LOC = "P22" ;
NET "hha" LOC = "P54" ;	NET "hha" LOC = "P54" ;
NET "hhb" LOC = "P55" ;	NET "hhb" LOC = "P55" ;
NET "hhc" LOC = "P52" ;	NET "hhc" LOC = "P52" ;
NET "hhd" LOC = "P53" ;	NET "hhd" LOC = "P53" ;
NET "hhe" LOC = "P86" ;	NET "hhe" LOC = "P86" ;
NET "hhf" LOC = "P50" ;	NET "hhf" LOC = "P50" ;
NET "hhg" LOC = "P3" ;	NET "hhg" LOC = "P3" ;
NET "hla" LOC = "P64" ;	NET "hla" LOC = "P64" ;
NET "hlb" LOC = "P65" ;	NET "hlb" LOC = "P65" ;
NET "hlc" LOC = "P61" ;	NET "hlc" LOC = "P61" ;
NET "hld" LOC = "P63" ;	NET "hld" LOC = "P63" ;
NET "hle" LOC = "P59" ;	NET "hle" LOC = "P59" ;
NET "hlf" LOC = "P60" ;	NET "hlf" LOC = "P60" ;
NET "hlg" LOC = "P56" ;	NET "hlg" LOC = "P56" ;
NET "MEAS" LOC = "P42" ;	NET "LED0" LOC = "P41" ;
NET "mha" LOC = "P73" ;	NET "LED2" LOC = "P39" ;
NET "mhb" LOC = "P74" ;	NET "MEAS" LOC = "P42" ;
NET "mhc" LOC = "P71" ;	NET "mha" LOC = "P73" ;
NET "mhd" LOC = "P72" ;	NET "mhb" LOC = "P74" ;
NET "mhe" LOC = "P68" ;	NET "mhc" LOC = "P71" ;
NET "mhf" LOC = "P70" ;	NET "mhd" LOC = "P72" ;
NET "mhg" LOC = "P66" ;	NET "mhe" LOC = "P68" ;
NET "mla" LOC = "P82" ;	NET "mhf" LOC = "P70" ;
NET "mlb" LOC = "P85" ;	NET "mhg" LOC = "P66" ;
NET "mlc" LOC = "P80" ;	NET "mla" LOC = "P82" ;
NET "mld" LOC = "P81" ;	NET "mlb" LOC = "P85" ;
NET "mle" LOC = "P78" ;	NET "mlc" LOC = "P80" ;
NET "mlf" LOC = "P79" ;	NET "mld" LOC = "P81" ;
NET "mlg" LOC = "P76" ;	NET "mle" LOC = "P78" ;
NET "SCOPE_SYNC" LOC = "P27" ;	NET "mlf" LOC = "P79" ;
NET "TA0" LOC = "P49" ;	NET "mlg" LOC = "P76" ;
NET "TA1" LOC = "P46" ;	NET "SCOPE_SYNC" LOC = "P27" ;
NET "TA2" LOC = "P43" ;	NET "TA0" LOC = "P49" ;
NET "TA_LED0" LOC = "P77" ;	NET "TA1" LOC = "P46" ;
NET "TA_LED1" LOC = "P67" ;	NET "TA2" LOC = "P43" ;
NET "TA_LED2" LOC = "P58" ;	NET "TA_LED0" LOC = "P77" ;
NET "TA_LED3" LOC = "P4" ;	NET "TA_LED1" LOC = "P67" ;
	NET "TA_LED2" LOC = "P58" ;
	NET "TA_LED3" LOC = "P4" ;

FWD_BWD_Muster_1:	PWM_Muster_1 und _2
<pre> NET "al" LOC = "P54" ; NET "ar" LOC = "P64" ; NET "bl" LOC = "P55" ; NET "br" LOC = "P65" ; NET "cl" LOC = "P52" ; NET "CLOCK" LOC = "P22" ; NET "cr" LOC = "P61" ; NET "dl" LOC = "P53" ; NET "dr" LOC = "P63" ; NET "el" LOC = "P86" ; NET "er" LOC = "P59" ; NET "fl" LOC = "P50" ; NET "fr" LOC = "P60" ; NET "gl" LOC = "P3" ; NET "gr" LOC = "P56" ; NET "LED&lt;0&gt;" LOC = "P41" ; NET "LED&lt;1&gt;" LOC = "P40" ; NET "LED&lt;2&gt;" LOC = "P39" ; NET "LED&lt;3&gt;" LOC = "P37" ; NET "LED&lt;4&gt;" LOC = "P36" ; NET "LED&lt;5&gt;" LOC = "P35" ; NET "LED&lt;6&gt;" LOC = "P97" ; NET "LED&lt;7&gt;" LOC = "P96" ; NET "SCOPE_SYNC" LOC = "P27" ; NET "TA0" LOC = "P49" ; NET "TA1" LOC = "P46" ; NET "TA2" LOC = "P43" ; NET "TA3" LOC = "P42" ; NET "TA_L" LOC = "P4" ; NET "TA_R" LOC = "P58" ; NET "TX_OUT" LOC = "P92" ; </pre>	<pre> NET "al" LOC = "P54" ; NET "ar" LOC = "P64" ; NET "bl" LOC = "P55" ; NET "br" LOC = "P65" ; NET "cl" LOC = "P52" ; NET "CLOCK" LOC = "P22" ; NET "cr" LOC = "P61" ; NET "dl" LOC = "P53" ; NET "dr" LOC = "P63" ; NET "el" LOC = "P86" ; NET "er" LOC = "P59" ; NET "fl" LOC = "P50" ; NET "fr" LOC = "P60" ; NET "gl" LOC = "P3" ; NET "gr" LOC = "P56" ; NET "LED&lt;0&gt;" LOC = "P41" ; NET "LED&lt;1&gt;" LOC = "P40" ; NET "LED&lt;2&gt;" LOC = "P39" ; NET "LED&lt;3&gt;" LOC = "P37" ; NET "LED&lt;4&gt;" LOC = "P36" ; NET "LED&lt;5&gt;" LOC = "P35" ; NET "LED&lt;6&gt;" LOC = "P97" ; NET "LED&lt;7&gt;" LOC = "P96" ; NET "PWM_OUT" LOC = "P82" ; NET "SCOPE_SYNC" LOC = "P27" ; NET "TA0" LOC = "P49" ; NET "TA1" LOC = "P46" ; NET "TA2" LOC = "P43" ; NET "TA3" LOC = "P42" ; NET "TA_L" LOC = "P4" ; NET "TA_R" LOC = "P58" ; NET "TX_OUT" LOC = "P92" ; </pre>

Wobbelmuster_1 und _2	XY_Muster_1
NET "al" LOC = "P54" ;	NET "al" LOC = "P54" ;
NET "ar" LOC = "P64" ;	NET "ar" LOC = "P64" ;
NET "bl" LOC = "P55" ;	NET "bl" LOC = "P55" ;
NET "br" LOC = "P65" ;	NET "br" LOC = "P65" ;
NET "cl" LOC = "P52" ;	NET "cl" LOC = "P52" ;
NET "CLOCK" LOC = "P22" ;	NET "CLOCK" LOC = "P22" ;
NET "cr" LOC = "P61" ;	NET "cr" LOC = "P61" ;
NET "dl" LOC = "P53" ;	NET "dl" LOC = "P53" ;
NET "dr" LOC = "P63" ;	NET "dr" LOC = "P63" ;
NET "el" LOC = "P86" ;	NET "el" LOC = "P86" ;
NET "er" LOC = "P59" ;	NET "er" LOC = "P59" ;
NET "fl" LOC = "P50" ;	NET "fl" LOC = "P50" ;
NET "fr" LOC = "P60" ;	NET "fr" LOC = "P60" ;
NET "gl" LOC = "P3" ;	NET "gl" LOC = "P3" ;
NET "gr" LOC = "P56" ;	NET "gr" LOC = "P56" ;
NET "LED<0>" LOC = "P41" ;	NET "LED<0>" LOC = "P41" ;
NET "LED<1>" LOC = "P40" ;	NET "LED<1>" LOC = "P40" ;
NET "LED<2>" LOC = "P39" ;	NET "LED<2>" LOC = "P39" ;
NET "LED<3>" LOC = "P37" ;	NET "LED<3>" LOC = "P37" ;
NET "LED<4>" LOC = "P36" ;	NET "LED<4>" LOC = "P36" ;
NET "LED<5>" LOC = "P35" ;	NET "LED<5>" LOC = "P35" ;
NET "LED<6>" LOC = "P97" ;	NET "LED<6>" LOC = "P97" ;
NET "LED<7>" LOC = "P96" ;	NET "LED<7>" LOC = "P96" ;
NET "PWM_OUT" LOC = "P82" ;	NET "PWM_OUT" LOC = "P82" ;
NET "SCOPE_SYNC" LOC = "P27" ;	NET "SCOPE_SYNC" LOC = "P27" ;
NET "TA0" LOC = "P49" ;	NET "TA0" LOC = "P49" ;
NET "TA1" LOC = "P46" ;	NET "TA1" LOC = "P46" ;
NET "TA2" LOC = "P43" ;	NET "TA2" LOC = "P43" ;
NET "TA3" LOC = "P42" ;	NET "TA3" LOC = "P42" ;
NET "TA_L" LOC = "P4" ;	NET "TA_L" LOC = "P4" ;
NET "TA_R" LOC = "P58" ;	NET "TA_R" LOC = "P58" ;
NET "TX_OUT" LOC = "P92" ;	NET "TX_OUT" LOC = "P92" ;

**Mustervorlagen**

<b>Aufgabe</b>	<b>Titel</b>	<b>Musterdateien</b>
1	Tageszeituhr	Uhrvorlage_1
	Tageszeituhr mit wählbarem Zeittakt	Uhrvorlage_2
2	Stoppuhr	Stoppuhrvorlage_1
3	Universalzähler, Frequenzmessung	Zaehlervorlage_1
	Universalzähler, Periodendauermessung	Zaehlervorlage_2
	Universalzähler, umschaltbar	Zaehlervorlage_3
	Universalzähler mit Unterdrückung führender Nullen	Zaehlervorlage_4
4	Vorwärts und rückwärts zählen	FWD_BWD_Vorlage_1
5	Pulsweitenmodulation, Grundsaltung	PWM_Vorlage_1
	Pulsweitenmodulation, verbessert	PWM_Vorlage_2
	Pulsweitenmodulation mit periodischem Durchlauf (Wobbeln), naiver Versuch (fehlerhaftig)	Wobbelvorlage_1
	Pulsweitenmodulation mit periodischem Durchlauf (Wobbeln), funktionsfähig	Wobbelvorlage_2
6	Dreiecksignale	XY_Vorlage_1