Praktikum Digitaltechnik SS 2014

Versuch 1

Stand: 22. 5. 2014

Versuchsplattform:
CPLD-Lehrgerät 12 mit CPLD Evaluation Board (Pollin). Eingebaute Ein- und Ausgabemittel: vier Taster, acht LEDs, serielle Schnittstelle. Entwicklungssoftware: Xilinx ISE 10.1.

Im ersten Versuch nutzen wir nur die Bedien- und Anzeigemittel (Taster, LEDs) des CPLD Evaluation Boards.

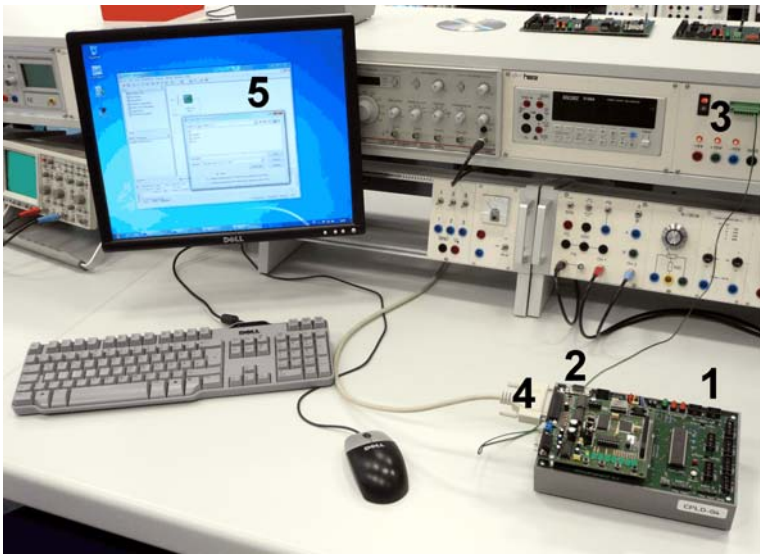


Abb. 1 Der Praktikumsplatz. 1 - CPLD-Lehrgerät 12; 2 - CPLD Evaluation Board (Pollin); 3 - Stromversorgung über Festspannungsnetzgerät 08; 4 - Programmierkabel (Parallelportanschluß); 5 - Entwicklungsumgebung.

Aufgaben:

- Den Entwicklungsgang kennenlernen.
- Passende Taktfrequenzen herstellen.
- Lauflicht (zunächst eine Richtung, dann zwei Richtungen).
- Zwei Binärzähler mit Handbetätigung.

Aufgabe 1: Den Entwicklungsgang kennenlernen

Das erste Projekt betrifft ein RS-Flipflop und ein Toggle-Flipflop, die mit Tasten und Leuchtdioden zu verbinden sind.

1. Software starten.
Die Anwendung heißt Xilinx ISE 10.1.

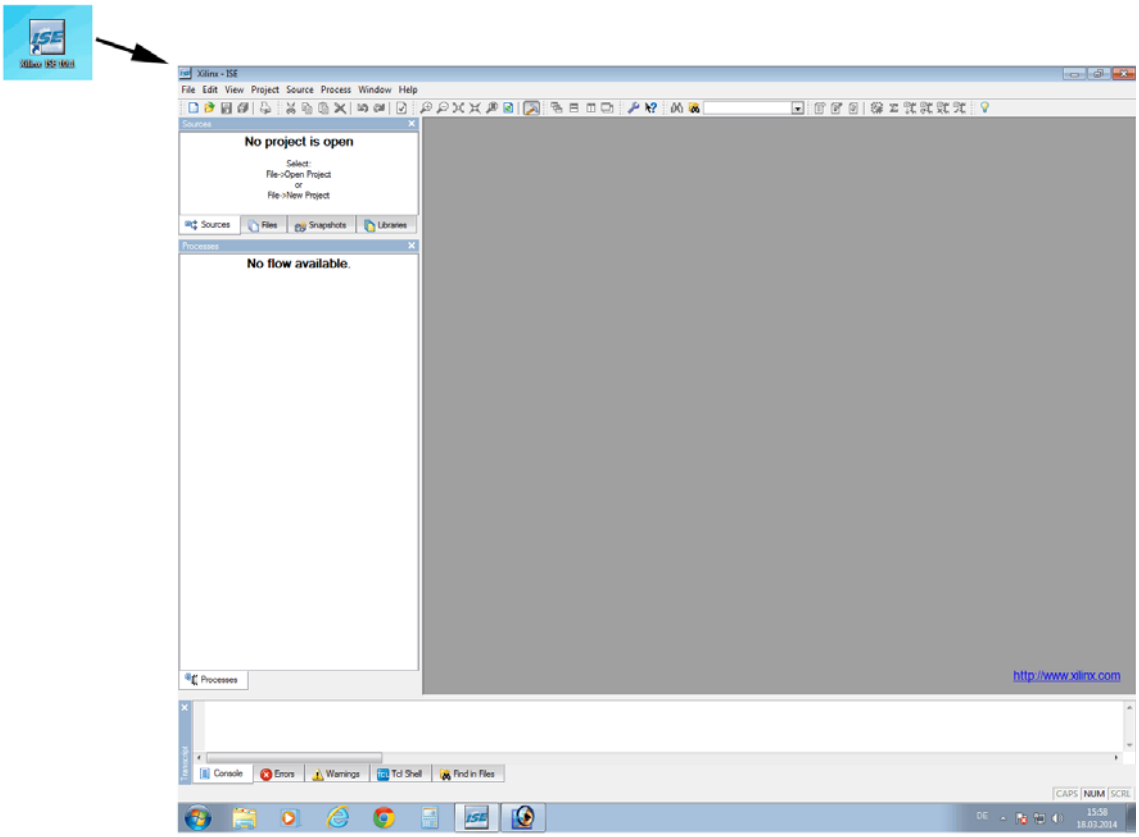


Abb. 2 Xilinx ISE 10.1 wurde gestartet.

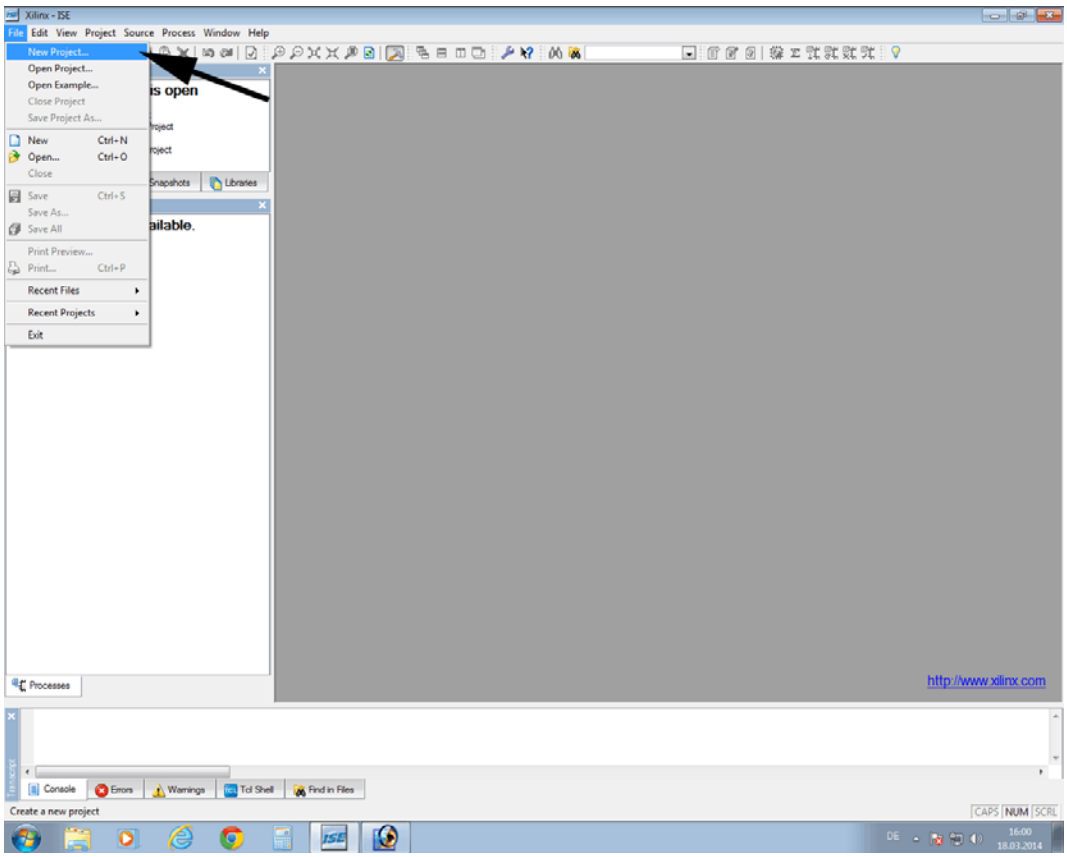


Abb. 3 Das Projekt wird angelegt (New Project).

2. Projekt benennen.

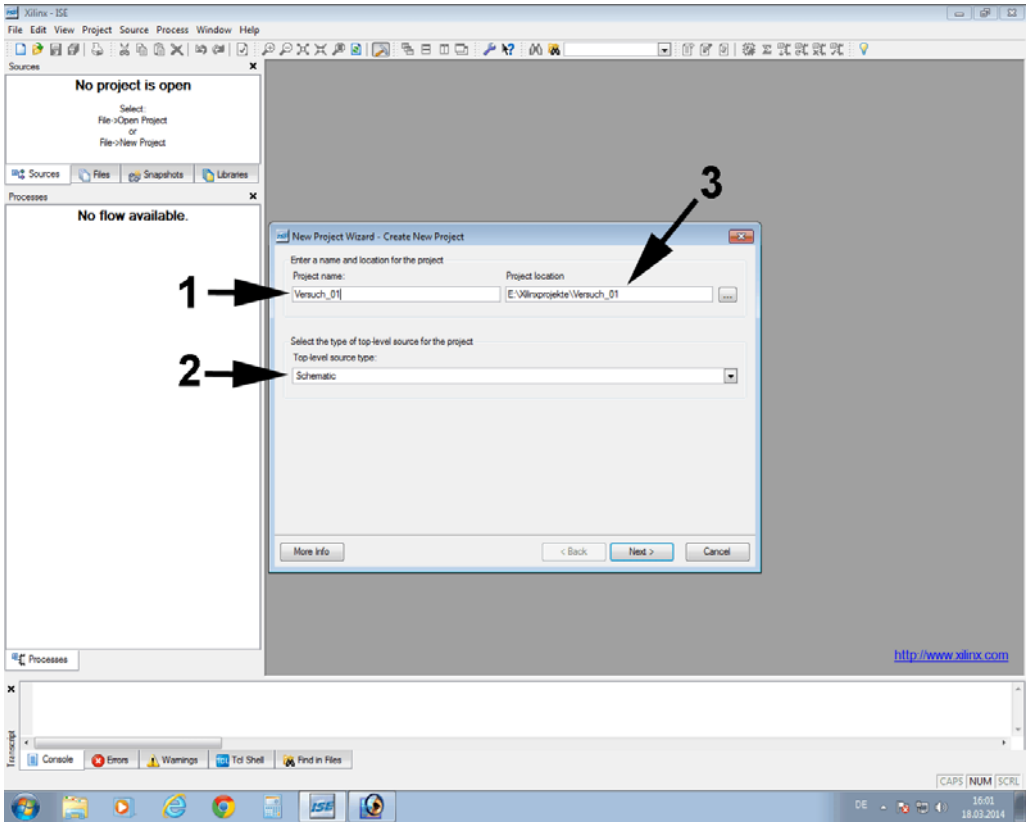


Abb. 4 Das neue Projekt erhält einen Namen. 1 - Namenseingabe. 2 - das Projekt soll als Schaltplan erfaßt werden; 3 - bitte ein vernünftiges Verzeichnis auswählen. Das Projekt nicht auf dem Desktop oder auf der Netzfestplatte ablegen....

ACHTUNG beim Vergeben von Dateinamen: Nur das lateinische Alphabet, Ziffern und die Unterstreichung (_) verwenden; keine Umlaute, kein Eszett (ß), keine Leerzeichen, keine Sonderzeichen.

3. Schaltkreis auswählen.

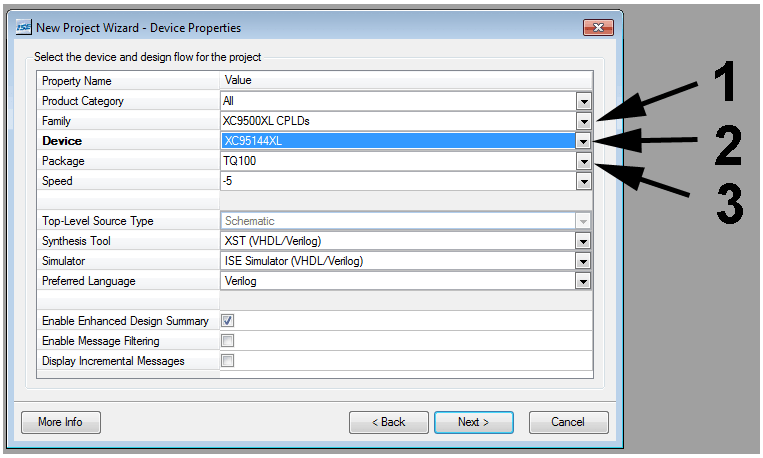


Abb. 5 Schaltkreisauswahl. 1 - es ist die Baureihe 9500XL. 2 - es ist der Typ 95144. 3 - der Schaltkreis befindet sich in einem Flatpack-Gehäuse mit 100 Anschlüssen.

4. Die erste Quelldatei einrichten.

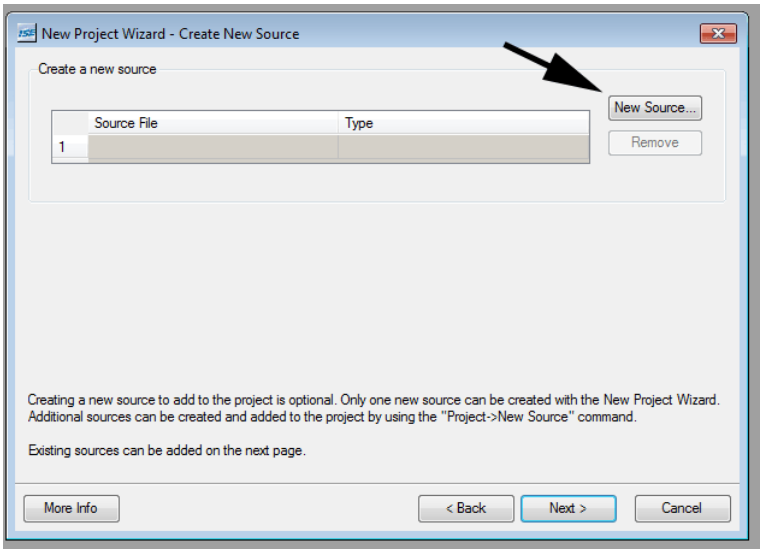


Abb. 6 Einrichten der ersten Quelldatei (“New Source”).

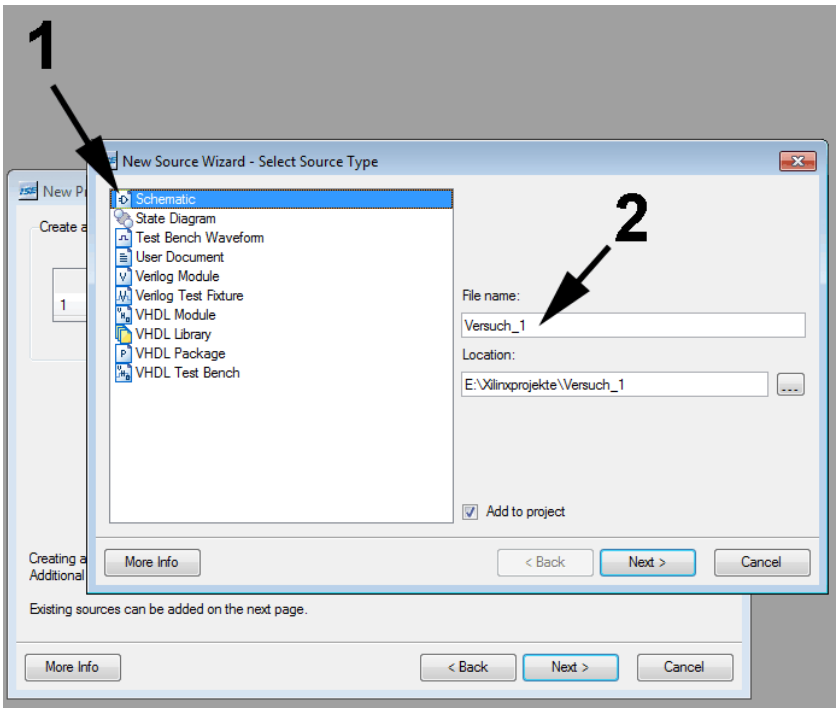


Abb. 7 Es soll ein Schaltplan sein. 1 - den Typ der Quelldatei wählen. 2 - Namen eingeben (beliebig; Dateiendung wird automatisch nachgesetzt).

Diese Dialoge sind zu durchlaufen, um die Quelldatei tatsächlich zu erzeugen:

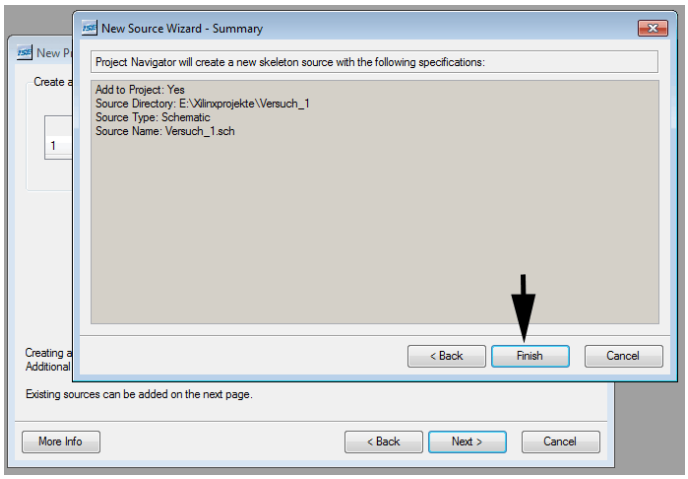


Abb. 8 Weiter mit “Finish”.

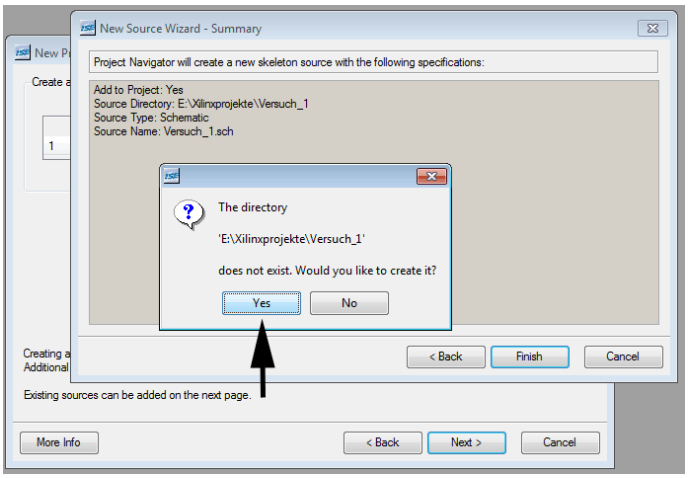


Abb. 9 Das Entwicklungssystem will nun das Projektverzeichnis anlegen. Erlauben (“Yes”).

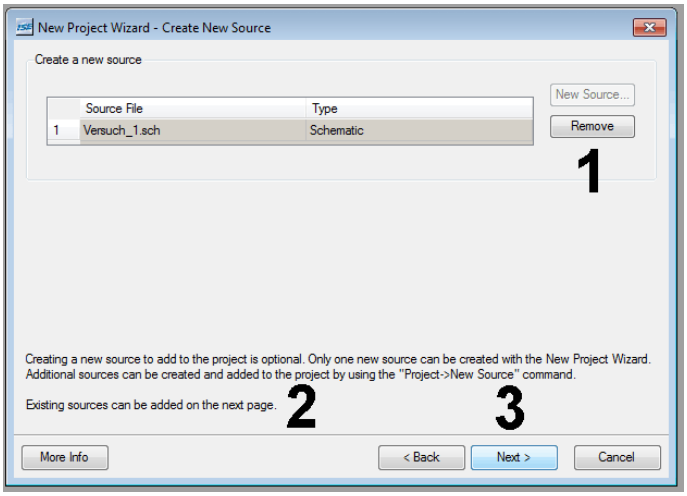


Abb. 10 Jetzt könnte man alles wieder rückgängig machen. 1 - Gelegenheit zum Entfernen der zuvor erzeugten Quelldatei. 2 - im nächsten Dialog können weitere Quellen hinzugefügt werden. 3 - weiter (“Next”).

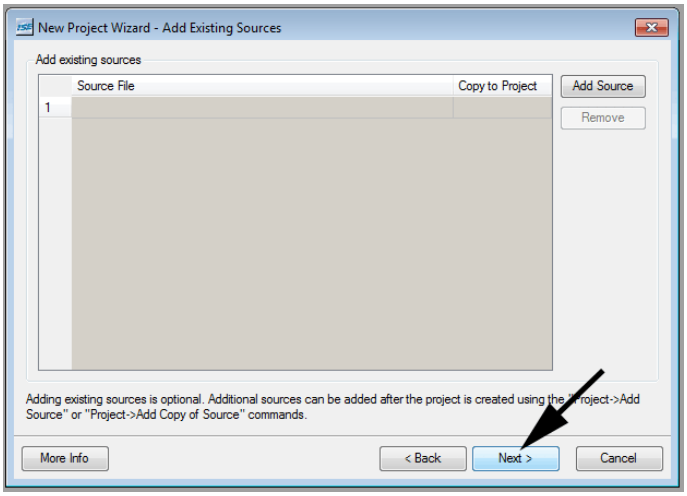


Abb. 11 Hier wird die Gelegenheit geboten, weitere Quelldateien hinzuzufügen. Brauchen wir nicht. Also weiter...

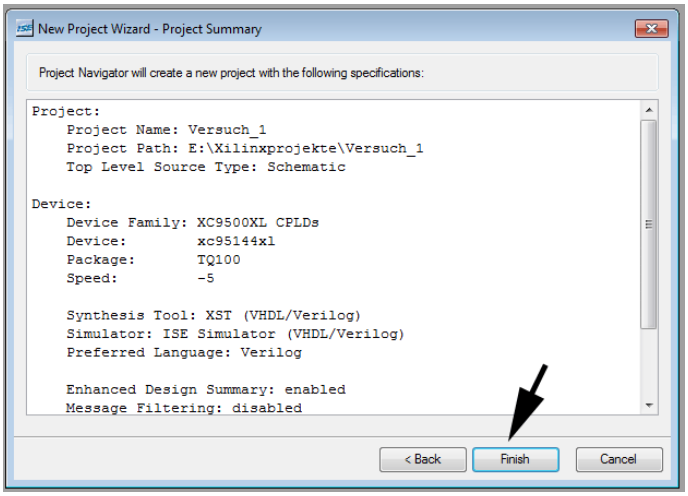


Abb. 12 Das ist der Überblick über das neue Projekt. Ggf. kontrollieren, ob alles stimmt. Wenn nicht, dann zurück. Sonst abschließen (“Finish”).

5. Schaltplan eingeben.

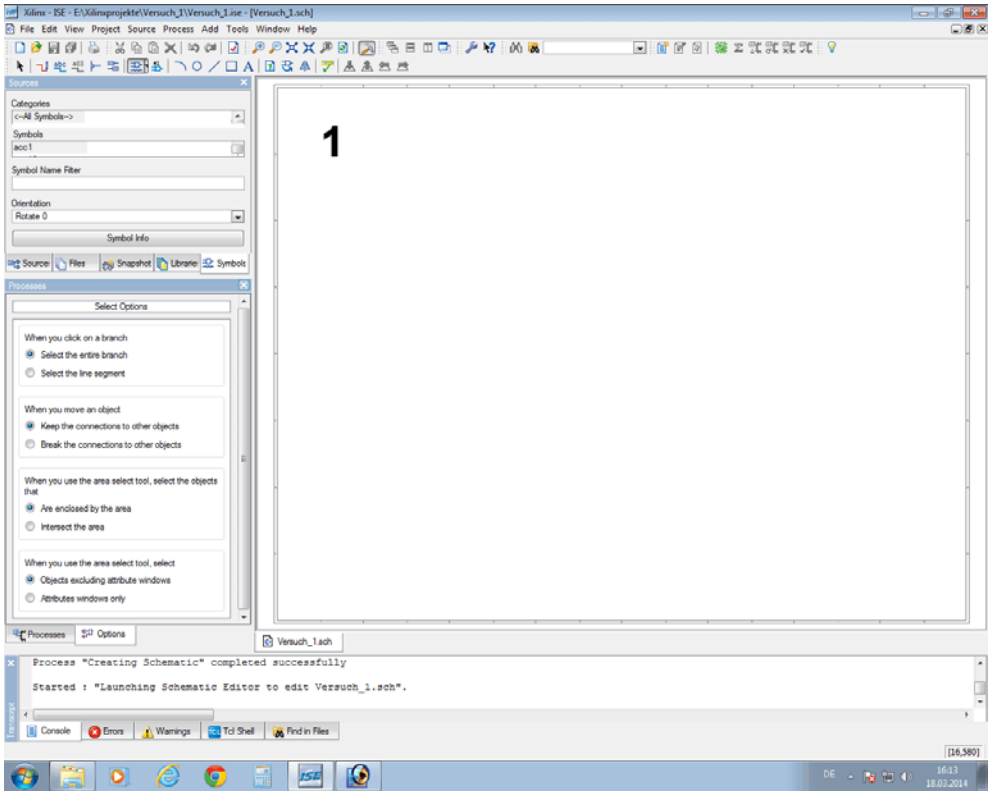


Abb. 13 Der Schaltplaneditor. 1 - die Zeichenfläche.

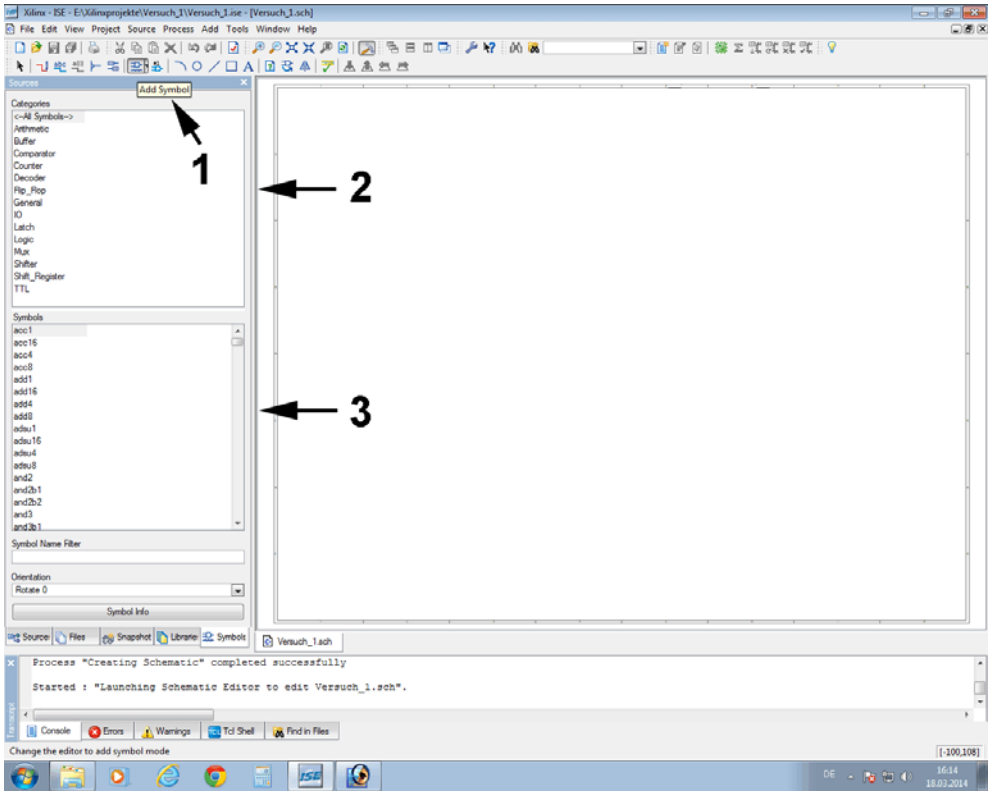


Abb. 14 Funktionselemente auswählen. 1 - Auswahlfunktion. Am linken Rand erscheint ein Katalog der verfügbaren Funktionselemente. 2 - Auswahl der Kategorie. 3 - hier wird das jeweilige Funktionselement aufgerufen.

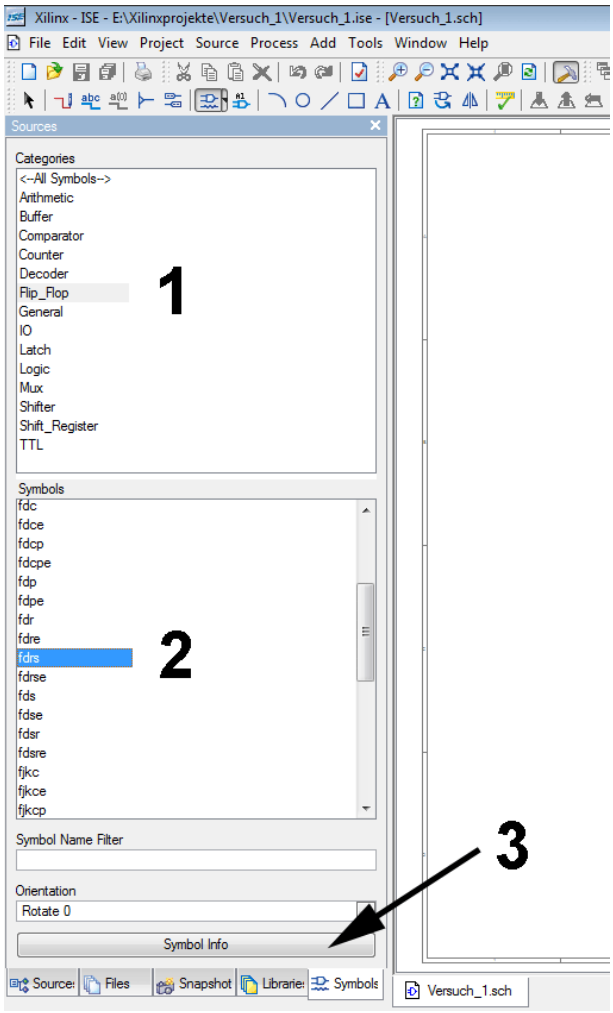
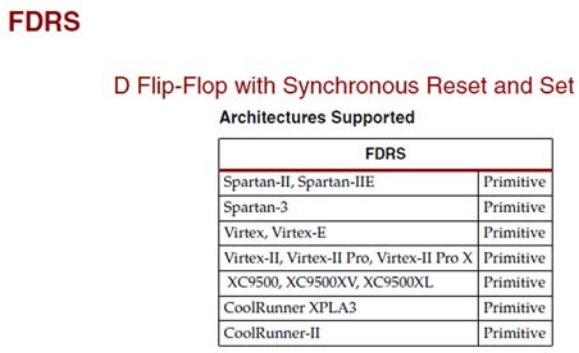


Abb. 15 Das erste Funktionselement soll ein RS-Flipflop sein. 1 - nachsehen, ob unter den Flipflops etwas Passendes zu finden ist. 2 - wir nehmen ein D-Flipflop mit zusätzlichen Setz- und Rücksetzeingängen. Vielleicht tut es der Typ *fdrs*? 3 - hierüber kann man genauer nachsehen.



FDRS

D Flip-Flop with Synchronous Reset and Set

Architectures Supported

FDRS	
Spartan-II, Spartan-IIE	Primitive
Spartan-3	Primitive
Virtex, Virtex-E	Primitive
Virtex-II, Virtex-II Pro, Virtex-II Pro X	Primitive
XC9500, XC9500XV, XC9500XL	Primitive
CoolRunner XPLA3	Primitive
CoolRunner-II	Primitive

Abb. 16 Die Symbolinformation ist ein Datenblatt. Hier ein Auszug daraus. Offensichtlich ist dieses Flipflop ungeeignet, da die Setz- und Rücksetzeingänge synchron wirken.

Bei Xilinx sind Set (S) und Reset (R) synchron wirkende Eingänge. Die entsprechenden asynchron wirkenden Eingänge heißen Preset (P; PRE) und Clear (C; CLR). Also nehmen wir besser ein Flipflop *fdcp*.

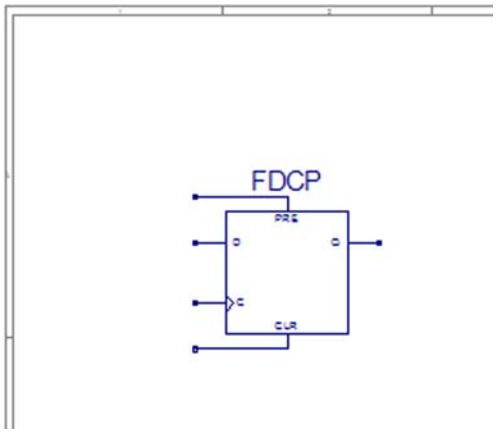


Abb. 17 Das gewünschte Flipflop auf der Zeichenfläche.

Die Taster der Versuchsplattform wirken invertiert, die Flipflopeingänge jedoch nicht. Wir müssen also Negatoren vorschalten.

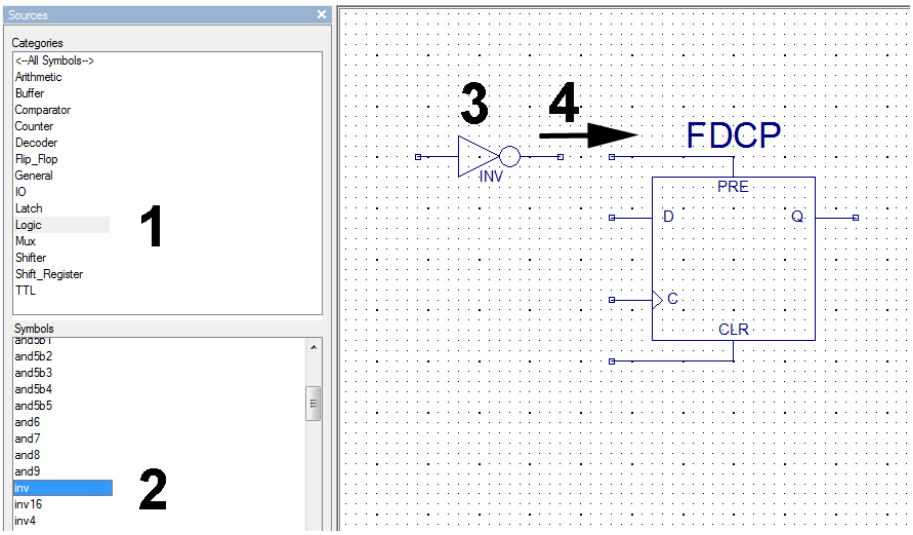


Abb. 18 So wird ein Negator eingebaut. 1 - er ist in der Kategorie Logic zu finden. 2 - er heißt hier ”inv” (Inverter). 3 - das ausgewählte Funktionselement. 4 - die Anschlüsse können direkt miteinander verbunden werden (Andocken).

Nun ist die Schaltung mit den Schaltkreisanschlüssen zu verbinden. Jeder Anschluß erfordert zwei Funktionselemente, einen *Buffer* und einen *Marker*. Ein Eingang braucht einen Input Buffer und einen Input Marker, ein Ausgang einen Output Buffer und einen Output Marker.

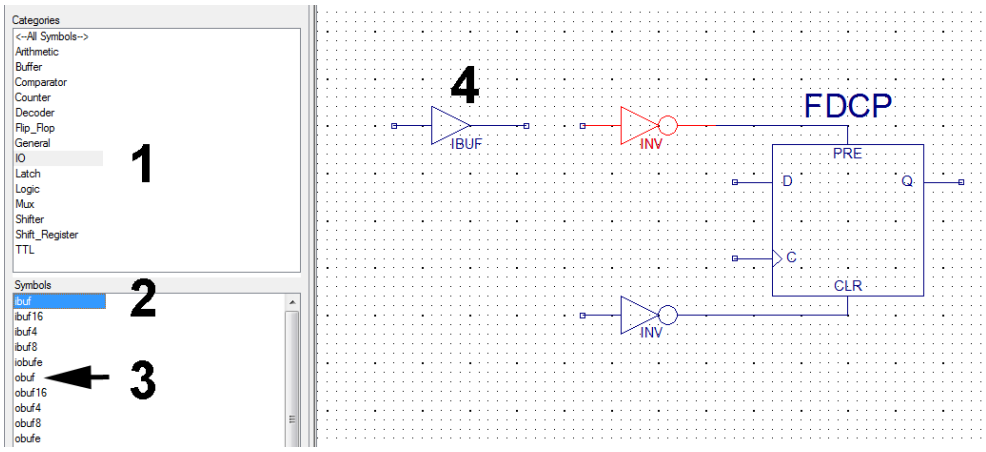


Abb. 19 Puffer einbauen. 1 - diese Funktionselemente finden wir in der Kategorie IO. 2 - der Input Buffer heißt ”ibuf”. 3 - der Output Buffer heißt “obuf”. 4 - ein ausgewählter Input Buffer.

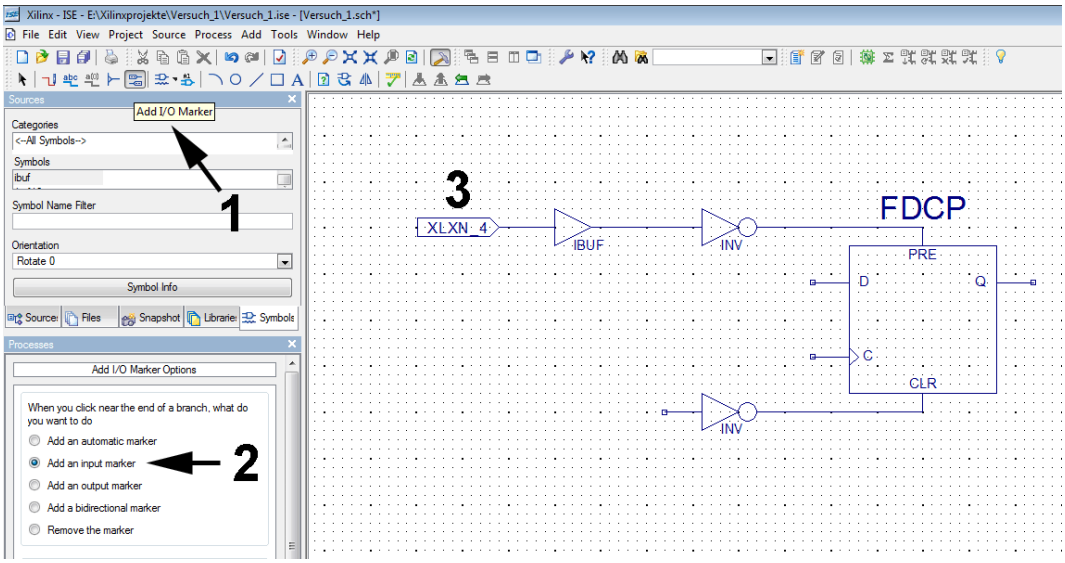


Abb. 20 Jetzt werden die Marker nachgesetzt. 1 - Funktionsauswahl. 2 - Auswahl der Art des Markers. 3 - ein Input Marker, der bereits angedockt wurde.

Die Signalbezeichner mit “XLXN” sind eine Eigenart des Entwicklungssystems. Sie werden automatisch vergeben. Auf Dauer kommt man damit nicht zurecht. Wir sollten schon Bezeichner haben, die man sich merken kann. Hierzu müssen wir die Signale umbenennen.

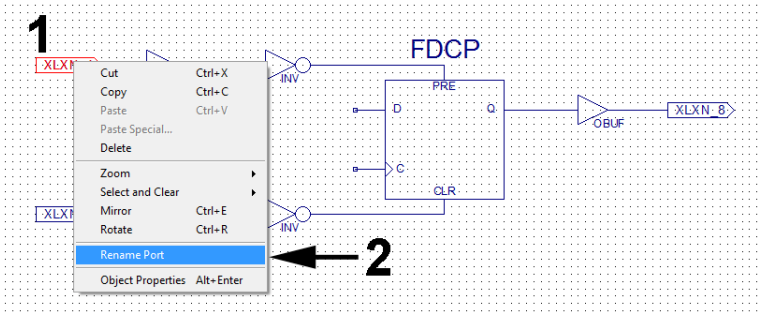


Abb. 21 Den Signalbezeichner in einem Marker ändern. 1 - Marker auswählen. Dann rechte Maustaste. 2 - wir benötigen die Funktion “Rename Port”.

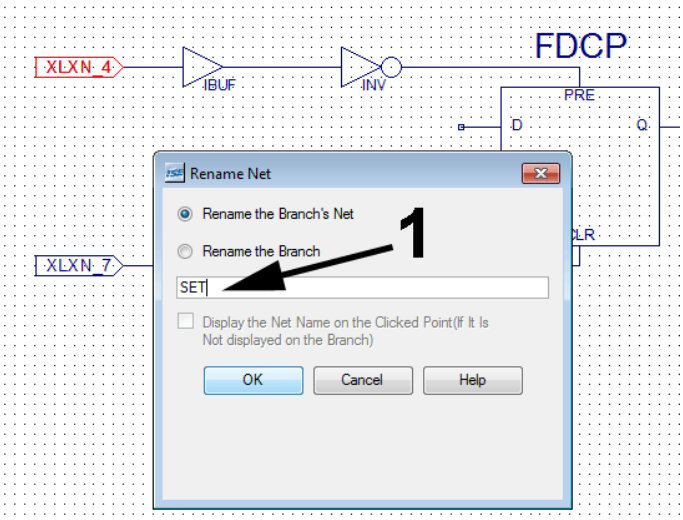


Abb. 22 Die Umbenennung. 1 - hier wird der neue Name eingegeben.

Die Schaltung hat noch offene Eingänge. Die darf es aber nicht geben. Also sind sie mit Festwerten zu beschalten.

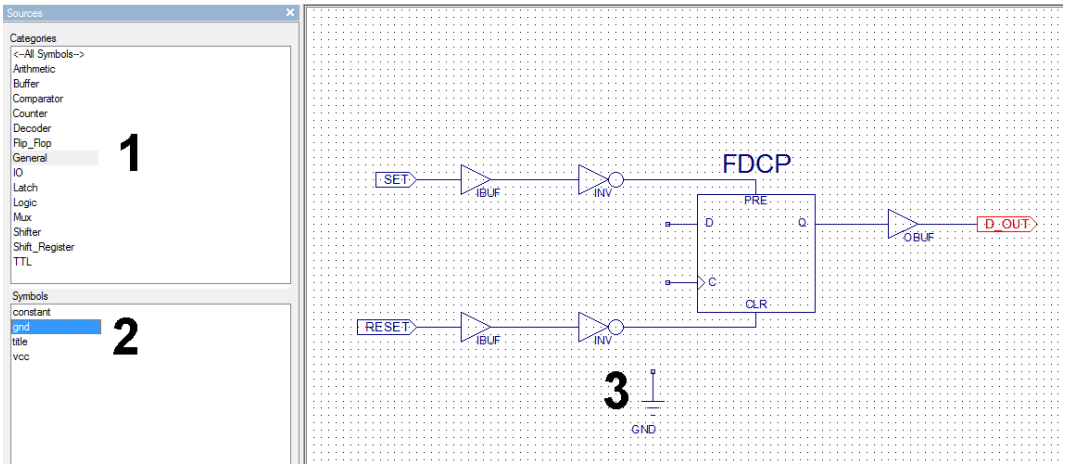


Abb. 23 Beschalten mit Festwerten. 1 die Festwerte finden wir in der Kategorie "General". 2 - **gnd** = Low, **vcc** = High. 3 - ein ausgewählter Festwert Low.

Die Low-Festwerte (*gnd*) lassen sich direkt andocken, die High-Festwerte (*vcc*) nicht. Wenn das Andocken nicht klappt oder zu unschön aussieht, sind Drahtverbindungen zu ziehen. Zumeist genügt es, nur die Anschlußpunkte auszuwählen. Das System kann die Leitungen automatisch verlegen (Autorouting). Wenn es nicht um Schönheit geht, ist das in vielen Fällen ausreichend.

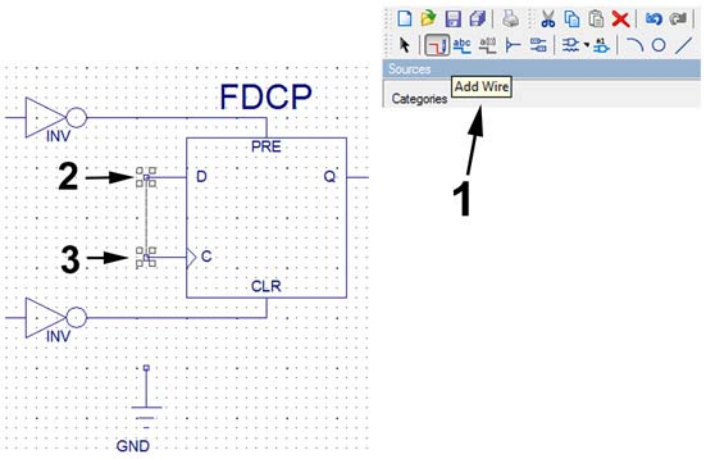


Abb. 24 Einen Draht ziehen. 1 - Funktionsauswahl. 2 - den ersten Anschluß anklicken. 3 - Verbindung zum nächsten Anschluß ziehen und dort klicken. Daß ein Anschluß richtig ausgewählt wurde, ist an den typischen vier kleinen Quadraten zu erkennen.

Die erste Teilschaltung ist fertig. Sie ist aber noch mit einem Toggle-Flipflop zu ergänzen.

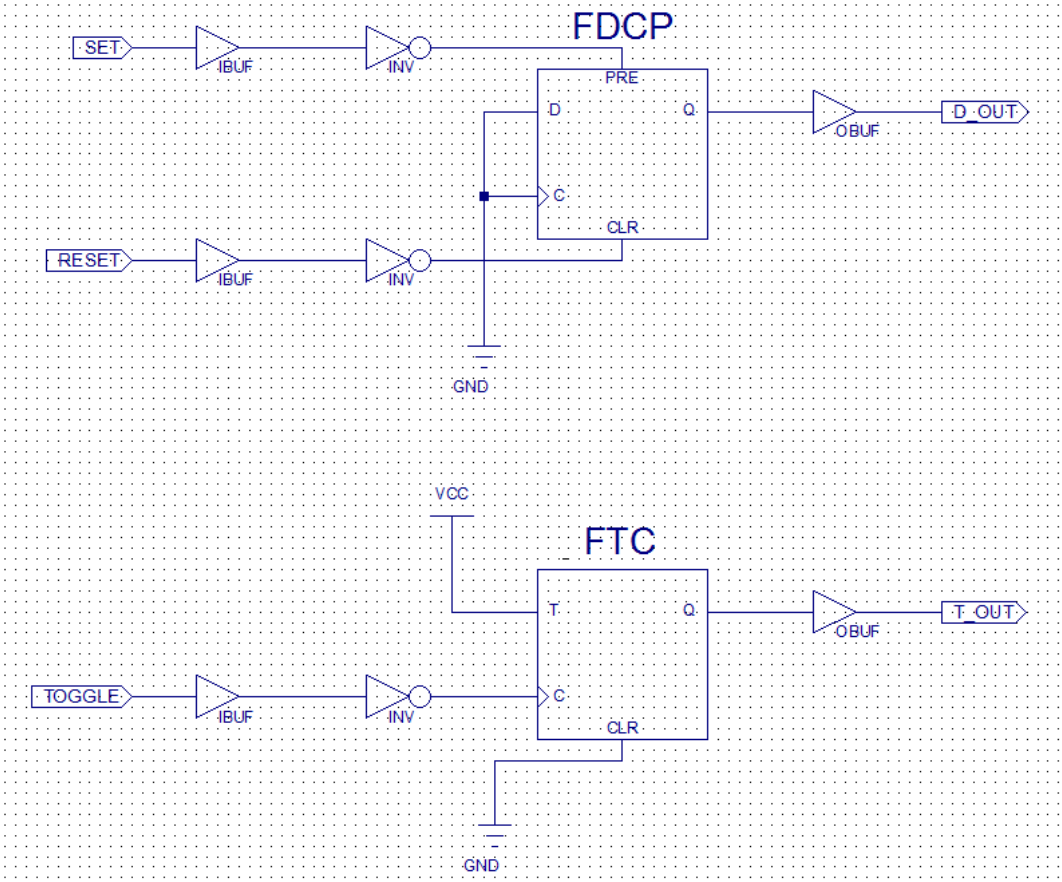


Abb. 25 Die fertig eingegebene Schaltung. 1 - in der Kategorie Flip-Flop finden wir ein passendes Toggle-Flipflop ftc. 2 - der Löscheingang muß inaktiv gehalten werden (Low), der Toggle-Eingang aktiv (High). Das Taktsignal kommt von der Taste. Deshalb ist es zu invertieren.

Jetzt kann die Schaltung implementiert werden.

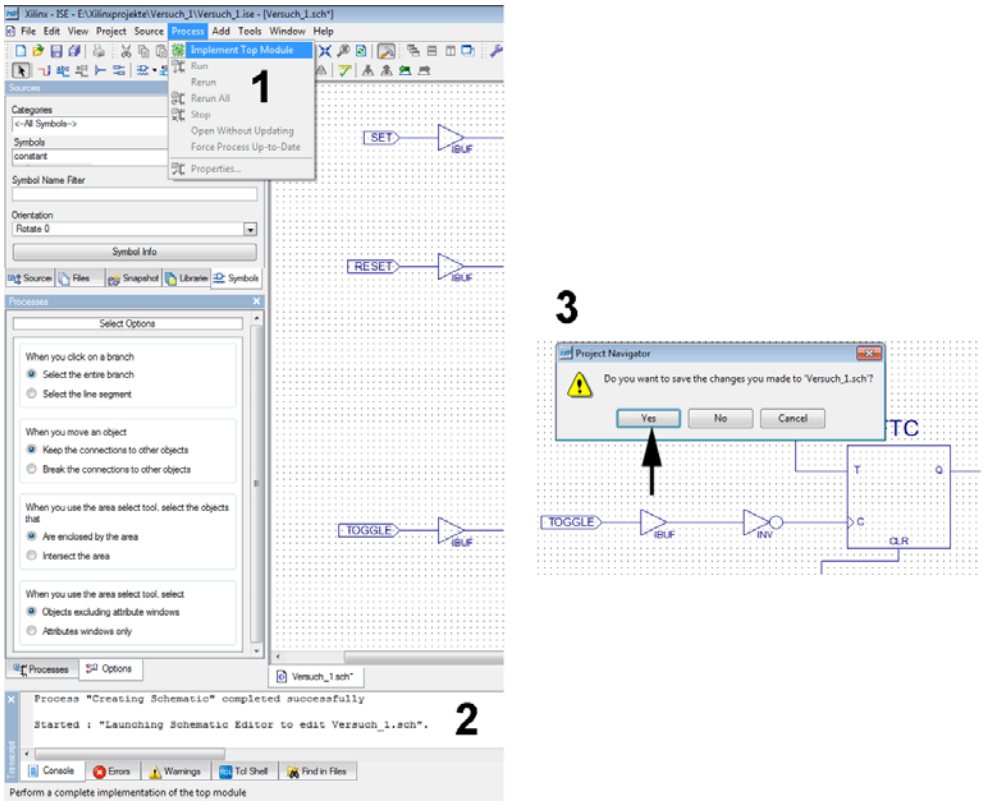


Abb. 26 Implementierung einer eingegebenen Schaltung. 1 - die Funktion heißt “Implement Top Module”. Sie ist im “Process”-Menü zu finden. 2 - Einzelheiten des Implementierungsablaufs werden im Konsolfenster (“Console”) angezeigt. 3 - diese Aufforderung sollte bestätigt werden...

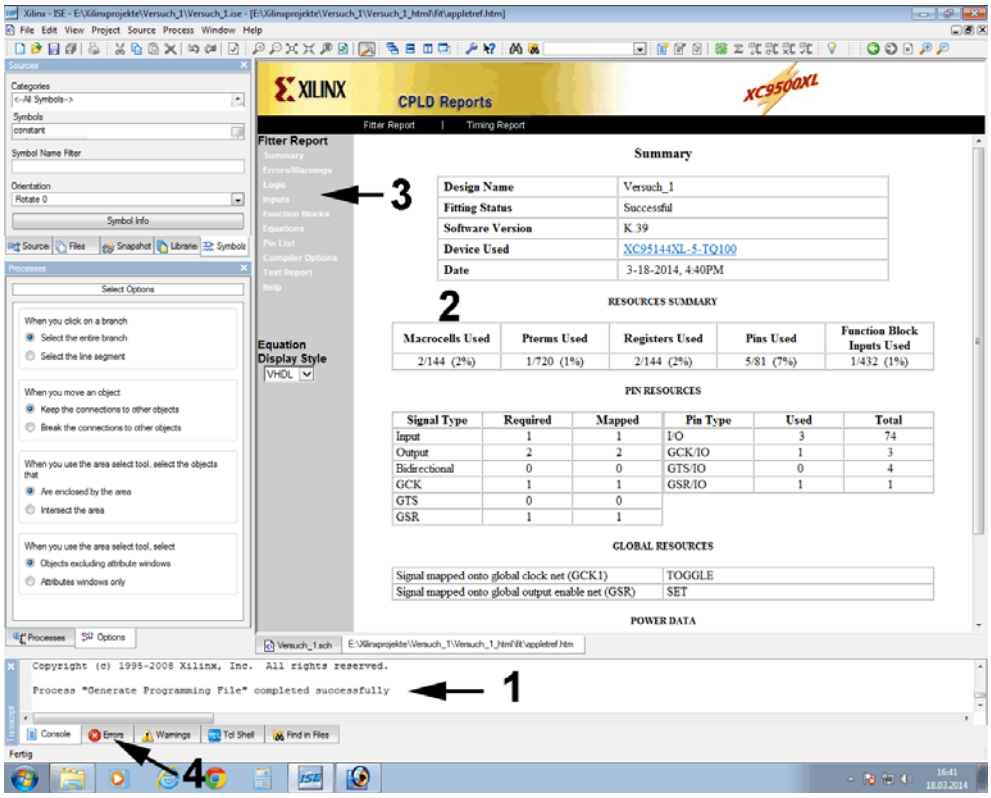


Abb. 27 So muß es am Ende aussehen. 1 - die entscheidende Erfolgsaussage. 2 - es wurden zwei Makrozellen verbraucht. Das ist o.k., denn wir haben zwei Flipflops eingesetzt. 3 - hier können detailliertere Berichte angefordert werden, darunter die Booleschen Gleichungen und die Anschlußbelegungen. 4 - wenn es nicht geklappt hat, ggf. hier nachsehen...

6. Ergänzende Einstellungen.

Sie dienen dazu, die Schaltung betriebssicher zu machen. Es kann sein, daß es ohne diese Einstellungen auch funktioniert. Im Problemfall sollte man aber wissen, wo ggf. einzugreifen ist.

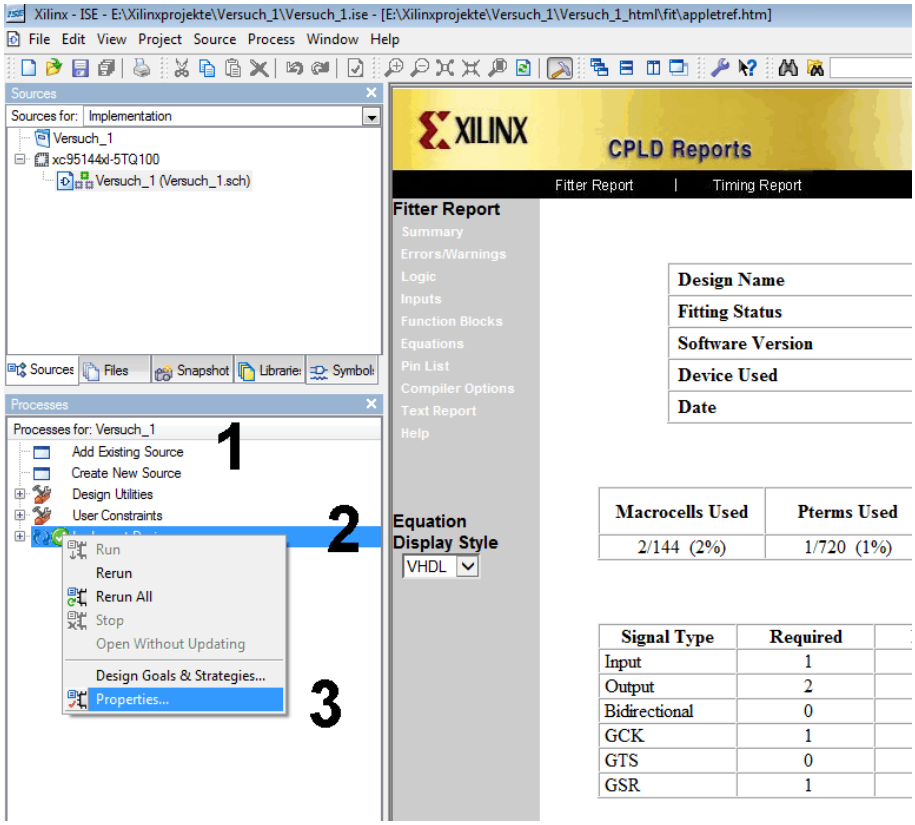


Abb. 28 Die Einstelldialoge aufrufen. 1 - den Prozeßdialog (“Processes”) in den Vordergrund holen. 2 - mit der rechten Maustaste auf “Implement Design” klicken. 3 - auf “Properties” klicken.

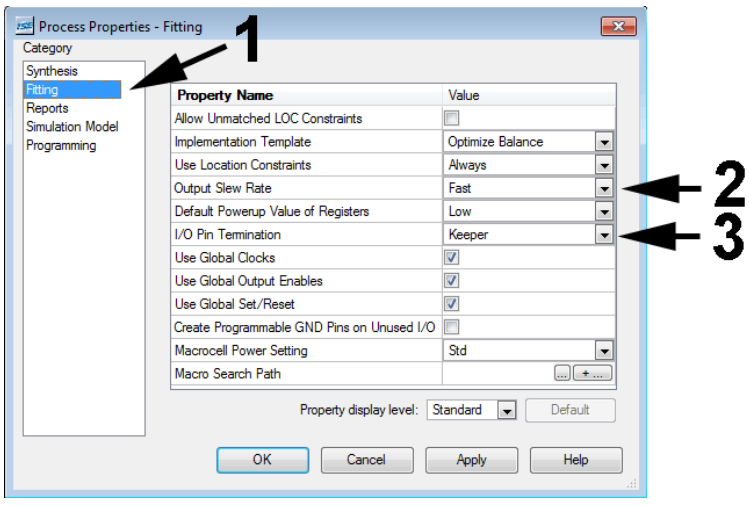


Abb. 29 Die wichtigsten Einstellungen. 1 - die Prozeßeigenschaften unter “Fitting” auswählen. 2 - hier können wir die Anstiegszeit bzw. Flankensteilheit (Slew Rate) der Ausgangssignale beeinflussen. 3 - Eingänge dürfen keinesfalls offen sein. Hier werden die E-A-Anschlüsse standardmäßig mit einer Halteschaltung (“Keeper”) auf einem festen Logikpegel gehalten. Im Versuchsaufbau kann man es üblicherweise so lassen.

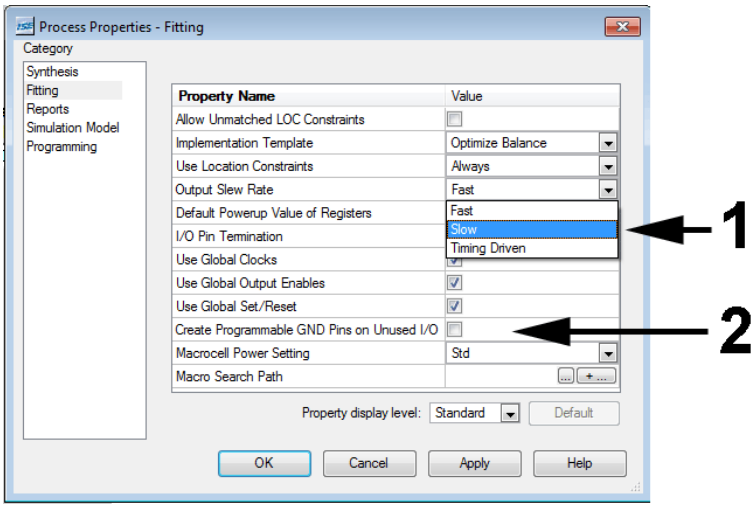


Abb. 30 Typische Änderungen. 1 - Verlängern der Anstiegszeit (von "Fast" auf "Slow"), um das Übersprechen (Crosstalk) gering zu halten. 2 - in fertigen Anwendungsschaltungen ggf. die ungenutzen Anschlüsse mit Masse verbinden. *Hinweis:* Es gibt auch Schaltkreise ohne Halteschaltungen (Keepers). Werden solche Typen eingesetzt, ist diese Einstellung unbedingt zu empfehlen.

7. Die Signale auf bestimmte Anschlüsse legen.

Das Entwicklungssystem belegt die Schaltkreisanschlüsse automatisch. Auf der Versuchsplattform sind aber die Tasten, LEDs usw. mit bestimmten Anschlüssen verbunden. Damit alles zusammenpaßt, braucht man eine weitere Quelldatei, das sog. Constraints File. *Hinweis:* Eine Übersicht über alle genutzten Anschlüsse befindet sich am Ende dieser Versuchsanleitung.

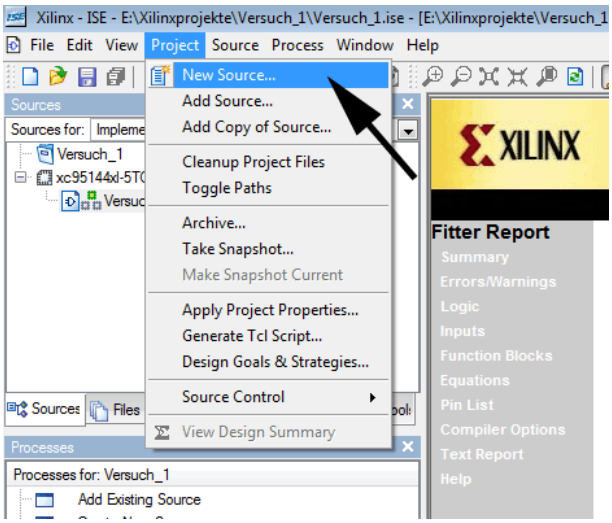


Abb. 31 Auswahl des Quelldateiassistenten über "Project" – "New Source".

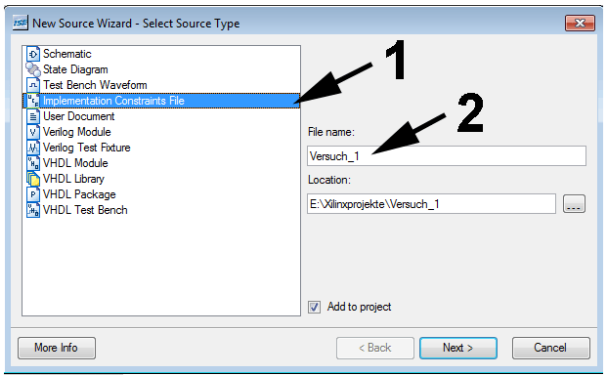


Abb. 32 Die Datei wird erzeugt. 1 - wir brauchen eine Constraints-Datei ("Implementation Constraints File". 2 - Eingabe des Dateinamens (beliebig; die Dateiendung wird automatisch gebildet).

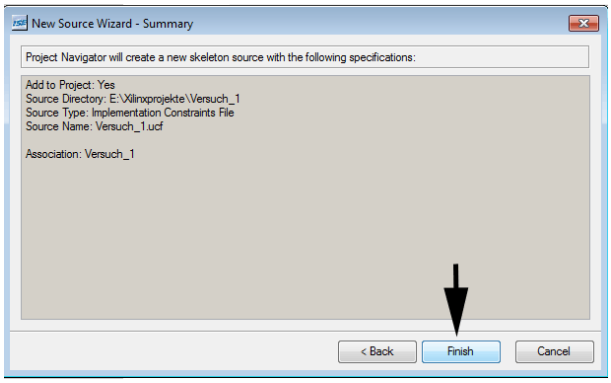


Abb. 33 Anschließend ist dieser Dialog zu durchlaufen.

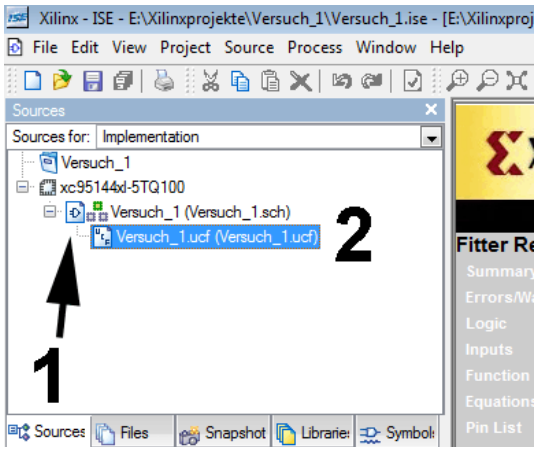


Abb. 34 Die neue Datei erscheint im Quellenverzeichnis. 1 - hier klicken; 2 - die neue Datei.

Wenn wir auf die Datei doppelklicken, sollte sie geöffnet werden, und zwar mit einem passenden Editor. Wir arbeiten hier mit dem sog. *Floorplan Editor*. Wird statt dessen ein anderes Programm gestartet, sind die Einstellungen zu ändern. Wird nichts gestartet, ist der Floorplaneditor im Programmstartmenü (Windows) zu aktivieren. Das Programm heißt *Pace*.

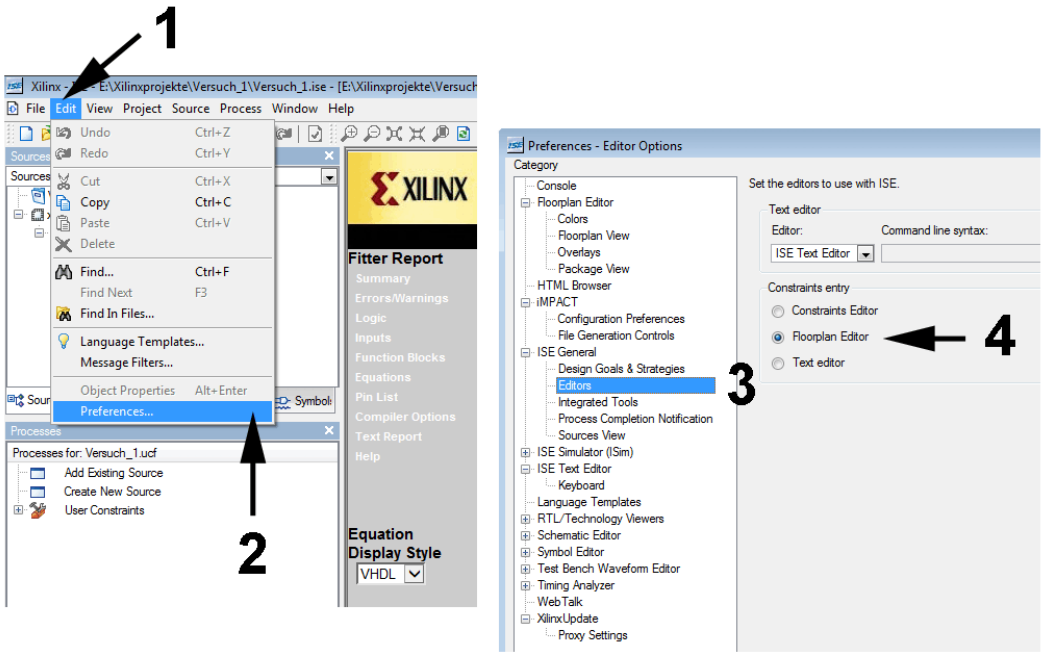


Abb. 35 So wird – bei Bedarf – der richtige Editor ausgewählt. 1 - Edit-Dalog. 2 - Vorzugseinstellungen (Preferences). 3 - Editorauswahl. 4 - Auswahl des Floorplan-Editors.

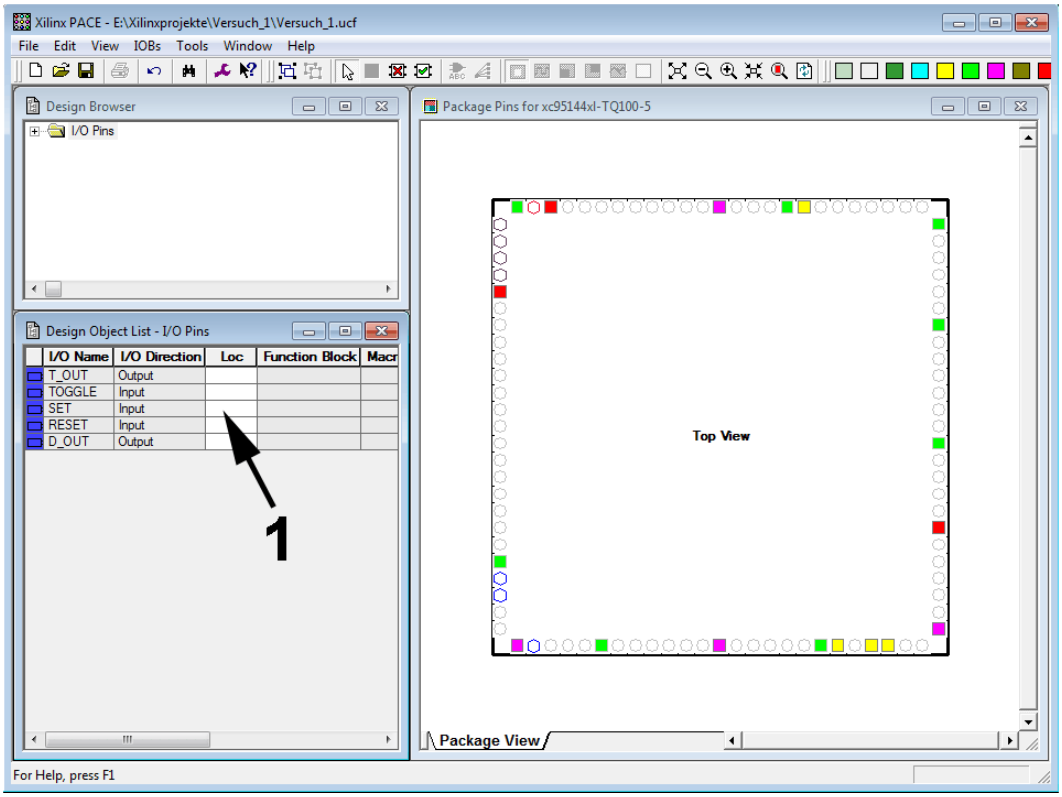


Abb. 36 Der geöffnete Floorplan-Editor. 1 - hier sind die Pin-Nummern einzutragen.

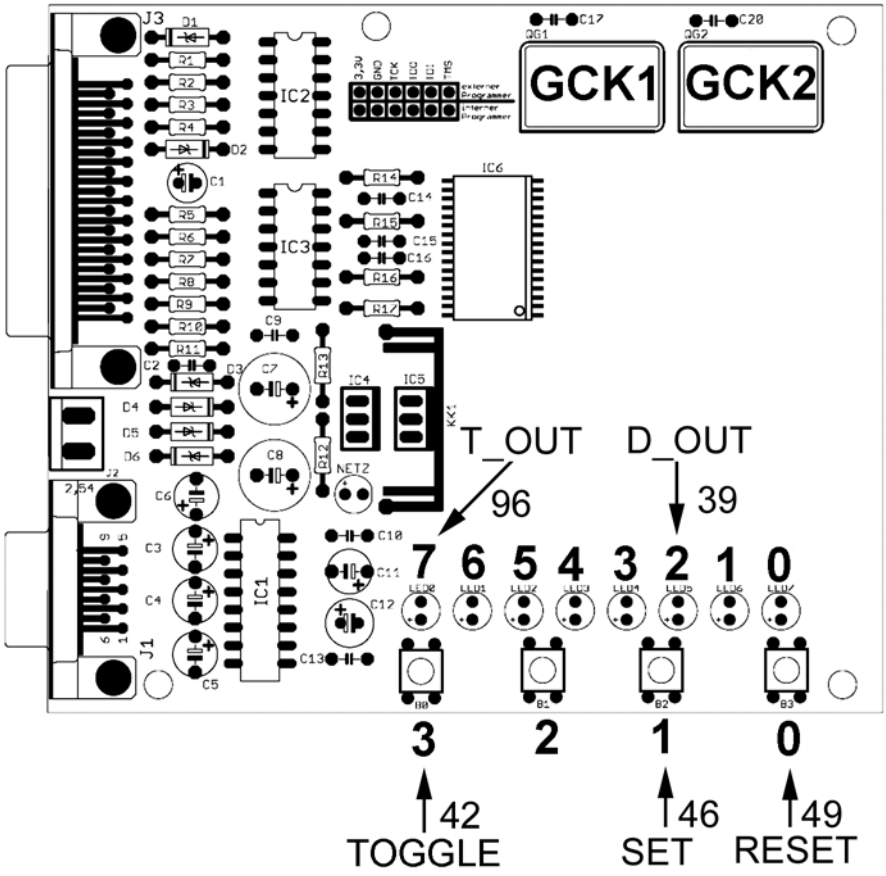


Abb. 37 Eine zweckmäßige Anschlußbelegung.

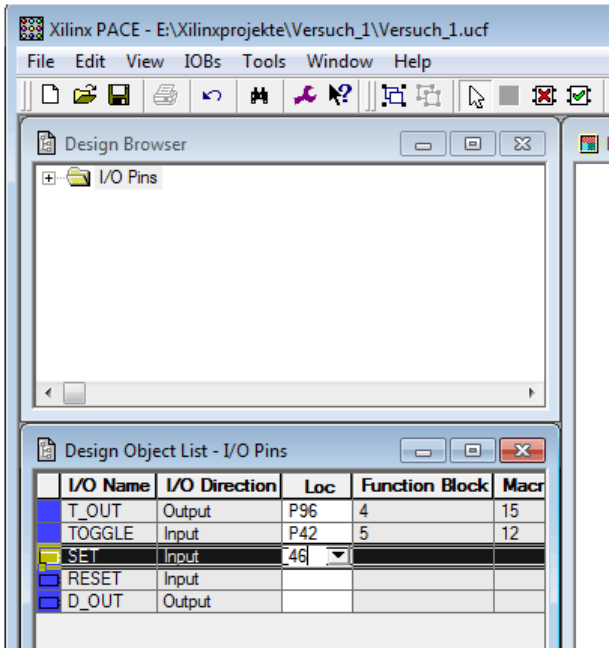


Abb. 38 Die Pin-Nummern werden eingetippt. Es genügen die Zahlen; die “P”s fügt der Editor automatisch hinzu.

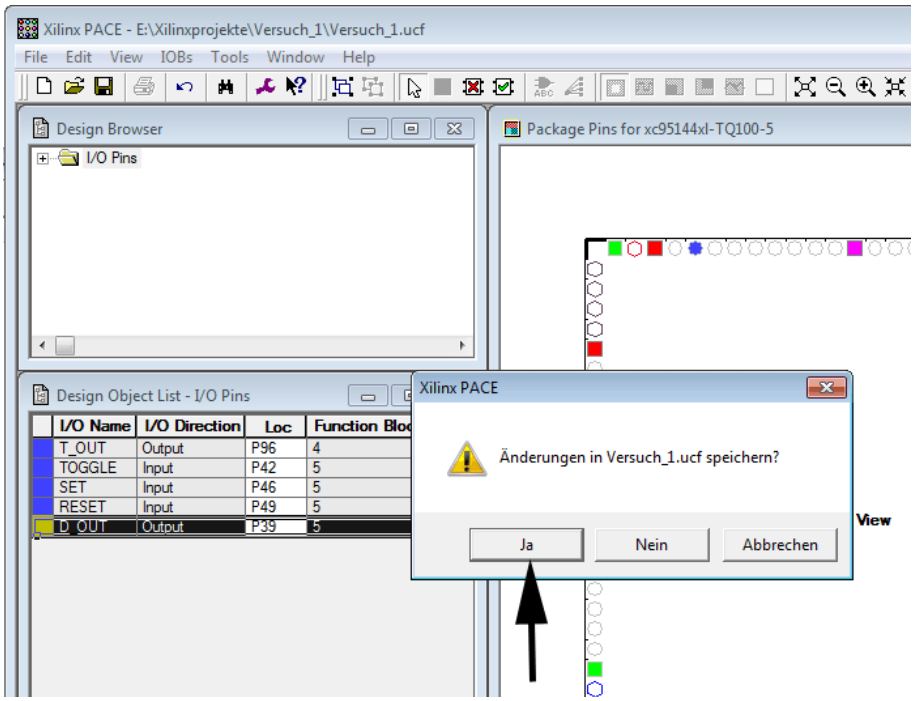


Abb. 39 Sind alle Pin-Nummern eingebene, so ist die Datei zu schließen. Die abschließende Speicheranfrage bestätigen.

8. Den Schaltkreis programmieren.

Wurden die Anschlußbelegungen eingegeben, ist die Implementierung erneut zu starten (Process – Implement Top Module). Nach deren Durchlauf stehen die Programmierdaten bereit. Jetzt kann das Programmierprogramm aufgerufen werden. Es heißt *IMPACT*. Versuchsplattform einschalten. Ggf. Verbindung zur Parallelschnittstelle des PC herstellen (Parallelportkabel anstecken).

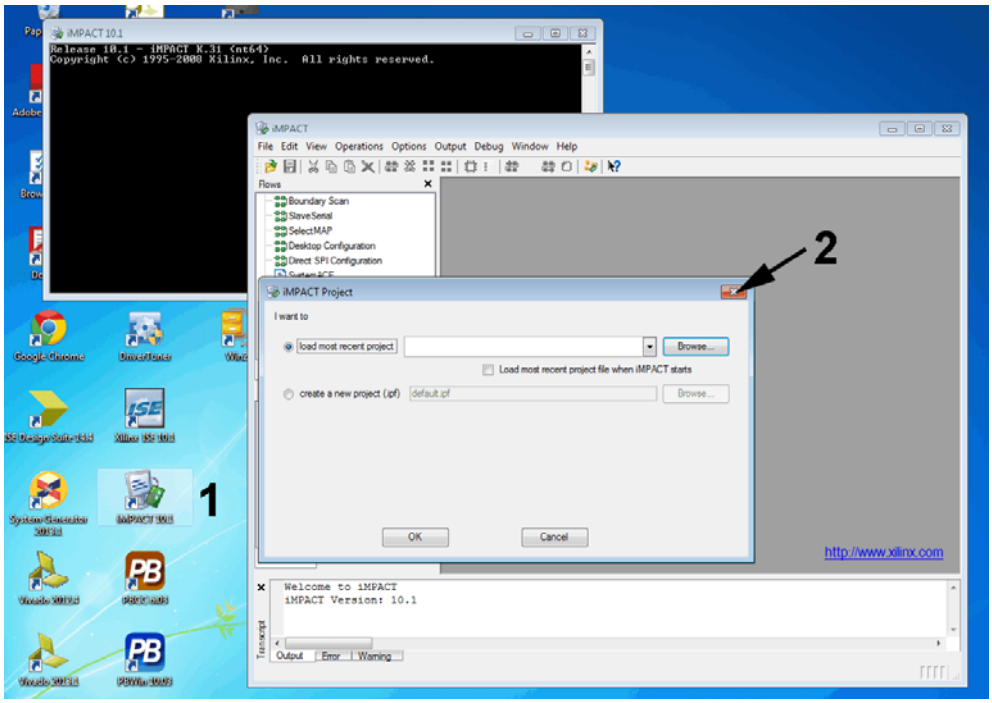


Abb. 40 Den Programmierer aufrufen. 1 - IMPACT starten. 2 - die IMPACT-Projekte haben mit den ISE-Projekten nichts zu tun. Brauchen wir hier nicht. Fenster schließen.

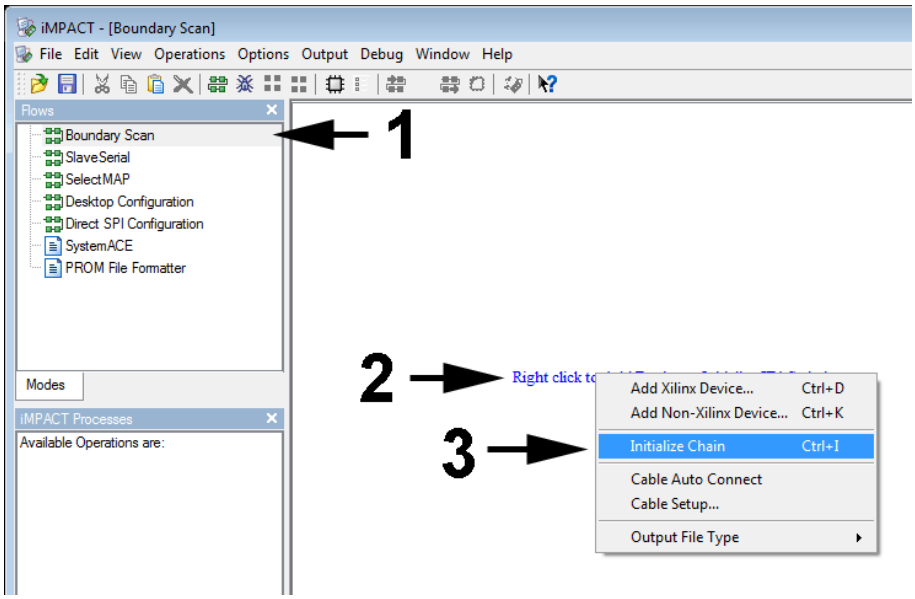


Abb. 41 So wird die Programmierschnittstelle angesprochen. 1 - Boundary Scan auswählen. 2 - mit der rechten Maustaste draufklicken. 3 - Funktion “Initialize Chain” ausführen.

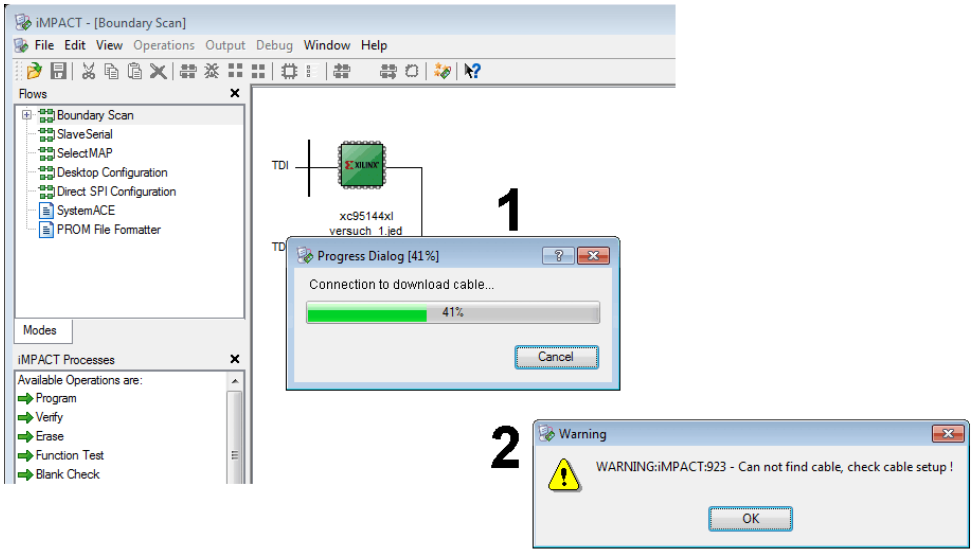


Abb. 42 Der Programmierer versucht, die Verbindung herzustellen. 1 - der Verbindungsaufbau läuft. 2 - diese abschließende Nachricht erscheint, wenn die Verbindung nicht hergestellt werden konnte.

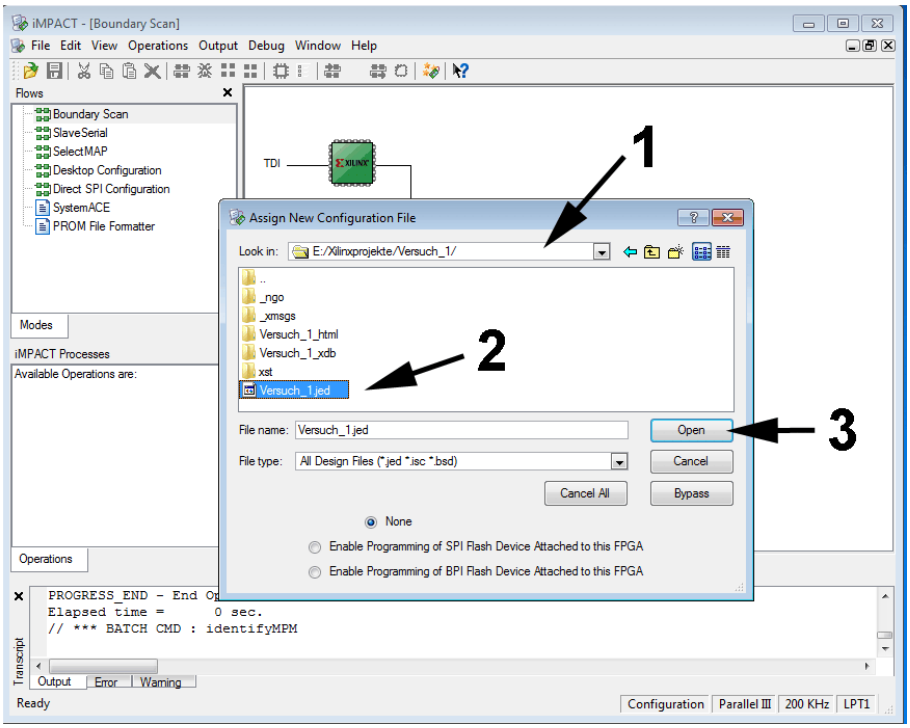


Abb. 43 Wenn die Verbindung in Ordnung ist, erscheinen ein grünes Schaltkreissymbol sowie ein Dateidialog. Dann muß die Programmierdatei gesucht werden. Die Dateiendung heißt *.jed* (Jedec File). 1 - Verzeichnis des Projekts suchen. 2 - Jedec-Datei auswählen. 3 - weiter...

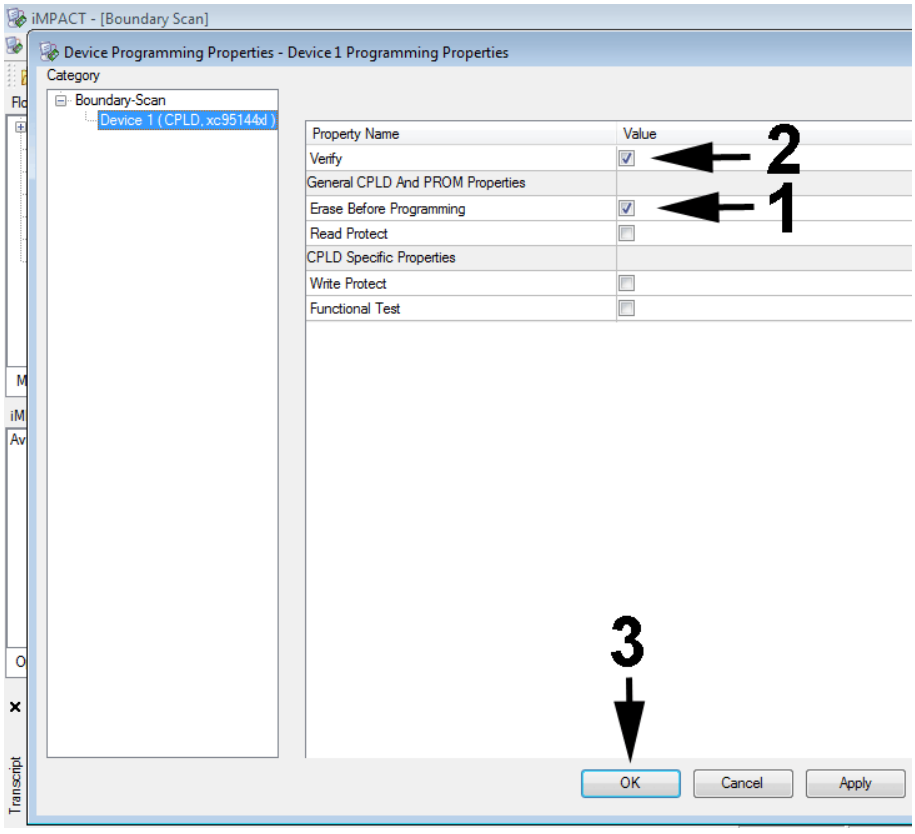


Abb. 44 Die Funktionsauswahl. 1 - vor dem Programmieren löschen; 2 - nach dem Programmieren auf Richtigkeit überprüfen. 3 - weiter...

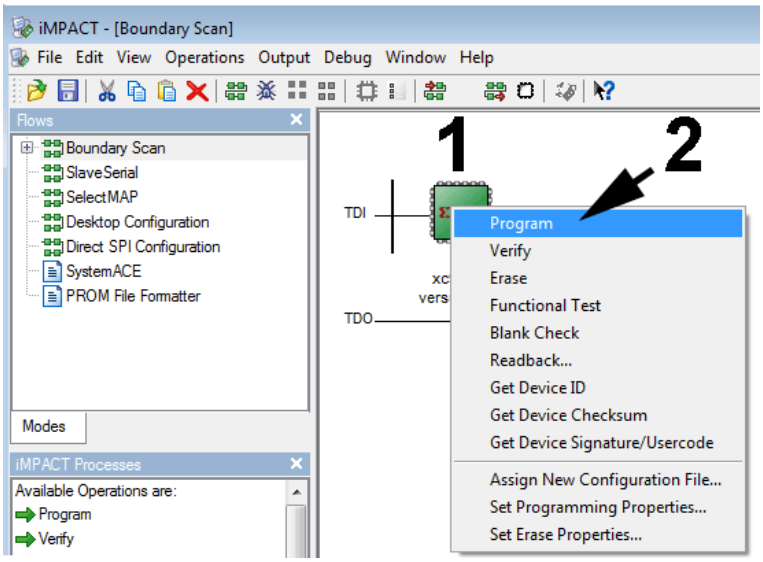


Abb. 45 Jetzt wird das Programmieren ausgelöst. 1 - mit der rechten Maustaste auf das Schaltkreissymbol klicken. 2 - dann auf “Program” klicken.

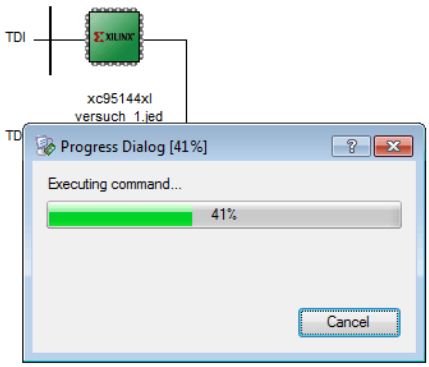


Abb. 46 Die Programmierung läuft.

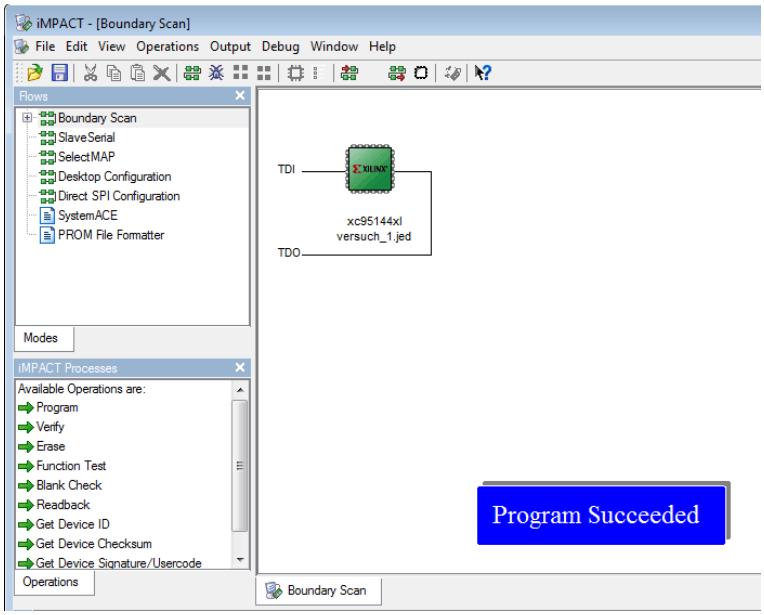


Abb. 47 Am Ende sollte es so aussehen. Nun kann die Schaltung ausprobiert werden. Das IMPACT-Programm nicht schließen, sondern nur minimieren. Bei den nachfolgenden Versuchen genügt dann ein erneutes Anklicken.

Aufgabe 2: Passende Taktfrequenzen herstellen

Das Toggle-Flipflop wird nicht immer richtig funktionieren. Das liegt daran, weil die Taste prellt. Man kann Tasten entprellen, indem man sie mit einem Takt von einigen zehn bis wenigen hundert Hz synchronisiert. Auch brauchen wir einen langsamen Takt, um beispielsweise ein Lauflicht vorführen zu können. Die Versuchsplattform hat aber nur einen Grundtakt von 16 MHz. Niedrigere Taktfrequenzen sollen erzeugt werden, indem wir diesen Takt im Verhältnis $1:2^{24} = 16\,777\,216$ teilen. Hierzu bauen wir einen 24-Bit-Zähler auf. Passende Zähler finden wir in der Kategorie *Counter*.

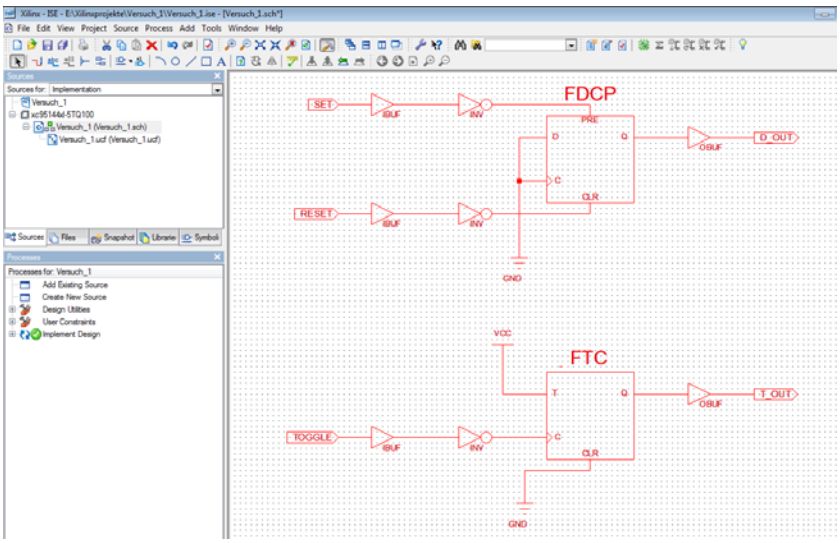


Abb. 48 Zurück zu Xilinx. Den Schaltplan aufrufen und die gesamte Schaltung löschen: 1. Auswahlfunktion. 2. Schaltung mit Cursor umfahren (Auswahlrechteck; hierzu linke Maustaste gedrückt halten). Schaltplan wird rot. 3. Löschtaste betätigen (Entf / DEL).

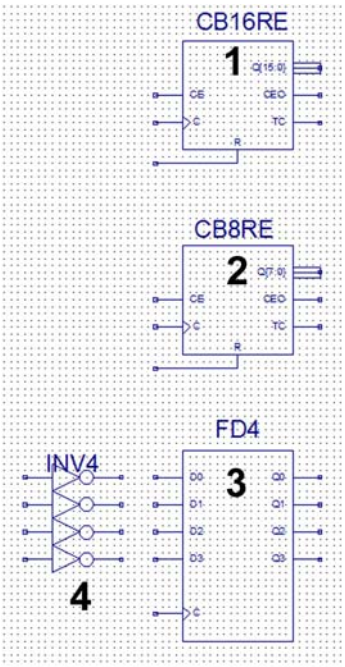


Abb. 49 Erst einmal die Funktionselemente bereitlegen. 1 - 16-Bit-Zähler. 2 - 8-Bit-Zähler. In weiser Voraussicht auf weitere Versuchsaufgaben verwenden wir Zähler mit *synchronem* Rücksetzen (R = Reset). 3 - wir entprellen gleich alle vier Tasten der Versuchsplattform (sozusagen auf Vorrat). In der Kategorie “Flip-Flop” finden wir ein Register mit vier D-Flipflops. 4 - ergänzend dazu aus der Kategorie “Logic” Negatoren im Viererpack (inv4).

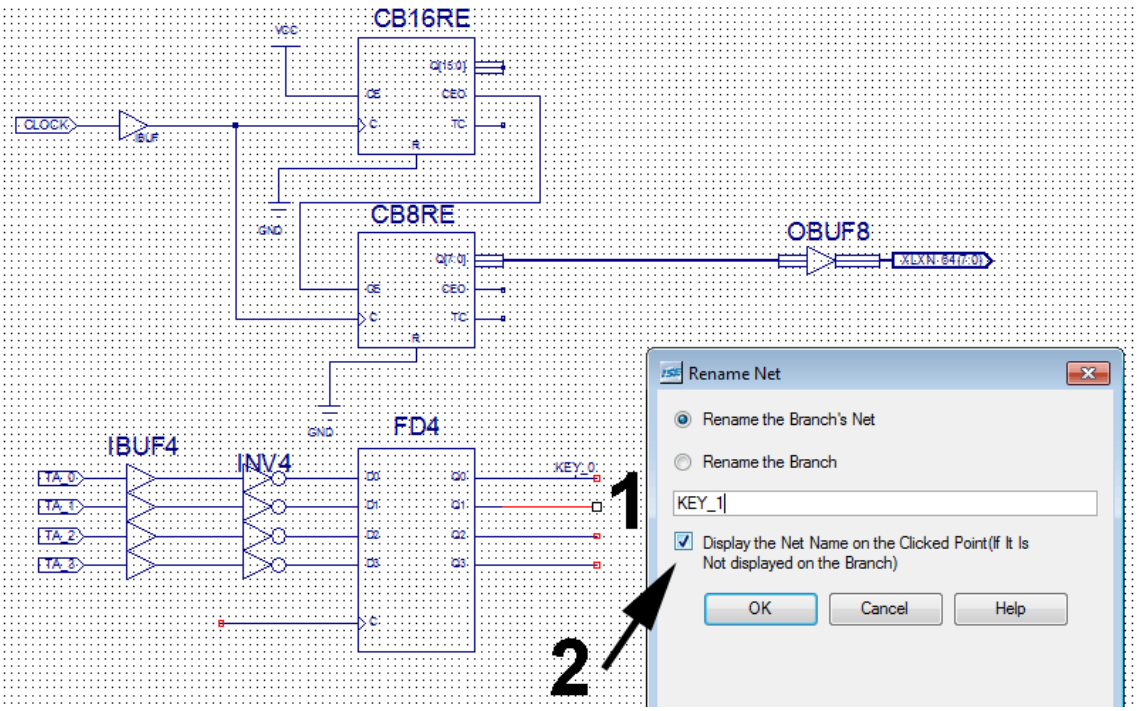


Abb. 50 Die (fast) fertige Schaltung. Die Ausgangssignale der höchstwertigen acht Zählerstellen sollen auf die LEDs geführt werden. Auch die Puffer gibt es in Vierfach- und Achtfachausführung. Einige Signale müssen noch benannt werden. 1 - Dieses Signal soll KEY_1 heißen. 2 - damit der Signalname im Schaltplan zu sehen ist, muß diese Funktion ausgewählt werden.

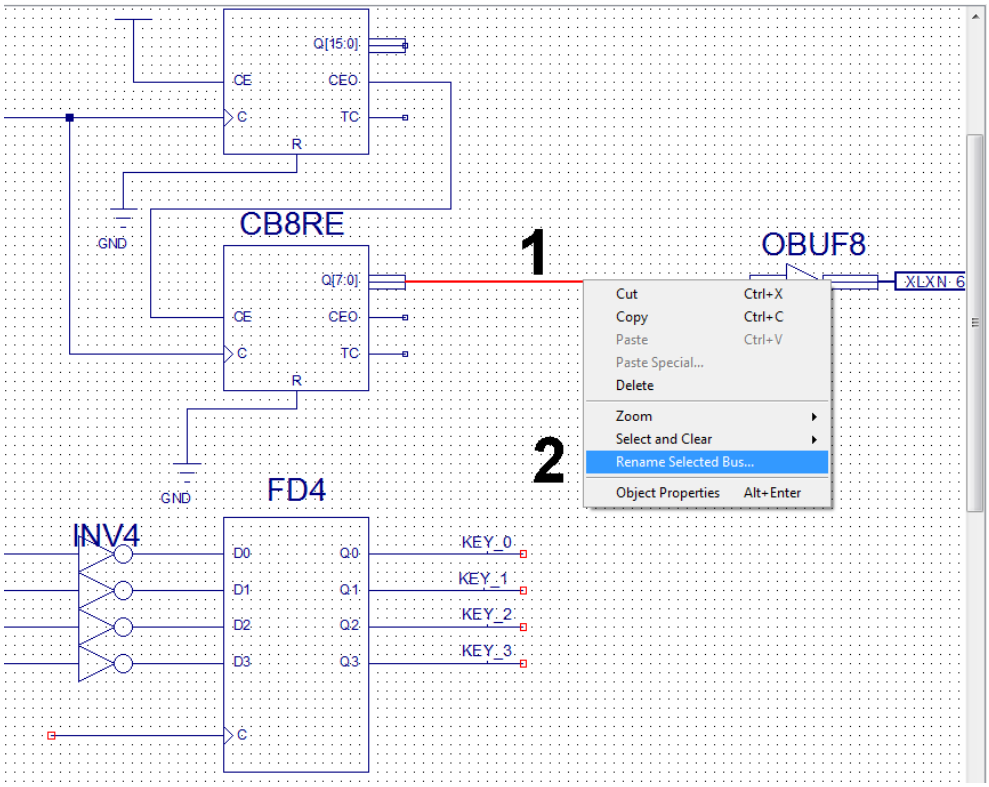


Abb. 51 Etwas Neues – ein Bus. Es ist eine Zusammenfassung mehrerer Signale, eine Art Kabelbaum. So erhält er seinen Namen: 1 - auf den Bus klicken (wird rot). Rechte Maustaste. 2 - dann auf “Rename Selected Bus” klicken.

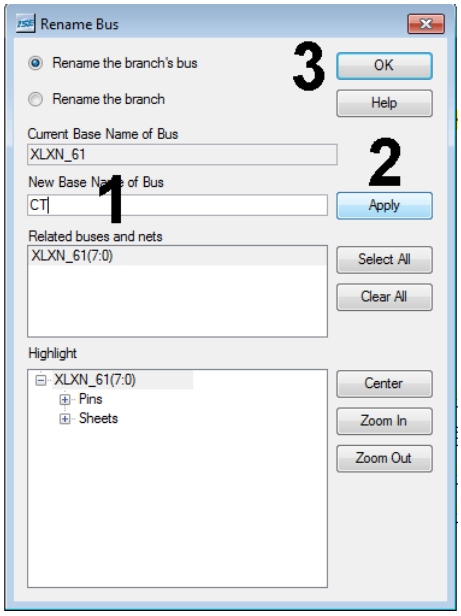


Abb. 52 Jetzt wird der Name einegeben. Der Bus soll CT heißen. Die acht Signale werden von 7 bis 0 durchnumeriert. 1 - nur den Namen eintippen. Die Indices der Signalleitungen werden automatisch hinzugefügt. 2 - "Apply" anklicken. 3 - "OK" anklickcn.

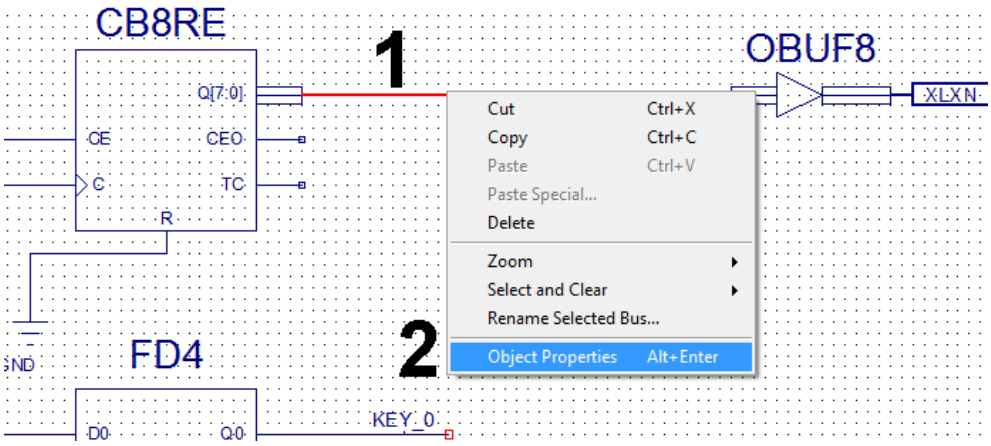


Abb. 53 Der Name ist aber noch nicht zu sehen. Wie machen wir ihn sichtbar? Den Bus nochmals anklicken. Rechte Maustaste. Dann auf "Object Properties" klicken.

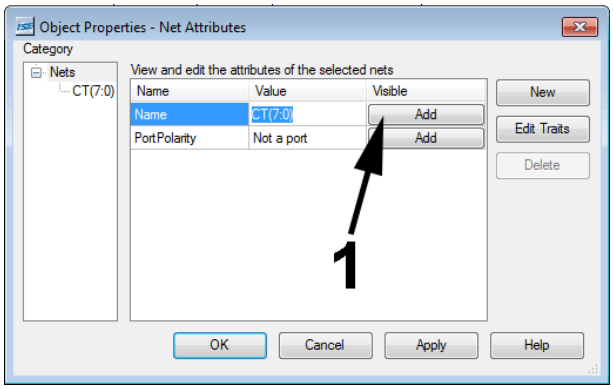


Abb. 54 Der Name soll hinzugefügt werden. Also auf "Add" klicken.

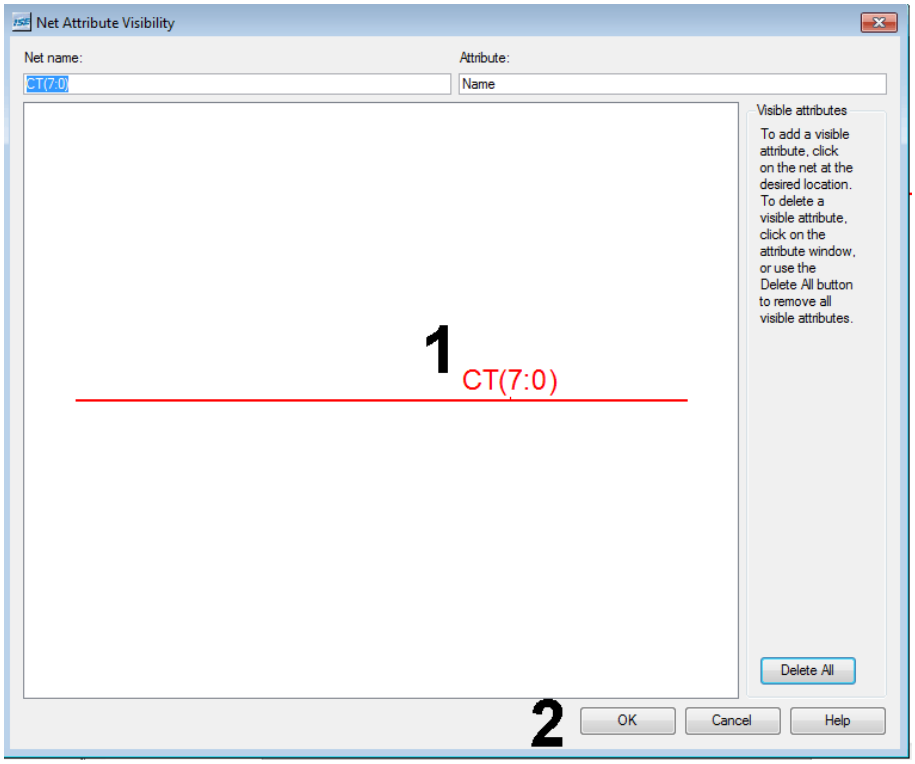


Abb. 55 Es erscheint der Leitungswegausschnitt aus dem Schaltplan. 1 - auf den Ort klicken, an dem der Name erscheinen soll (kann später – im Schaltplan – immer noch verschoben werden). 2 - dann auf “OK” klicken.

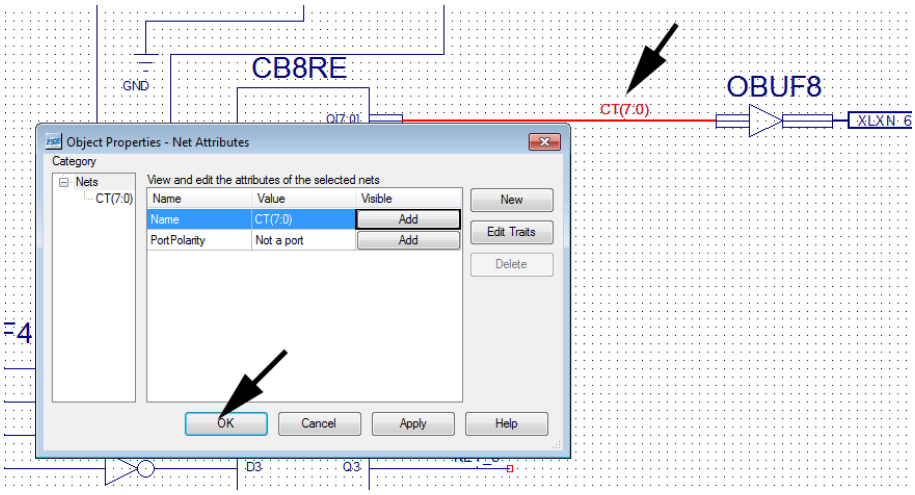


Abb. 56 Der Busname erscheint im Schaltplan. Jetzt auf “OK” klicken. Geschafft...

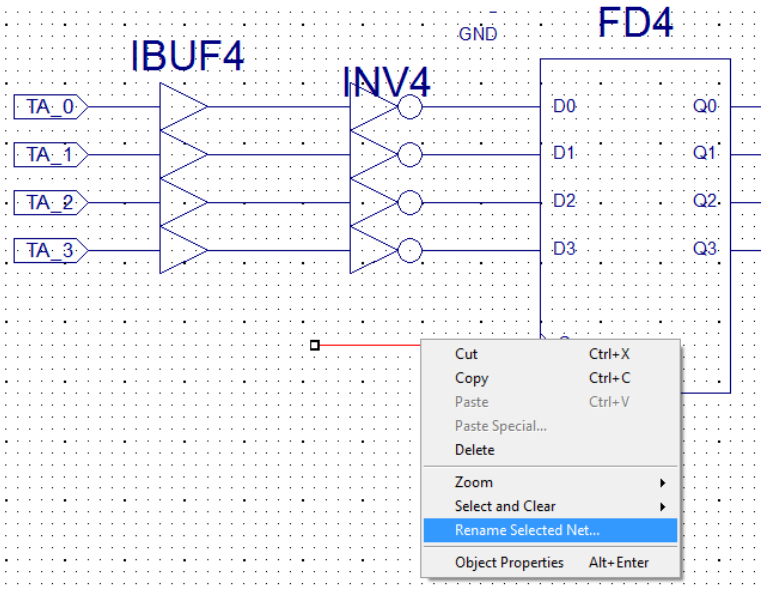


Abb. 57 Nun kann der Takteingang der Entprellflipflops benannt werden. Wir schließen ihn an den dritten Ausgang des Zählers an. Es ist die Busleitung CT(2). Leitung anklicken. Rechte Maustaste, dann “Rename Selected Net”.

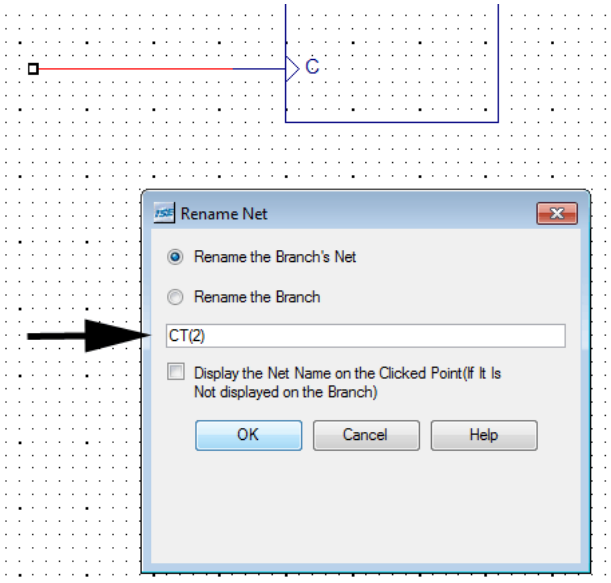


Abb. 58 Den Namen der Busleitung eintragen. Dann “OK”.

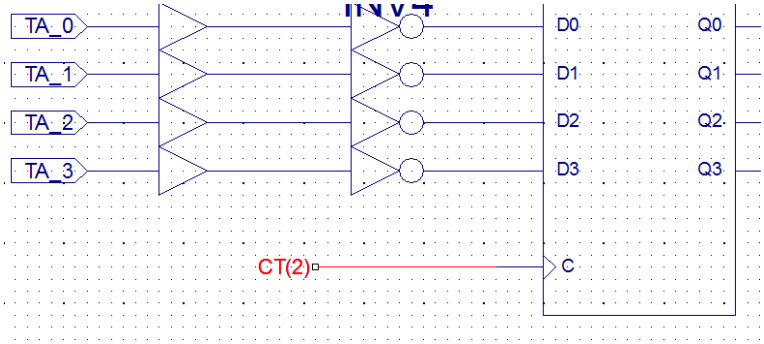


Abb. 59 In solchen Fällen erscheint der Name automatisch am Anfang der Leitung.

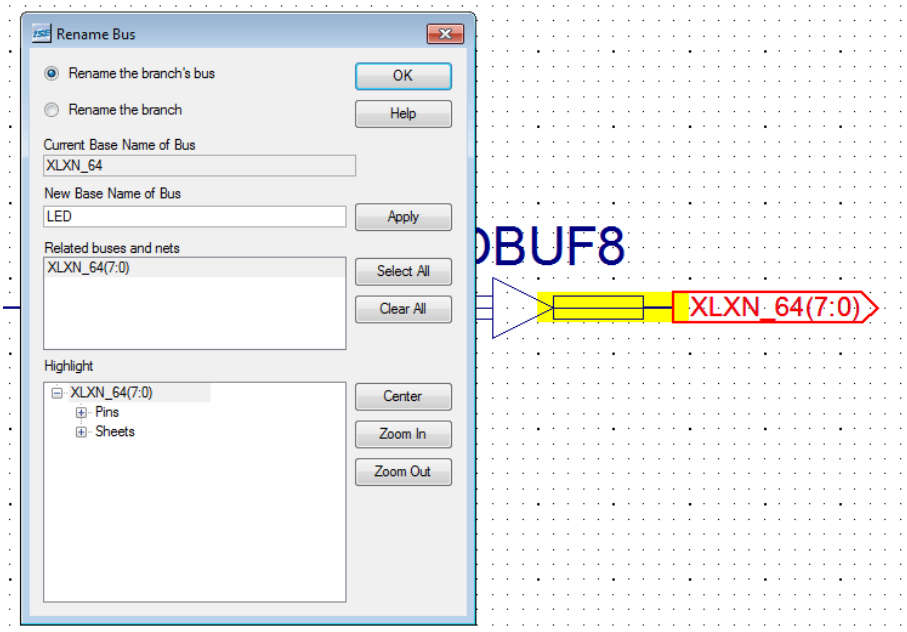


Abb. 60 Nun erhalten auch die Ausgänge ihre Namen. Es ist ein Bus für die acht LEDs. Er soll deshalb LED(7:0) heißen. Auch hier: 1. eintragen, 2. “Apply”, 3. “OK”.

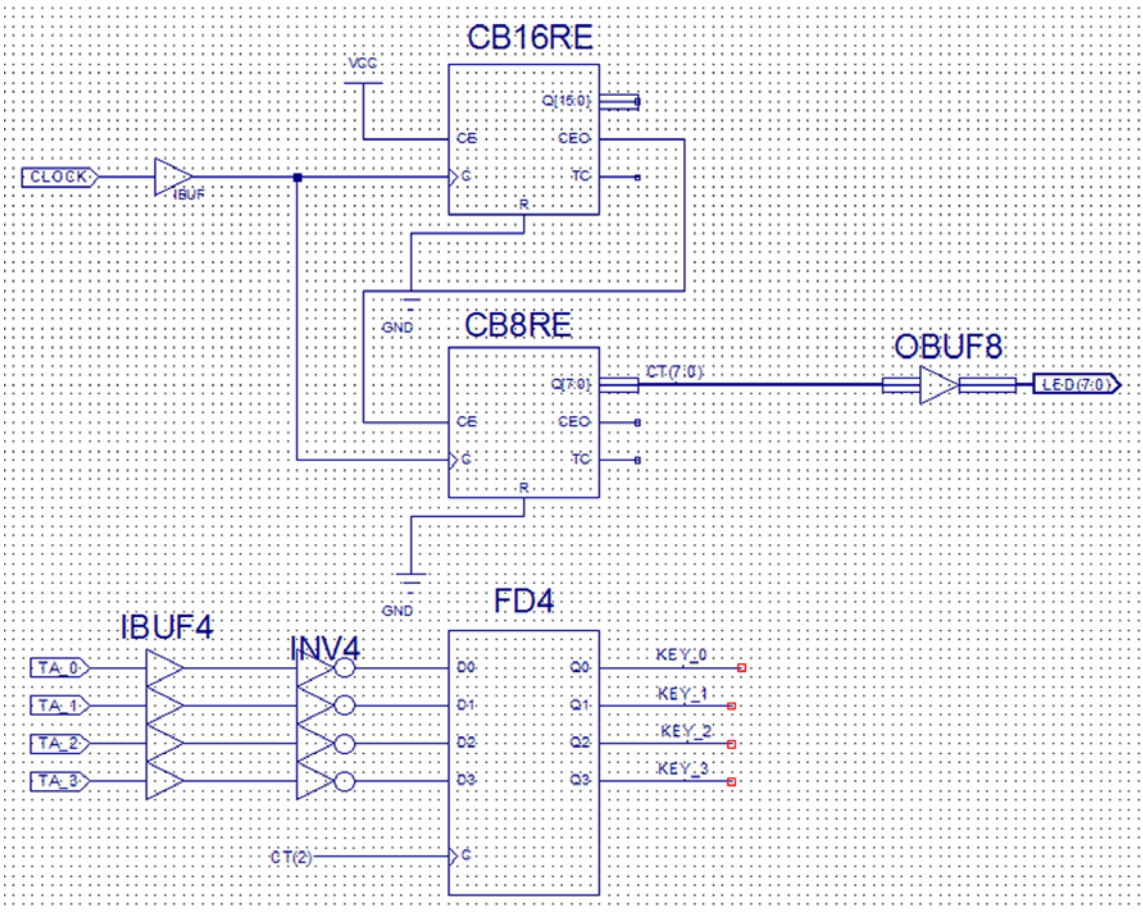


Abb. 61 Die fertige Schaltung.

Jetzt die Constraints-Datei aufrufen und alle Pin-Nummern löschen. Die Datei wieder schließen (speichern). Nun alles implementieren. Dann die Constraints-Datei wieder ausfüllen. Erneut implementieren. Schaltkreis programmieren An den LEDs muß das typische Zählverhalten zu beobachten sein (ganz links am langsamsten blinken, ganz rechts flimmern).

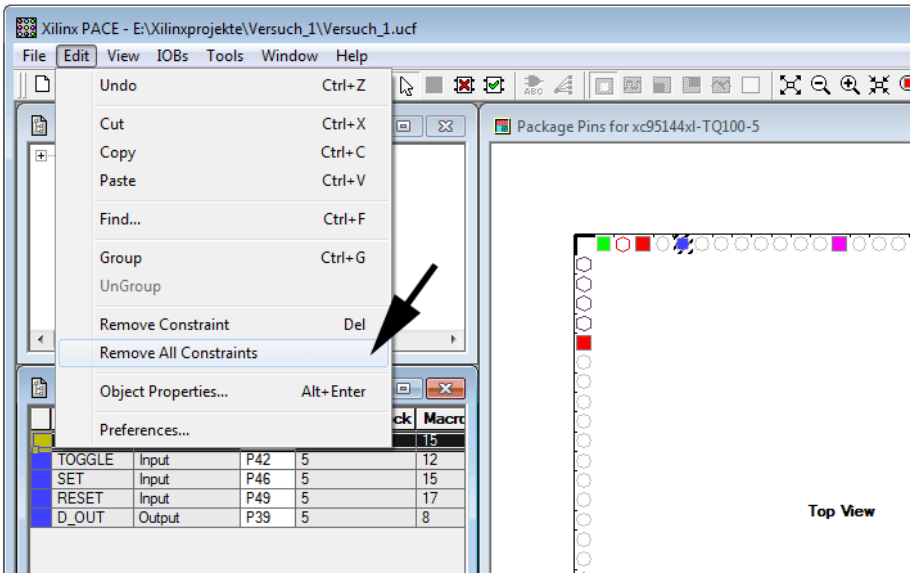


Abb. 62 Mit dieser Funktion können alle Pin-Nummern auf einmal gelöscht werden.

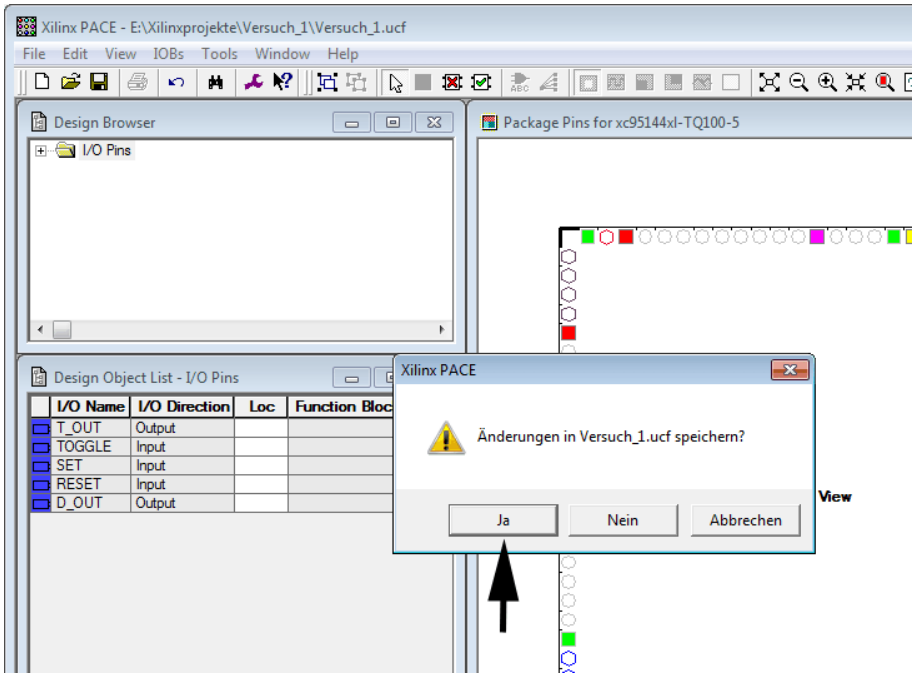


Abb. 63 Nun die Konfigurationsdatei ohne Pin-Nummern speichern. Dann das Projekt implementieren. *Hinweis:* In der Constraints-Datei dürfen nur solche Signale Pin-Nummern haben, die es auch im Schaltplan gibt. Ansonsten bricht die Implementierung mit einer Fehlermeldung ab.

Design Object List - I/O Pins

	I/O Name	I/O Direction	Loc	Function Block	Macr
	TA_3	Input	P42	5	12
	TA_2	Input	P43	5	14
	TA_1	Input	P46	5	15
	TA_0	Input	P49	5	17
	LED<7>	Output	P96	4	15
	LED<6>	Output	P97	4	17
	LED<5>	Output	P35	5	2
	LED<4>	Output	P36	5	5
	LED<3>	Output	P37	5	6
	LED<2>	Output	P39	5	8
	LED<1>	Output	P40	5	9
	LED<0>	Output	P41	5	11
	CLOCK	Input	P22	1	17

Abb. 64 Die Constraints-Datei erneut aufrufen und die neuen Pin-Nummern eingeben. Speichern, dann erneut implementieren.

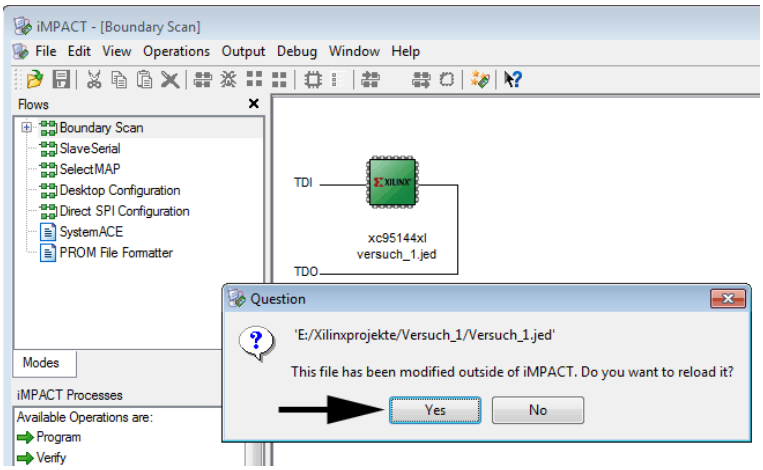


Abb. 65 Wird IMPACT wieder aktiviert, so erscheint diese Anfrage. Sie muß mit “Yes” bestätigt werden, da die Programmierdatei vom Entwicklungssystem neu erzeugt wurde. Dann wird erneut programmiert. Erscheint die Anfrage nicht, ist etwas falsch gelaufen...

Aufgabe 3: Lauflicht

Die Schaltung wird um ein Schieberegister ergänzt, dessen Ausgänge an die LEDs angeschlossen werden. Um die Benennungen zu halten, wird zunächst der *Output Buffer* gelöscht.

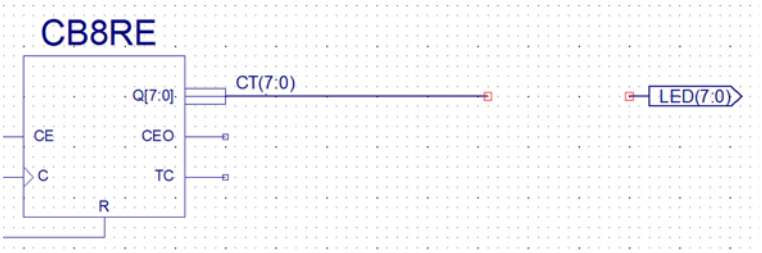


Abb. 66 Ein keiner Trick: Wenn wir einen Signalweg nur anders anschließen, nicht aber komplett löschen und dann neu zeichnen wollen, entfernen wir ein Funktionselement, das in diesem Signalweg liegt. Hier ist es der Ausgangspuffer.

Durch Löschen des Ausgangspuffers wurde der Signalweg getrennt, so daß nunmehr das Schieberegister eingebaut werden kann. Wir verschalten es als selbsteinschwingenden Ringzähler. In der Kategorie “Shift_Register” finden wir passende Typen, in der Kategorie “Logic” das NOR-Gatter mit sieben Eingängen.

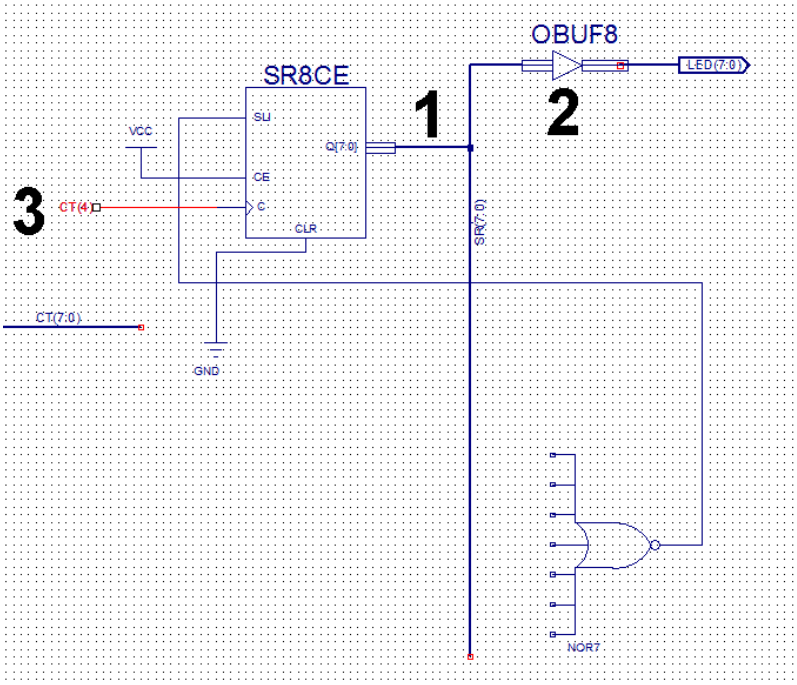


Abb. 67 Die Grundschaltung. 1 - das Schieberegister hat einen Busausgang. Er ist hier schon benannt (als “SR(7:0)”) und mit den LED-Ausgängen verbunden worden. 2- dabei wurde der Ausgangspuffer wieder eingebaut. 3 - auch der Takteingang des Schieberegisters wurde benannt und so an an einen der Zählerausgänge angeschlossen.

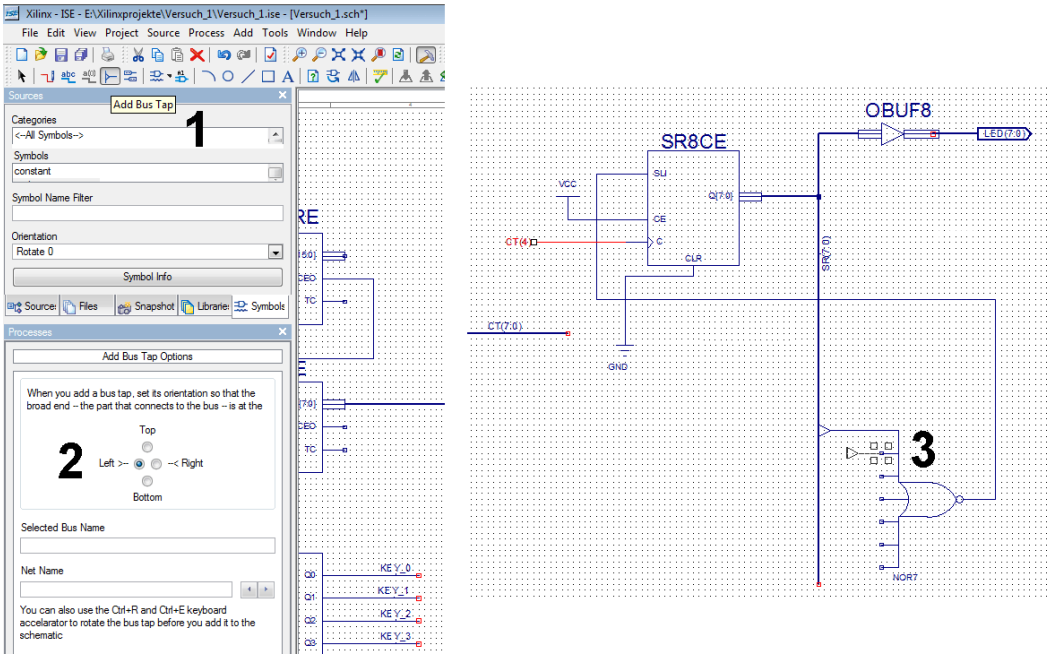


Abb. 68 Jetzt wird der Bus mit dem NOR-Gatter verbunden. Die einzelnen Leitungen werden über Anzapfungen, sog. *Bus Taps*, angeschlossen. 1 - Funktionsauswahl zum Hinzufügen von *Bus Taps*. 2 - hier wird die Orientierung der Anzapfung gewählt. Mitdenken! 3 - so wird ein *Bus Tap* angefügt. Es dockt automatisch an den Bus an.

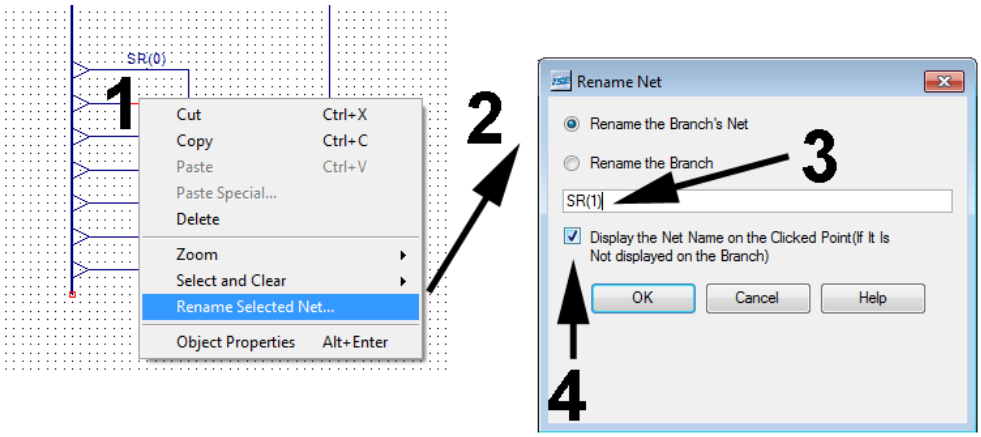


Abb. 69 Nun müssen die *Bus Taps* einzeln benannt werden – eine mühevolle Arbeit; es hilft alles nichts... 1 - Leitung anklicken. Rechte Maustaste. 2 - “Rename Selected Net”. 3 - den Namen eintragen. 4 - dieses Kontrollkästchen aktivieren, damit der Name im Schaltplan auch zu sehen ist. Dann “OK”.

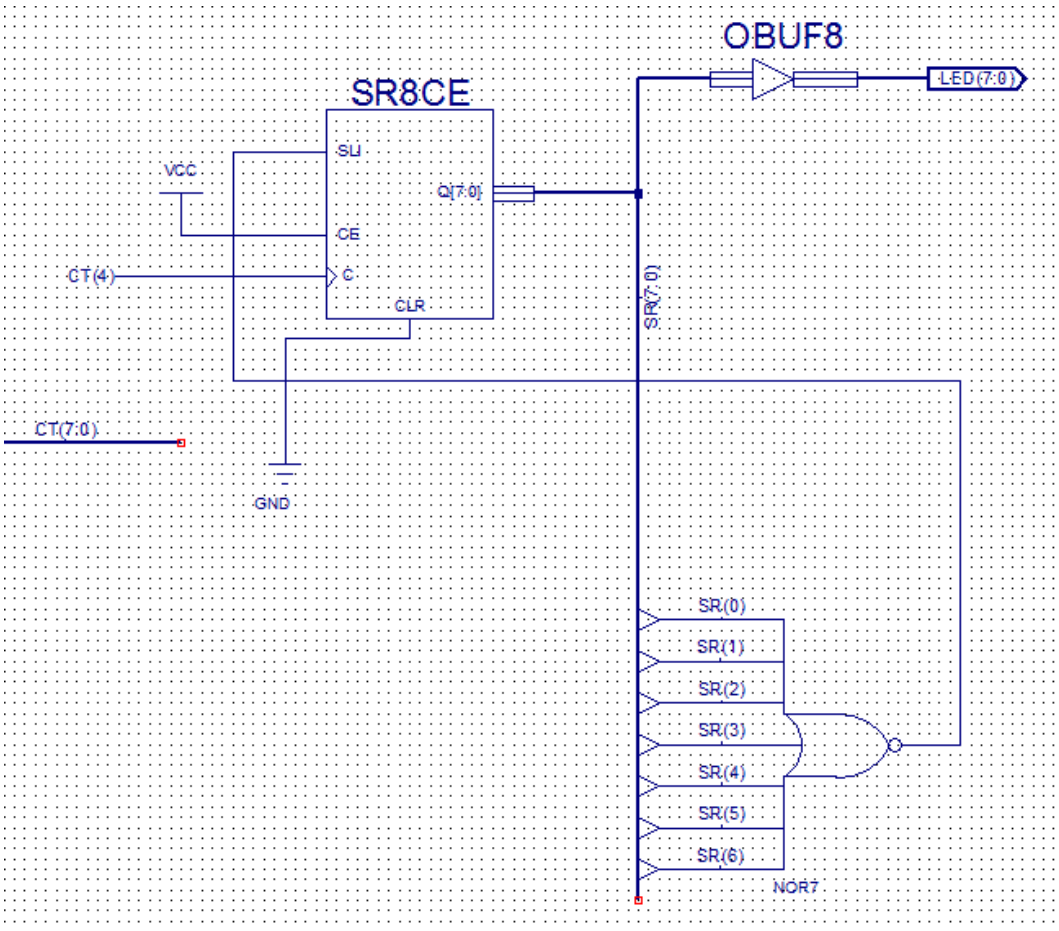


Abb. 70 Die fertige Schaltung. An der Constraints-Datei ist nichts zu ändern. Implementieren, programmieren und zusehen, ob es funktioniert.

Aufgabe 4: Lauflicht in beiden Richtungen

Der Leuchtpunkt soll zyklisch hin und her laufen. Es gibt viele Möglichkeiten, dieses Problem zu lösen. Sie dürfen gern alternative Lösungen ausprobieren. Unser Lösungsvorschlag beginnt damit, einen Zustandsautomaten zu skizzieren, der das Gewünschte leistet.

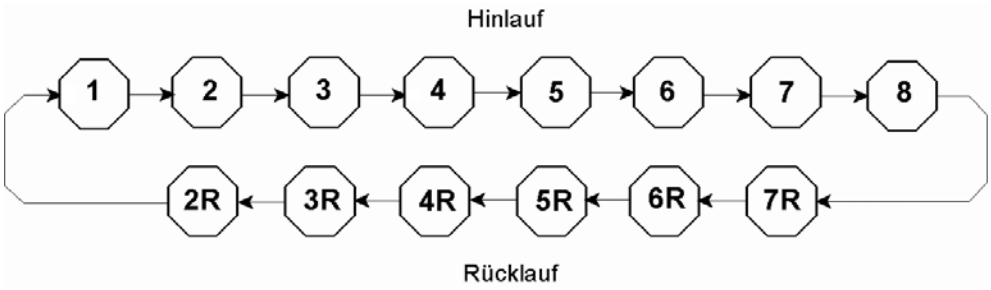


Abb. 71 Es sind insgesamt 14 Zustände. 1...8: die Aktivierung der einzelnen LEDs beim Hinlauf. Wenn die 8. LED leuchtet, wird der Rücklauf ausgelöst. Dann muß wieder die 7. LED leuchten usw. Wenn die 1. LED leuchtet, beginnt wieder der Hinlauf (2. LED. 3. LED usw.). So ergeben sich die 8 + 6 = 14 Zustände.

Die erste und die achte LED sind in jeweils einem einzigen Zustände aktiv, die verbleibenden 6 LEDs sind es in jeweils zwei Zuständen. Wir lösen die Aufgabe, indem wir das Schieberegister entsprechend verlängern.

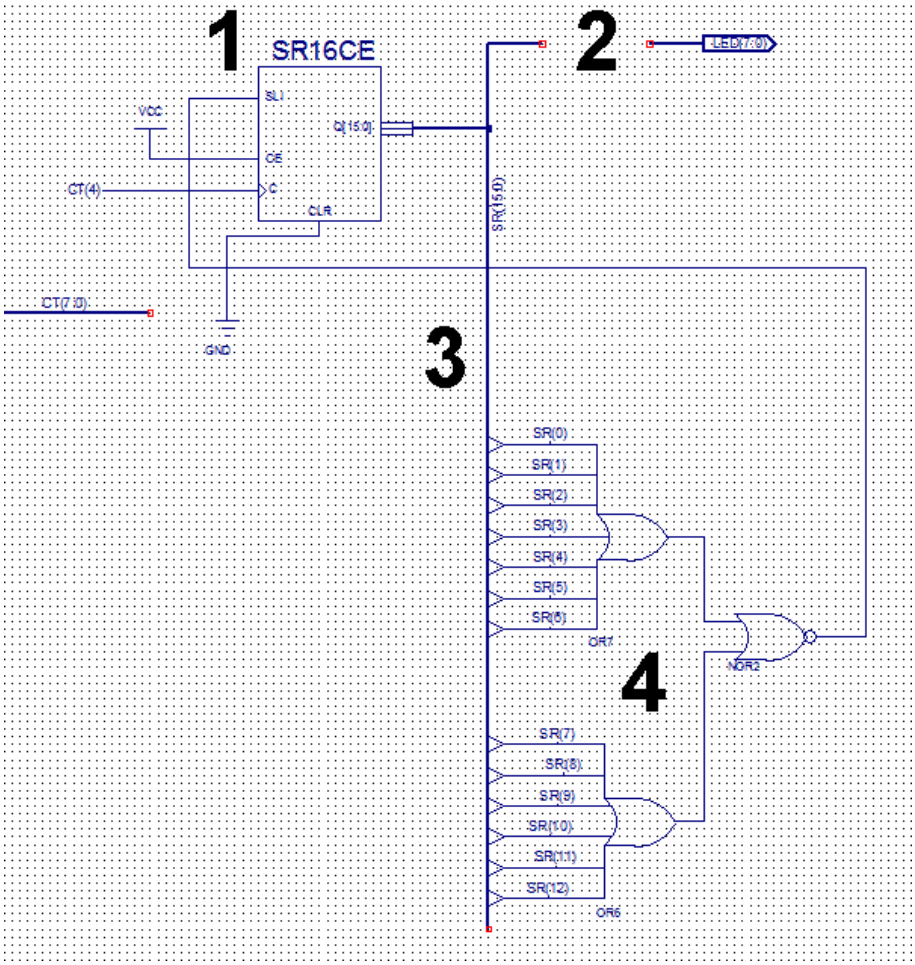


Abb. 72 Der Umbau. 1 - das alte Schieberegister löschen. Den entsprechenden 16-Bit-Typ auswählen und einfügen (dockt an die vorhandenen Anschlüsse an). 2 - den Ausgangspuffer löschen. 3- den alten Bus und das XOR nebst Bus Taps löschen. Neuen Bus zeichnen und entsprechend benennen (SR(15:0). 4 - Das Schieberegister soll einen Umlaufperiode von 14 Takten haben. Also brauchen wir ein NOR mit 13 Eingängen. Das könen wir so aufbauen wie hier gezeigt. Dann die Bus Taps anschließen und benennen.

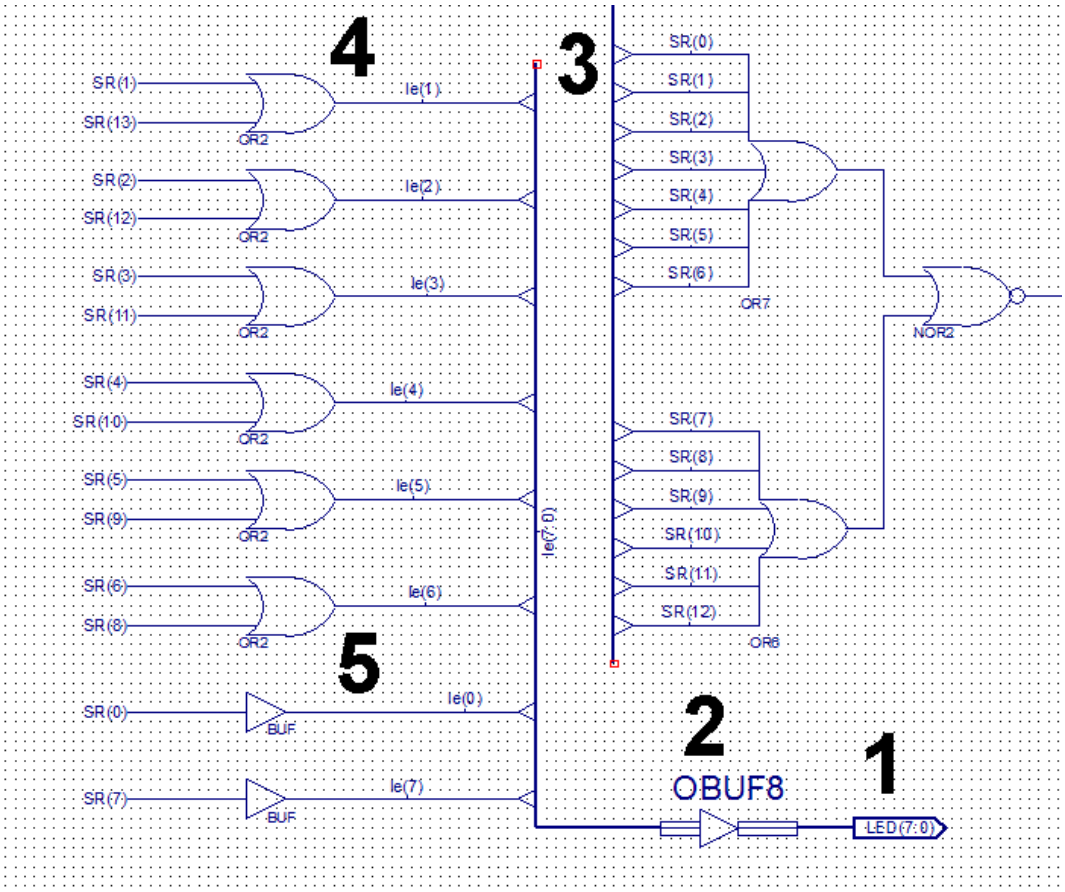


Abb. 73 Wir wollen diese Schaltung bauen. 1 - den Marker der LED-Ausgänge hierher schaffen. 2 - einen Ausgangspuffer vorsetzen. 3 - am Eingang des Ausgangspuffers beginnend einen Bus zeichnen und benennen (le(7:0)). 4 - für die 6 LEDs mit zwei Zuständen brauchen wir 6 ODER-Gatter mit zwei Eingängen. 5 - die erste und die achte LED sind direkt anzuschließen. Das geht aber nicht so ohne weiteres. Deshalb sehen wir zwei Puffer (“BUF”) vor. Dann die Bus Taps einzeichnen und benennen.

Nun sind die Eingänge der Gatter und Puffer an das Schieberegister anzuschließen. Hierzu könnten wir den SR-Bus an den Gattereingängen vorbeiführen und Bus Taps einzeichnen. Das wird aber sehr unübersichtlich. Deshalb schließen wir hier die Leitungen über ihre Namen an.

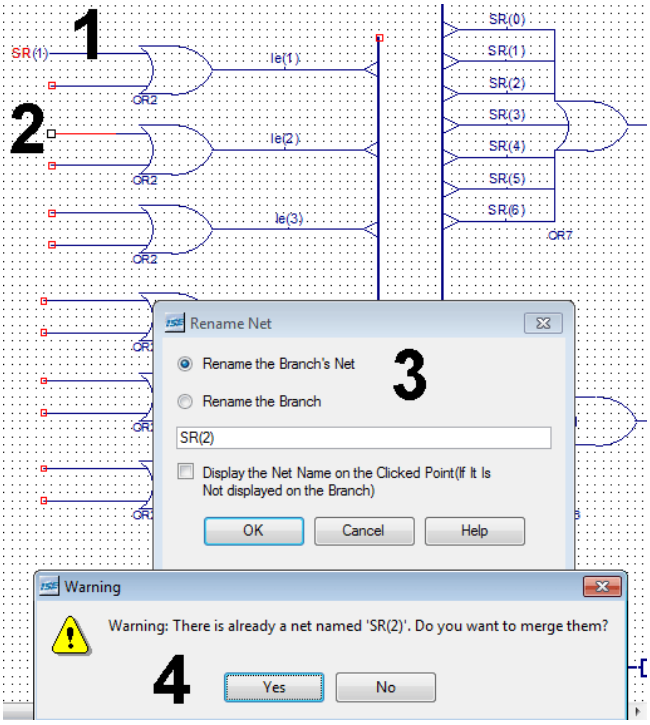


Abb. 74 Anschlüsse benennen. Die Anschlüsse der Funktionselemente lassen sich nicht benennen. 1 - deshalb müssen wir ein kurzes Leitungsstück anfügen. 2 - das Leitungsstück anklicken. Rechte Maustaste. 3 - Namen eintragen. 4 - nach “OK” erscheint diese Nachricht. Es ist kein Fehler, sondern eine Betsätigung, daß wir auf dem richtigen Weg sind – denn das Signal, das wir anschließen wollen, existiert schon. Also “Yes”.

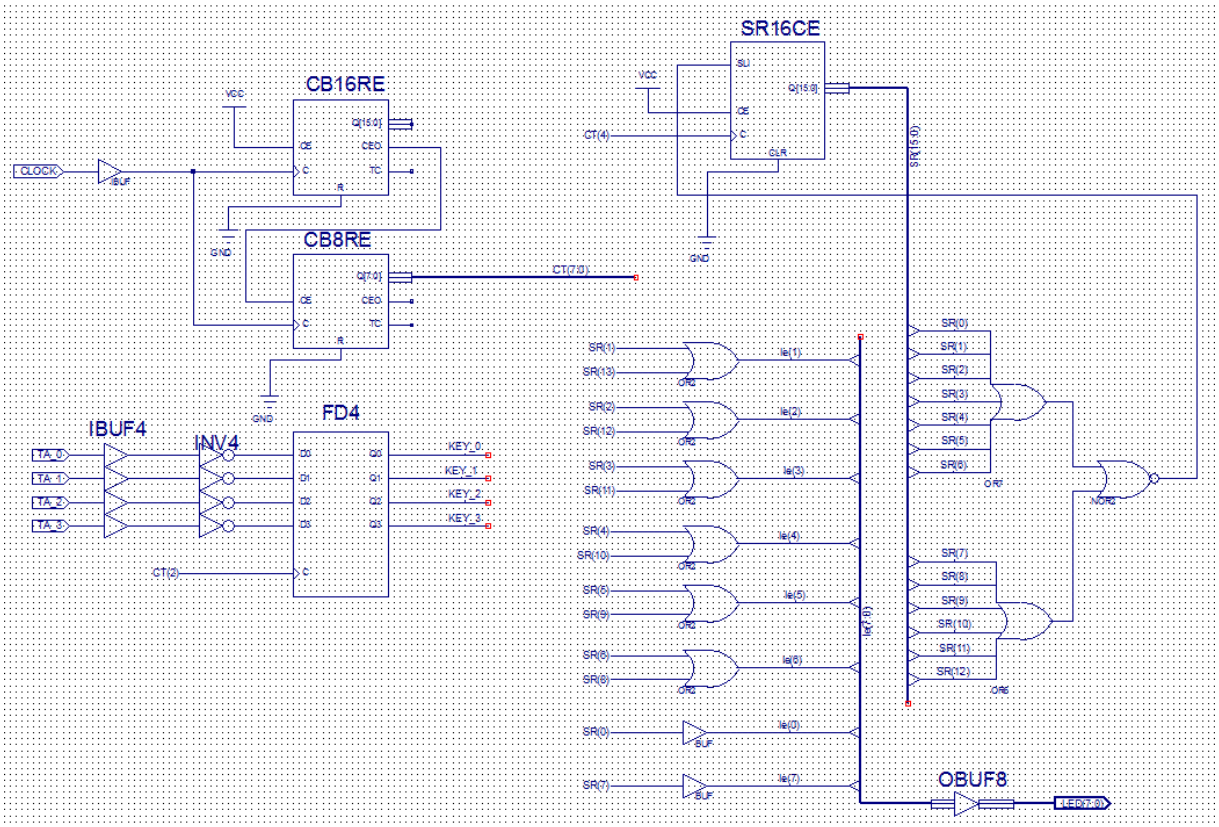


Abb. 75 Die fertige Schaltung. An der Constraints-Datei ist nichts zu ändern. Implementieren, programmieren und zusehen, ob es funktioniert.

Aufgabe 5: Zwei Binärzähler mit Handbetätigung und Anzeige

Im zweiten Versuch wollen wir einen Impulsmustergenerator als Prüfgenerator für serielle Schnittstellen bauen. Hierbei ist ein Prüfzeichen von acht Bits Länge einzugeben. Wir haben aber keine acht Kippschalter. Deshalb ordnen wir zwei Einstellzähler zu vier Bits an, deren Ausgänge auf die LEDs geführt und deren Takteingänge von zwei Tasten angesteuert werden. Der bisherige Schieberegisterentwurf wird hierzu gelöscht. Die LED-Ausgänge bleiben.

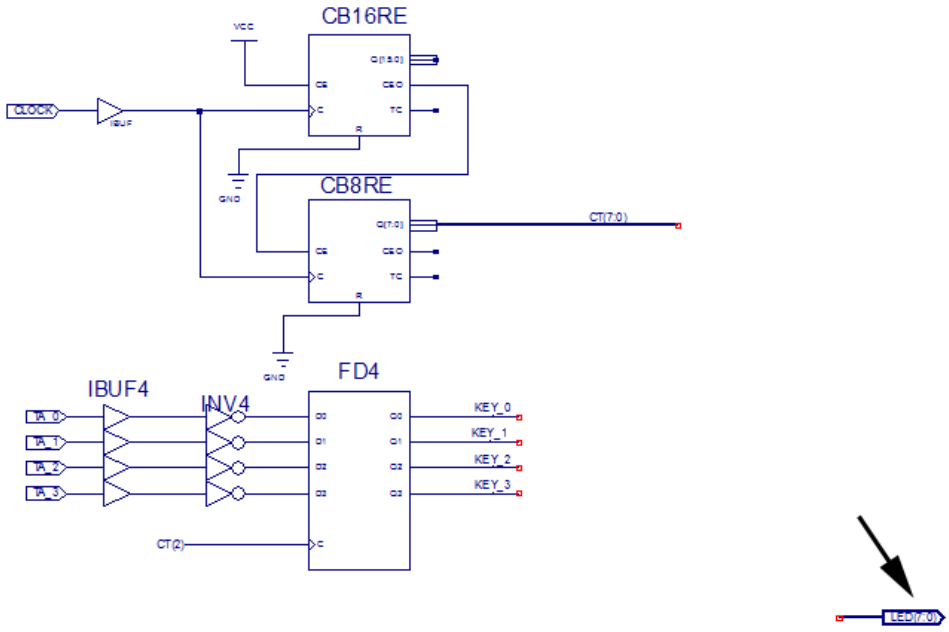


Abb. 76 Die Schieberegisterschaltung wurde gelöscht. Den Output Marker der LEDs aber behalten (Pfeil).

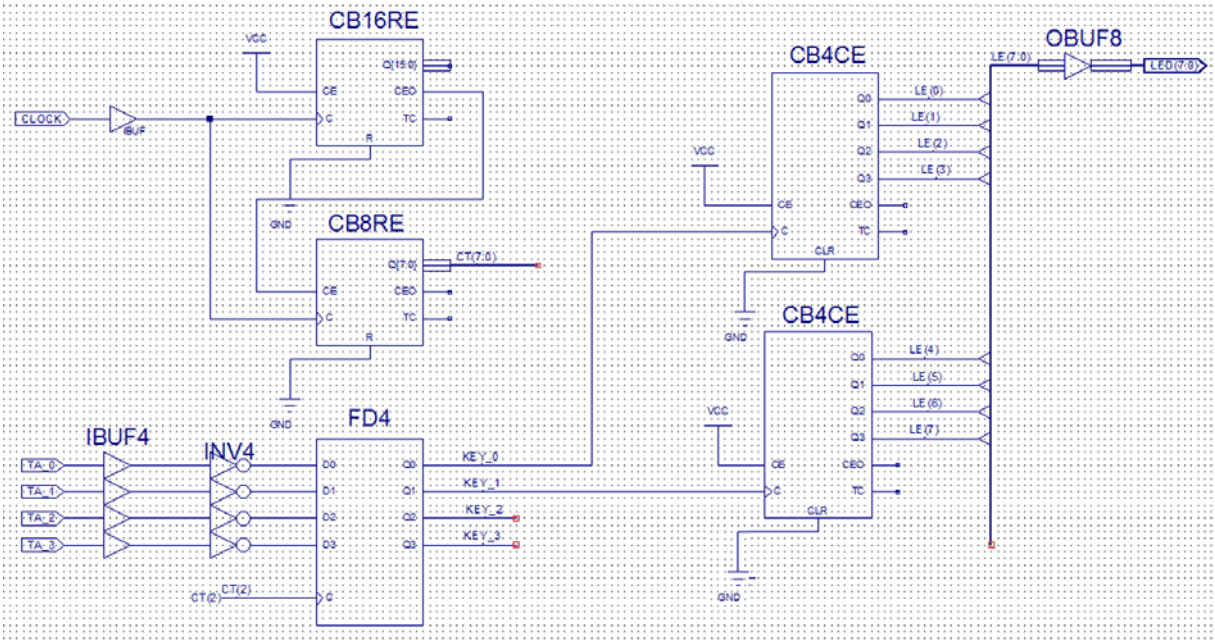


Abb. 77 Die fertige Schaltung. An der Constraints-Datei ist nichts zu ändern. Die Taste 0 bedient die niederwertigen Binärstellen, die Taste 1 die höherwertigen. Ausprobieren...

Aufgabe 6: Siebensegmentanzeige

Jetzt sollen – um den Bedienkomfort zu verbessern – zwei Siebensegmentanzeigen angeschlossen werden. Wir verwenden die Siebensegmentanzeige 09/13 und nutzen zunächst nur zwei Anzeigemodule. Jedes wird mit einem E-A-Port des CPLD-Lehrgeräts 12 verbunden. Segment A entspricht Bitposition 0, Segment B entspricht Bitposition 1 usw. Welche Ports wir verwenden, ist gleichgültig. Wir müssen nur die Constraints-Datei entsprechend ausfüllen.

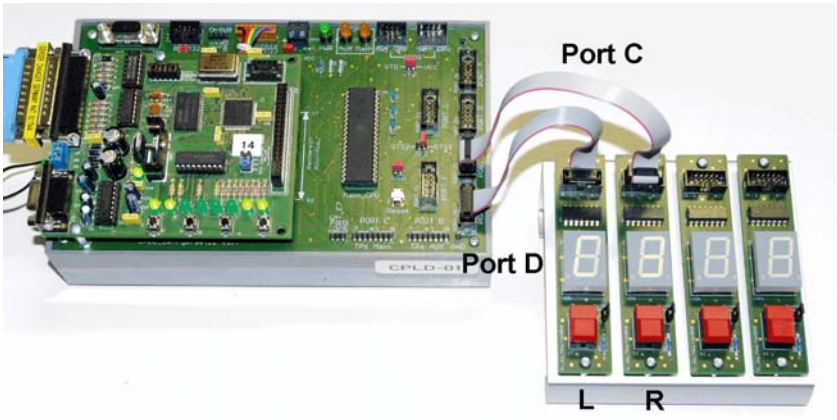


Abb. 78 Anschließen der Siebensegmentanzeige 09/13. Hier die Vorzugsvariante, die den nachfolgenden Abbildungen zugrunde liegt. Nur zwei Anzeigemodule. Das linke (L) an Port C, das rechte (R) an Port D. Die verbleibenden Module können dann später an die Ports A und B angeschlossen werden, ohne daß es einen Kabelsalat gibt.

Die Siebensegmentdecoder sind als Funktionselemente schon vorgefertigt. Sie sind als Kopien externer Quellen dem Projekt hinzuzufügen.

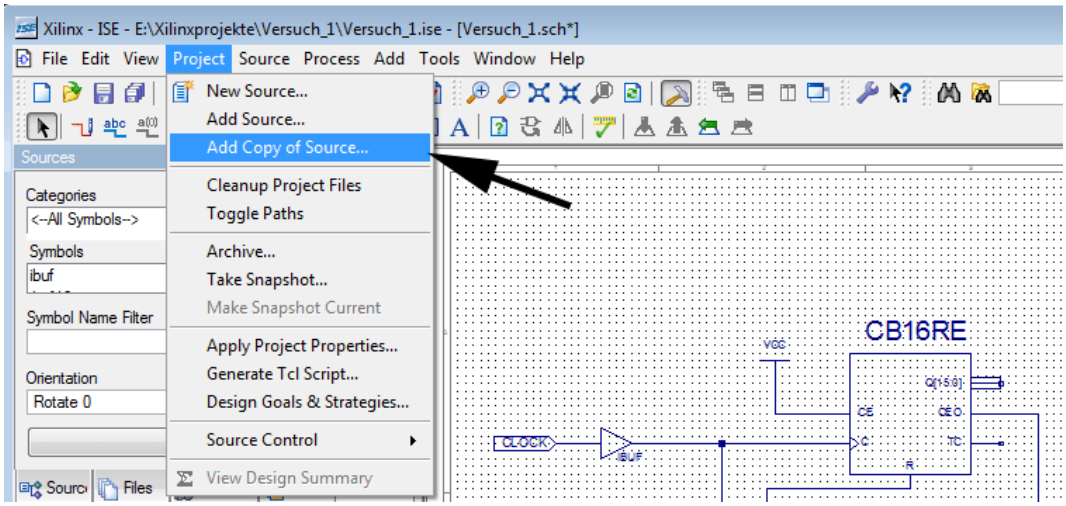


Abb. 79 Die Funktionselementeauswahl läuft über “Project” – “Add Copy of Source”. Die Quellen als Kopien hinzufügen, weil wir die Originale behalten wollen.

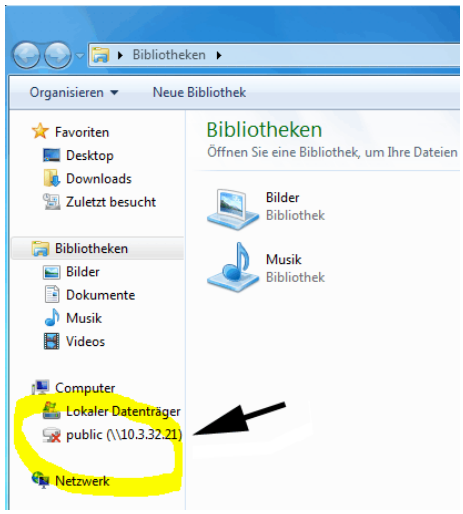


Abb. 80 Die Daten befinden sich auf der Netzfestplatte. Auswählen.

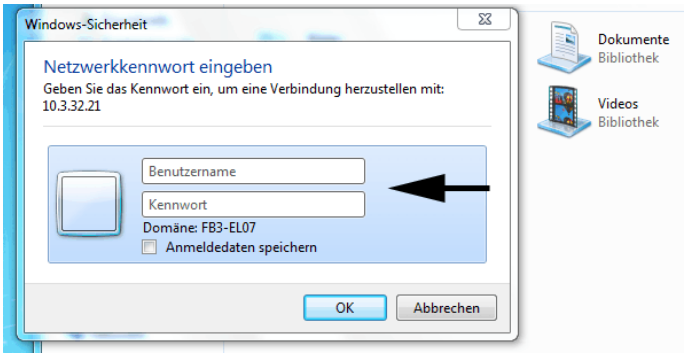


Abb. 81 Benutzername und Kennwort eingeben.

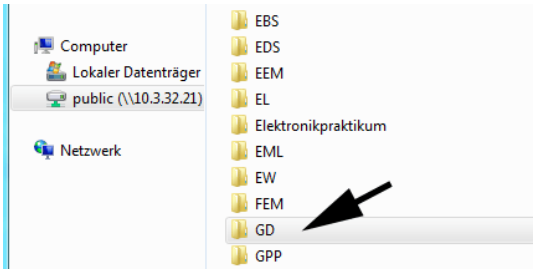


Abb. 82 Verzeichnis auswählen.

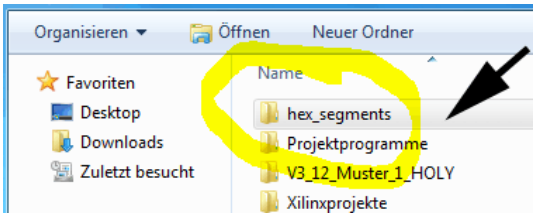


Abb. 83 Unterverzeichnis auswählen.

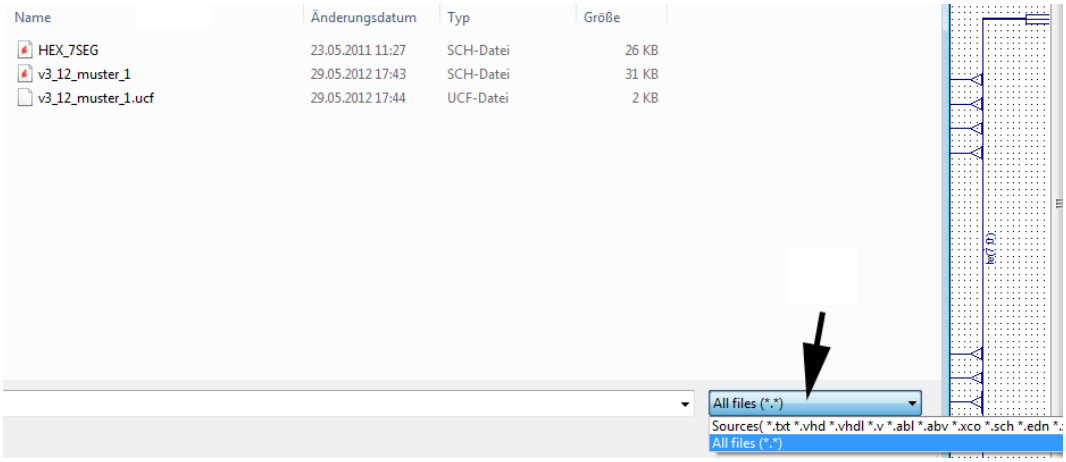


Abb. 84 Wir brauchen den Schaltplan (.sch) und das Schaltsymbol (.sym). Damit wir diese Datentypen zu sehen bekommen, müssen wir die Datentypauswahl auf “All files” umschalten.

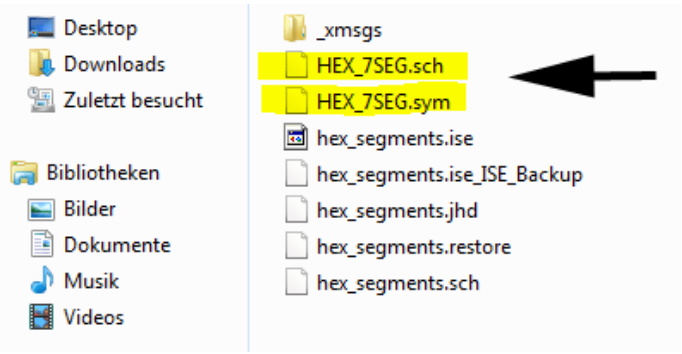


Abb. 85 Dateiauswahl. Hinweis: Beide Dateien müssen den gleichen Namen haben, sonst funktioniert es später nicht (hier: HEX_7SEG.sch (Schaltplan) und HEX_7SEG.sym (Schaltsymbol)).

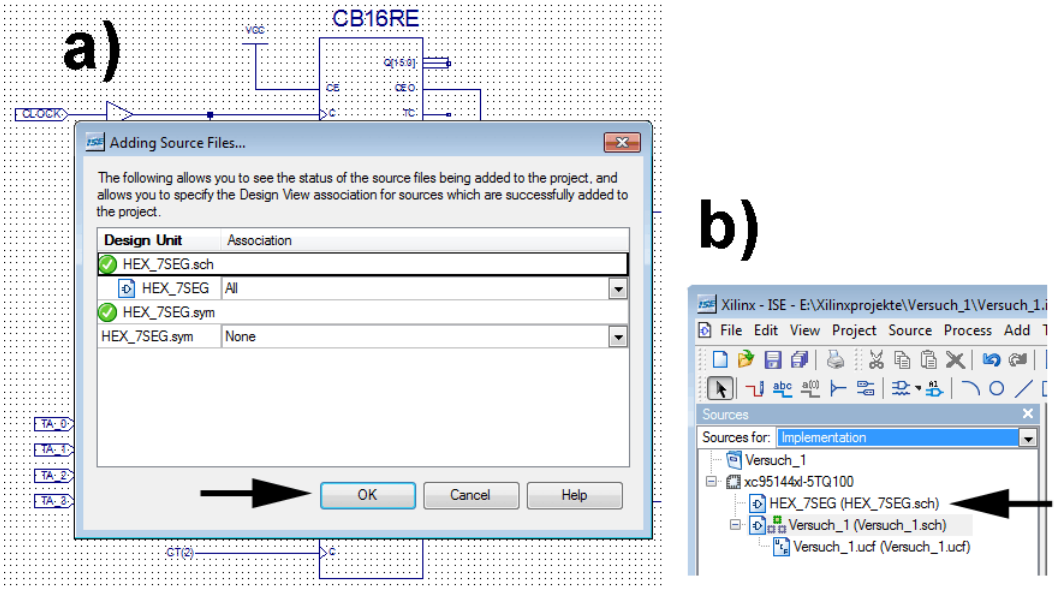


Abb. 86 Die Quellen werden hinzugefügt. a) Auswahlanzeige. b) der hinzugefügte Schaltplan sollte jetzt im Verzeichnis der Quellen erscheinen.

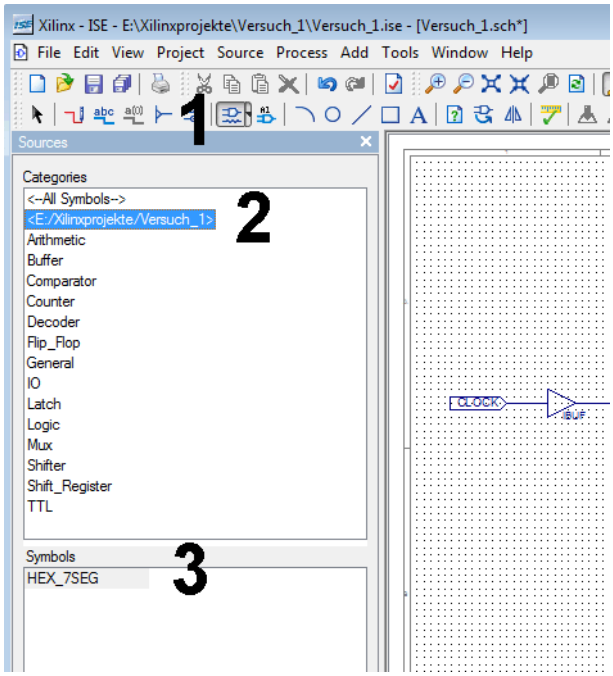


Abb. 87 Jetzt muß das Funktionselement zur Auswahl bereitstehen. 1 - Funktionselementeauswahl; 2 - die neue Kategorie der zum Projekt hinzugefügten Funktionselemente; 3 - das Funktionselement Siebensegmentdecoder.

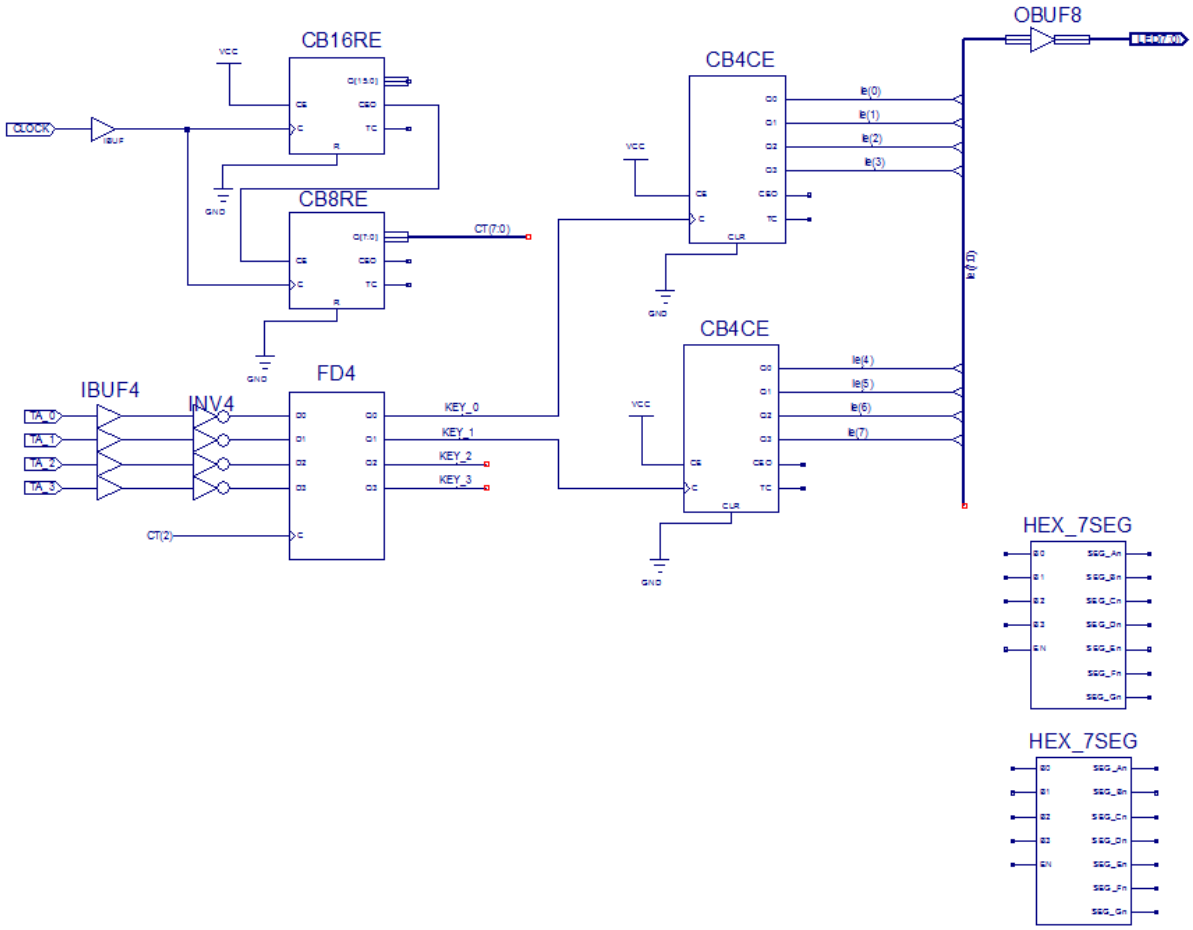


Abb. 88 Etwas Platz schaffen (die vorhandene Schaltung nach oben schieben) und zwei Siebensegmentdecoder hinzufügen.

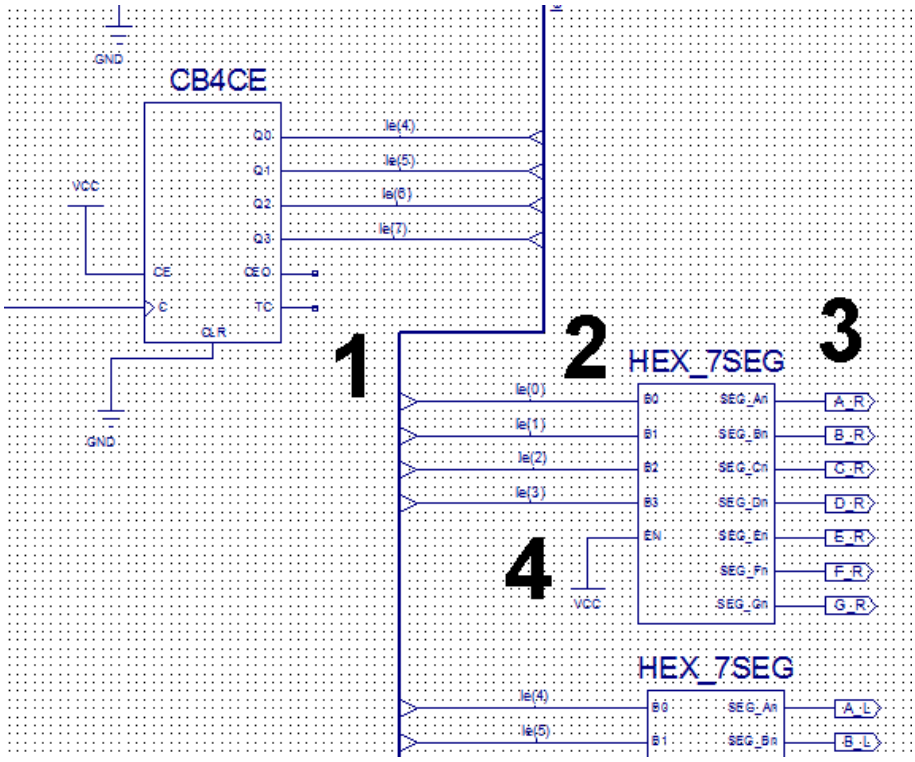


Abb. 89 Die Funktionselemente in die Schaltung einbauen. 1 - hierzu den Bus der Zählerausgänge so verlängern, daß es nach etwas aussieht. 2 - über Bus Taps mit den Siebensegmentdecodern verbinden. 3 - die Ausgangspuffer sind schon eingebaut. Es genügt, Output Marker anzufügen. 4 - der Erlaubniseingang braucht einen High-Pegel.

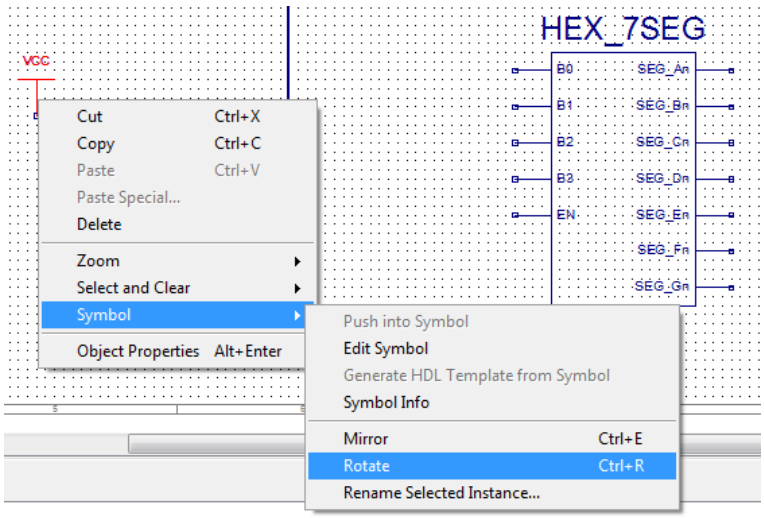


Abb. 90 Wie man ein Symbol umdreht (vgl. Position 4 (VCC) in der Abbildung oben): 1. Symbol auswählen. 2. Rechte Maustaste. 3. “Symbol”. 4. “Rotate”. Ggf. mehrmals rotieren.

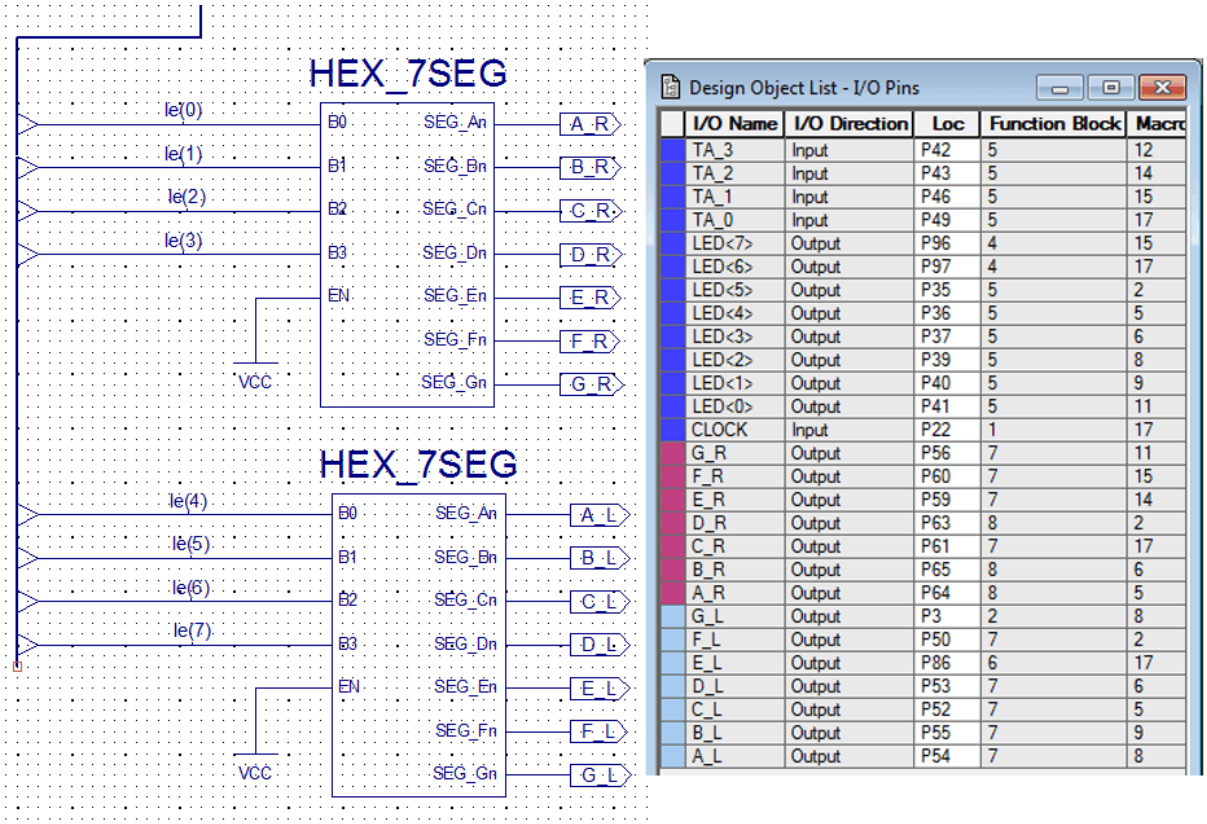


Abb. 91 Die fertige Siebensegmenterweiterung. R = rechte, L = linke Anzeige. Constraints-Datei gemäß Anschlußtabelle ergänzen. Ausprobieren...

Aufgabe 7: Die Tasten der Anzeigemodule ausnutzen

Es liegt nahe, die Tasten unter den Siebensegmentanzeigen zum Zählen auszunutzen. Hierzu brauchen wir zwei weitere Entprellflipflops.

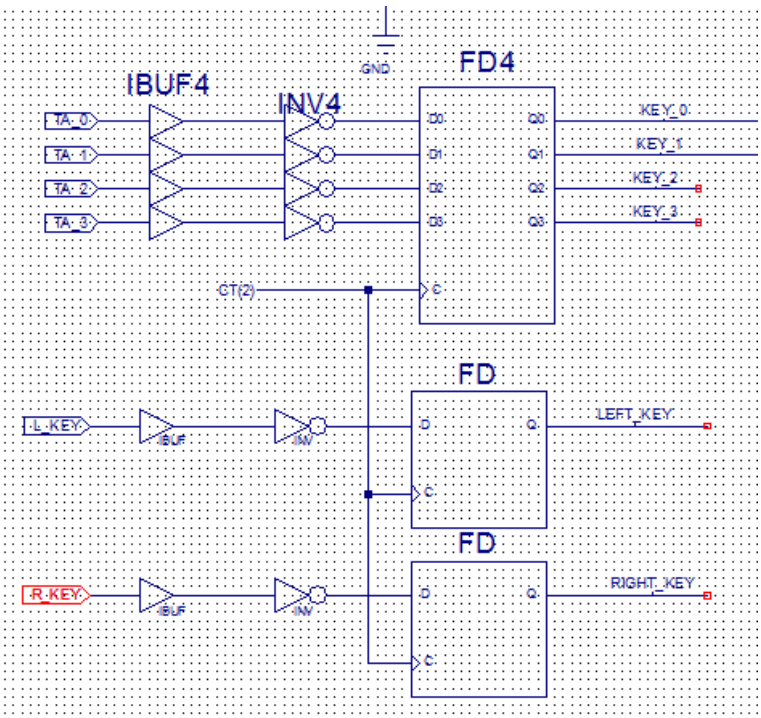


Abb. 92 Die Schaltung um zwei Entprellflipflops erweitern.

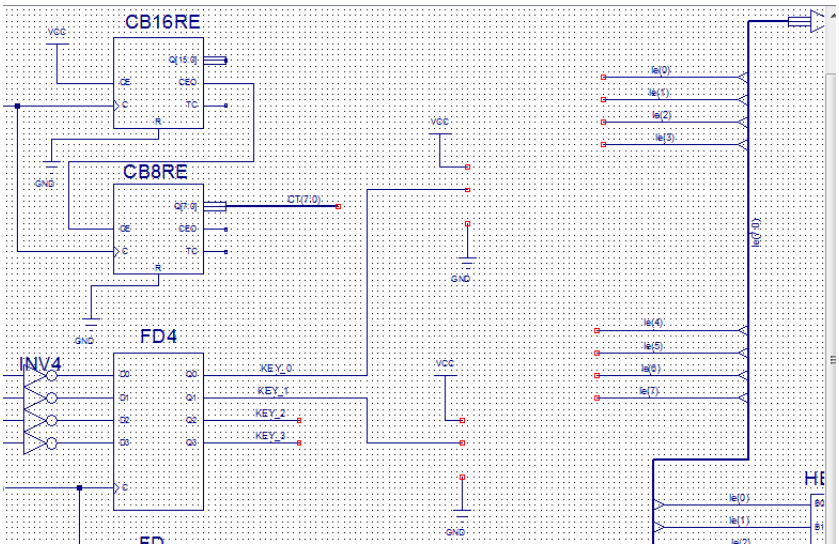


Abb. 93 Nun müssen die Ausgangssignale der Entprellflipflops an die Takteingänge der Zähler angeschlossen werden. Sie sind aber bereits mit anderen Quellen verbunden. Wir wollen diese Signale (KEY0, KEY1) jedoch nicht löschen, sondern beibehalten. Deshalb folgender Trick: 1. die Zähler löschen. 2. Die Signale von ihren Anschlußpositionen gleichsam wegschubsen (Geduldsspiel mit der Maus; dabei linke Taste gedrückt halten). 3. Die Zähler wieder einfügen. Alle Anschlüsse docken wieder an, nur die Takteingänge nicht. Alternativ dazu könnte man die Signale auch löschen, später neu zeichnen und benennen.

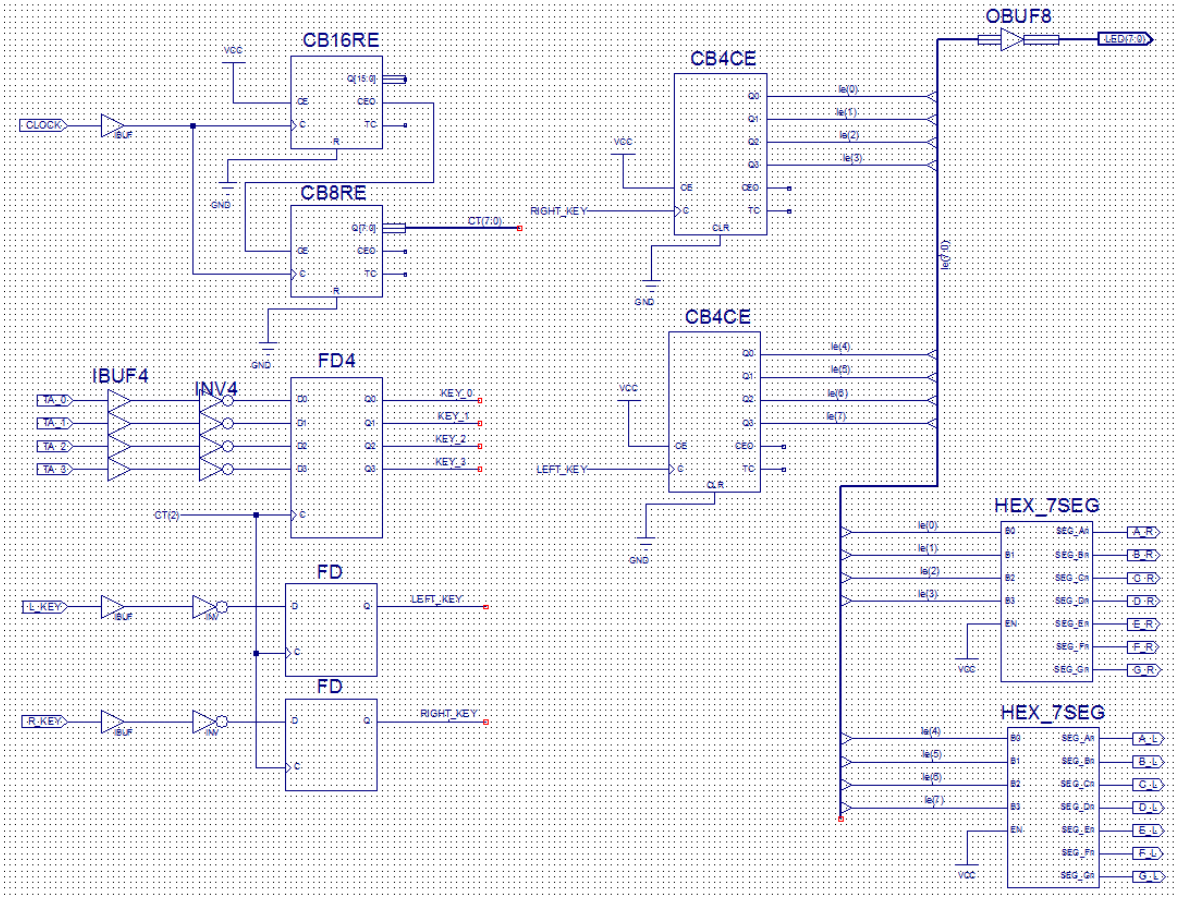
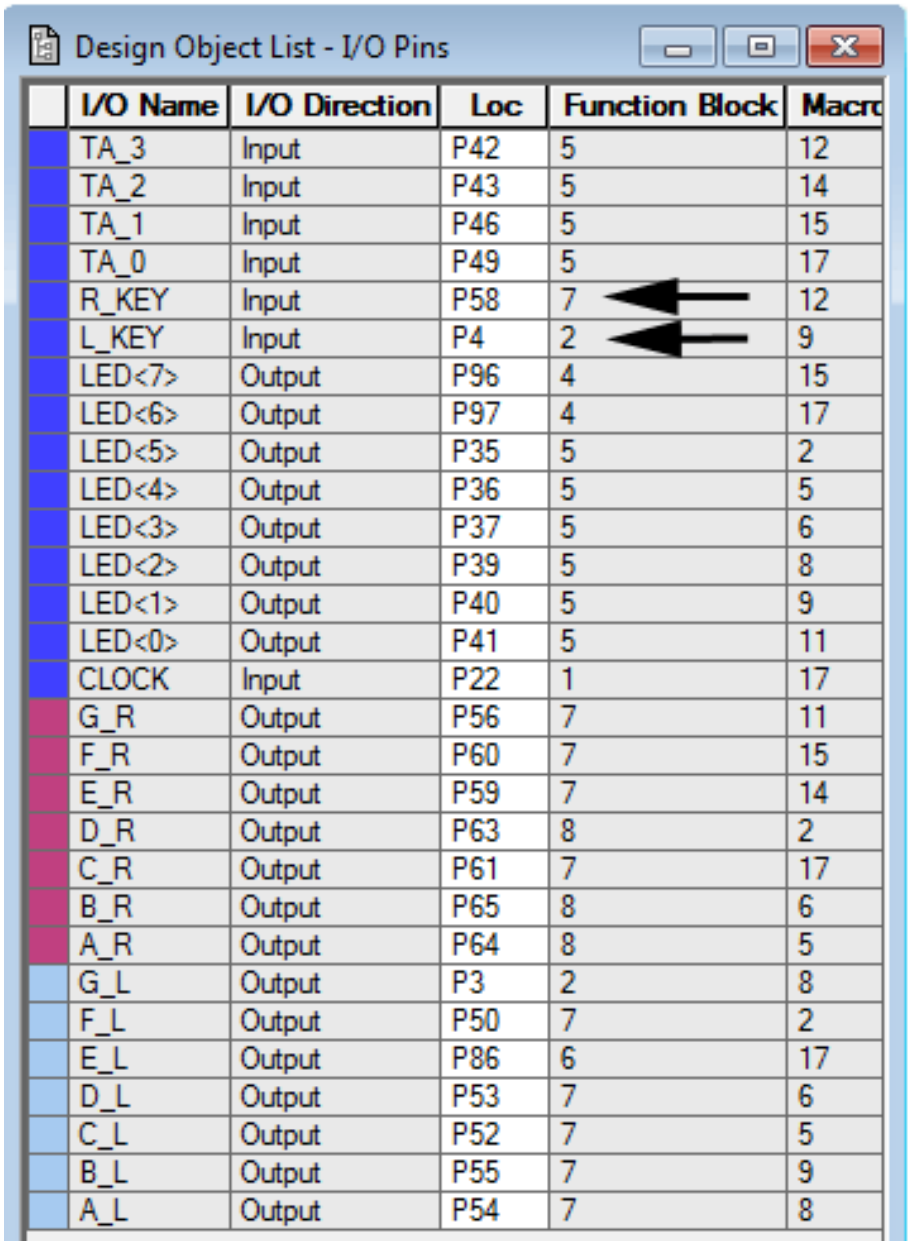


Abb. 94 Die fertige Schaltung. Die Zählertakte werden nicht direkt, sondern über ihre Namen angeschlossen. Dann kann man später die Verbindungen leichter ändern.



Design Object List - I/O Pins

I/O Name	I/O Direction	Loc	Function Block	Macro
TA_3	Input	P42	5	12
TA_2	Input	P43	5	14
TA_1	Input	P46	5	15
TA_0	Input	P49	5	17
R_KEY	Input	P58	7	12
L_KEY	Input	P4	2	9
LED<7>	Output	P96	4	15
LED<6>	Output	P97	4	17
LED<5>	Output	P35	5	2
LED<4>	Output	P36	5	5
LED<3>	Output	P37	5	6
LED<2>	Output	P39	5	8
LED<1>	Output	P40	5	9
LED<0>	Output	P41	5	11
CLOCK	Input	P22	1	17
G_R	Output	P56	7	11
F_R	Output	P60	7	15
E_R	Output	P59	7	14
D_R	Output	P63	8	2
C_R	Output	P61	7	17
B_R	Output	P65	8	6
A_R	Output	P64	8	5
G_L	Output	P3	2	8
F_L	Output	P50	7	2
E_L	Output	P86	6	17
D_L	Output	P53	7	6
C_L	Output	P52	7	5
B_L	Output	P55	7	9
A_L	Output	P54	7	8

Abb. 95 Nun nur noch die zusätzlichen Tastensignale in die Constraints-Datei eintragen...

Die Anschlüsse der Bedien- und Anzeigeelemente:

Anschluß	Pin
Taste 0 (KEY0#)	49
Taste 1 (KEY1#)	46
Taste 2 (KEY2#)	43
Taste 3 (KEY3#)	42
LED 0	41
LED 1	40
LED 2	39
LED 3	37
LED 4	36
LED 5	35
LED 6	97
LED 7	96
Quarztakt 16 MHz (GCK1)	22
Das zweite Taktsignal (GCK2)	23
Gesamtrücksetzen (GSR)	99
Serielle Schnittstelle, Sendedaten (TXD)*	94
Serielle Schnittstelle, Empfangsdaten (RXD)*	92

*: Benennung (gemäß Originalschaltplan) aus Sicht der angeschlossenen Datenendeinrichtung.
Die Sendedaten vom CPLD gehen an Pin 92, die Empfangsdaten zum CPLD kommen von Pin 94.

Siebensegmentanzeige an Port A

Anschluß	Pin
Segment A	82
Segment B	85
Segment C	80
Segment D	81
Segment E	78
Segment F	79
Segment G	76
Taste oder Dezimalpunkt	77

Siebensegmentanzeige an Port B

Anschluß	**Pin**
Segment A	73
Segment B	74
Segment C	71
Segment D	72
Segment E	68
Segment F	70
Segment G	66
Taste oder Dezimalpunkt	67

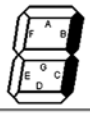
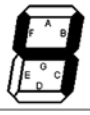
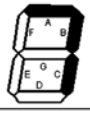
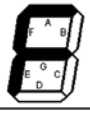
Siebensegmentanzeige an Port C

Anschluß	**Pin**
Segment A	64
Segment B	65
Segment C	61
Segment D	63
Segment E	59
Segment F	60
Segment G	56
Taste oder Dezimalpunkt	58

Siebensegmentanzeige an Port D

Anschluß	**Pin**
Segment A	54
Segment B	55
Segment C	52
Segment D	53
Segment E	86
Segment F	50
Segment G	3
Taste oder Dezimalpunkt	4

Die hexadezimale Siebensegmentanzeige:

d	c	b	a	G	F	E	D	C	B	A	
0	0	0	0	0	1	1	1	1	1	1	
0	0	0	1	0	0	0	0	1	1	0	
0	0	1	0	1	0	1	1	0	1	1	
0	0	1	1	1	0	0	1	1	1	1	
0	1	0	0	1	1	0	0	1	1	0	
0	1	0	1	1	1	0	1	1	0	1	
0	1	1	0	1	1	1	1	1	0	1	
0	1	1	1	0	0	0	0	1	1	1	
1	0	0	0	1	1	1	1	1	1	1	
1	0	0	1	1	1	0	1	1	1	1	
1	0	1	0	1	1	1	0	1	1	1	
1	0	1	1	1	1	1	1	1	0	0	
1	1	0	0	0	1	1	1	0	0	1	
1	1	0	1	1	0	1	1	1	1	0	
1	1	1	0	1	1	1	1	0	0	1	
1	1	1	1	1	1	1	0	0	0	1	

Die Innenschaltung des Siebensegmentdecoders

Die Segmentsignale wirken aktiv Low.