

Aufgabe 5: ChipScope und andere integrierte Testhilfen

G. Kemnitz

19. Juli 2007

Beim Hardwareentwurf entstehen wie bei anderen Programmierarbeiten Fehler. Diese müssen gefunden und beseitigt werden. Fehlersuche und Beseitigung kosten üblicherweise 50% bis 80% der Gesamtentwurfszeit. In den vergangenen Aufgaben wurden folgende Möglichkeiten zum Test und zur Fehlersuche genutzt:

- Syntax- und Plausibilitätstest: Kontrolle, dass das Programm "ein Wort der Sprache VHDL ist" und das die einzelnen Projektdateien zusammenpassen.
- Simulation: Kontrolle der Ausgaben der Schaltungsbeschreibung für ausgewählte Eingaben.
- Test: Bereitstellung der Eingaben über Schalter und Kontrolle der Ausgaben der fertig programmierten Schaltung an Leuchtdioden.

Mit den Leuchtdioden, Tastern und Schaltern lassen sich nur langsame Signaländerungen vorgeben und beobachten. Für die Beobachtung schneller Signaländerungen benötigt man einen Logikanalysator. Ein Logikanalysator ist ein Messgerät zur Aufzeichnung und Visualisierung von Bitvektoren. Im Modus Timing-Analyse werden die Daten im festen Zeitraster aufgezeichnet (Auflösung typisch 1 ns bis 1 ms). Im State-Modus werden die Aufzeichnungszeitpunkte durch Datenflanken der auszuwertenden Daten bestimmt. Die Eingabedaten, für die die Ausgaben aufgezeichnet werden, müssen auf andere Weise bereitgestellt werden, typisch aus der Anwendungsumgebung oder von integrierten Testmustergeneratoren.

ChipScope: Der Anschluss einen Logikanalysators an ein Prüfobjekt ist zum Teil problematisch (Herausführung aller zu beobachtenden Signale aus den Schaltkreisen, zunehmend kleinere Pinabstände an den Schaltkreisen, Mehrlagenleiterplatten etc.). ChipScope ist eine Technik speziell für hochintegrierte programmierbare Logikschaltkreise mit internen Blockspeichern, bei dem die Mitschreiblogik und der Speicher des Logikanalysators mit in den Schaltkreis programmiert wird. Die mitgeschriebenen Signalverläufe können vom Arbeitsplatzrechner über einen Testbus ausgelesen, dargestellt und weiterverarbeitet werden.

1 Versuchsaufbau

Prüfobjekt ist ein autonomer Automat (Automat der nur einen Takt- und einem Initialisierungseingang) der zyklisch mit einer Taktperiode von 20 ns ($f_{CLK} = 50$ MHz) eine Zustandsfolge durchläuft. Die Ausgänge sind auf Leuchtdioden geführt.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Aufgabe4 is
port(CLK: in std_logic;
      btn_2FT: std_logic;
      sw      : in std_logic_vector(1 downto 1);
      led     : out std_logic_vector(7 downto 0));
```

```

end Aufgabe4;

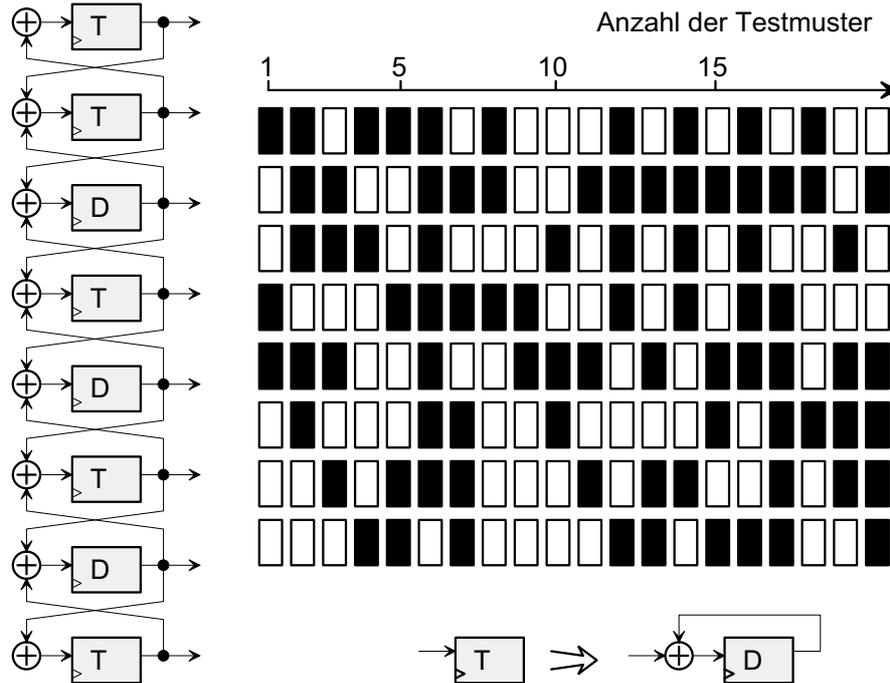
architecture a of aufgabe4 is
  constant Init: std_logic_vector(7 downto 0):=x"19";
  signal state: std_logic_vector(7 downto 0):=Init;
begin
  process(CLK, btn_2FT) -- Zellenautomat
  begin
    if btn_2FT='1' then
      state<=Init;
    if CLK'event and CLK='1' and SW(1)='1' then
      state<= (state(6 downto 0) & '0') xor
              ('0' & state(7 downto 1)) xor
              (state and "10101011");
      led<=state;
    end if;
  end process;
end a;

```

Es handelt sich hier um einen sog. Zellenautomaten, d.h. einen Automaten, in dem der Folgezustand einer Speicherzelle nur eine Funktion des Ist-Zustandes der Zelle und der Nachbarzellen ist:

$$\begin{aligned}
\text{state}(0) &= \text{state}(0) \oplus \text{state}(1) \\
\text{state}(1) &= \text{state}(0) \oplus \text{state}(1) \oplus \text{state}(2) \\
&\dots
\end{aligned}$$

Diese Schaltung wird als Testmuster-generator-Zufallsgenerator für integriertes Selbsttestfunktionen benutzt. Die nachfolgende Abbildung zeigt den Signalfluss und die ersten 20 Ausgabevektoren nach der Initialisierung:



Am Beispiel dieser Schaltung sollen folgende Aufgaben gelöst werden:

1. Statische Funktionskontrolle mit niedrigem Takt
2. Test mit ChipScope in voller Geschwindigkeit
3. Zusatzaufgabe: Periodenmessung mit einer integrierten Testhilfe

2 Projekt vorbereiten

Legen Sie ein neues Verzeichnis

H:\TGP\Aufgabe4

an und kopieren Sie die Design-Files aus dem Netz in dieses Verzeichnis:

- Aufgabe4.npl Projektdatei
- Aufgabe4.vhd Entity für das Projekt und Gerüst für die Funktionsbeschreibung
- Praktikum.ucf Constraints-Datei der Versuchsbaugruppe, identisch mit der ucf-Datei aus Aufgabe 1

Anschließend öffnen Sie das Projekt ("File", "Open Project", "H:\TGP\Aufgabe4\", "Aufgabe4").

3 Statische Funktionskontrolle

Übernehmen Sie aus Aufgabe3.vhd die Taktteilerfunktion zur Erzeugung des CLK1Hz. Verwenden Sie diesen Takt für den Zellenautomaten. Übersetzen und Download. Kontrollieren Sie, dass die Ausgaben für die ersten 20 Takte mit denen aus der Abbildung übereinstimmen. Anschließend Takt für Zellenautomaten auf 50 MHz zurückändern.

4 ChipScope mit ChipScope Inserter einbauen

- Erzeugen Sie mit "Create New Source" eine neue Source-Datei.
 - Typ: "ChipScope Definition...". Name: cs. Weiter
 - Zuordnung zum Source aufgabe4. Weiter.
 - Fertigstellen.

Konfigurieren des "integrierten Logikanalysators" und Anschluss an die Schaltung

- Doppelclick auf neu erzeugte Datei im Projektbaum öffnet den "Core Inserter".
- Zweimal auf Next, bis zum Menue "ILA" (ILA -"integrierter Logikanalysator").
- Reiter "Trigger Parameter": Die Triggerbedingung eines Logikanalysators beschreibt, bei welchen Werten an den Triggereingängen die Aufzeichnung beginnt. In der Übung soll die Aufzeichnung bei einem bestimmten 8-Bit-Wert auf dem Bus state beginnen. Einstellung wie in folgender Abb. belassen.

ILA Select Integrated Logic Analyzer Options

Trigger Parameters Capture Parameters Net Connections

Trigger Input and Match Unit Settings

Number of Input Trigger Ports: 1 Number of Match Units Used: 1

TRIG0: Trigger Width: 8 Match Type: Basic

 # Match Units: 1 Bit Values: 0, 1, X

 Counter Width: Disabled Functions: =, <>

Trigger Condition Settings

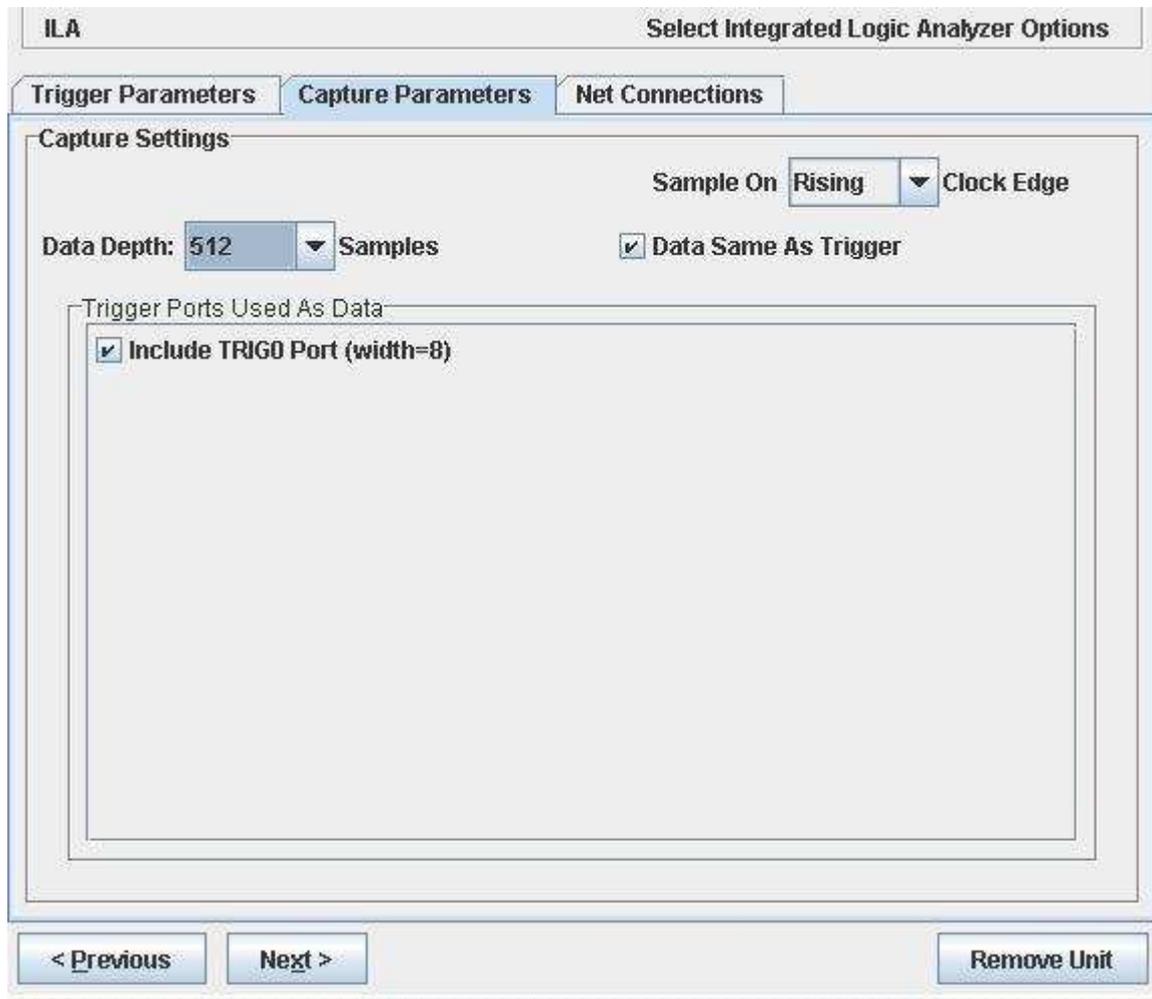
Enable Trigger Sequencer Max Number of Sequencer Levels: 16

Storage Qualification Condition Settings

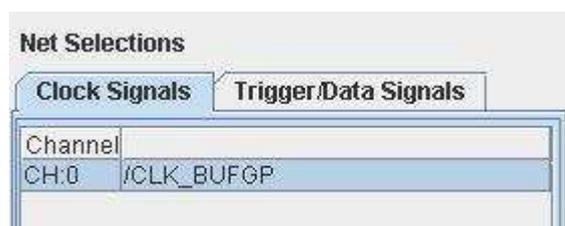
Enable Storage Qualification

< Previous
Next >
Remove Unit

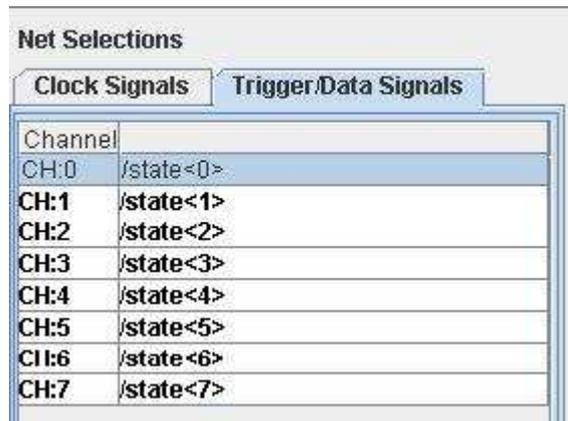
- Reiter "Capture Parameter": Es sollen 512 aufeinanderfolgende Bytes vom Bus state jeweils mit steigender Taktflanke aufgezeichnet werden. Es muss nur "Data Depth" auf 512 erhöht und "Data Same as Trigger" ankreuzt werden.



- Reiter "Net Connection" auf "Modify Connections" klicken. Hier erscheinen auch Signalnamen, die das System bei der Synthese automatisch vergibt. Sortieren Sie nach "Net Name", indem Sie auf die Spaltenüberschrift klicken.
- Stellen Sie sicher, daß sie rechts im Fenster "Net Selections" auf "Clock Signals" sind und dort "CH:0" markiert ist. Wählen Sie unten links das Net "CLK_BUFGP" aus und klicken Sie unten Rechts auf "Make Connections".



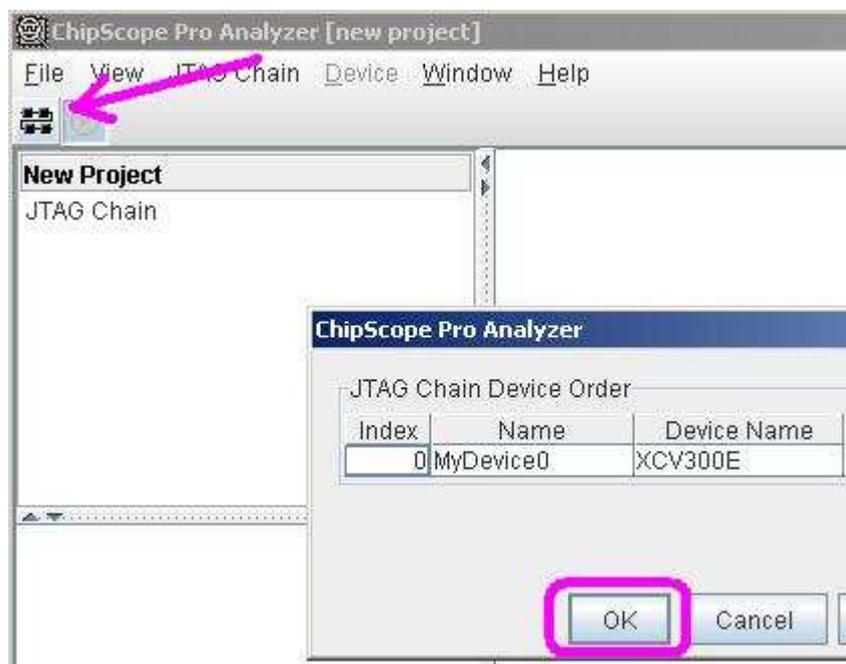
- Wechseln Sie oben Rechts auf "Trigger/Data Signals" und verbinden Sie für alle 8 Eingänge des "integrierten Logikanalysators CH*i* mit state<*i*>.



Die "Net Connections" im ILA-Fenster sollten von Rot auf Schwarz gewechselt haben. Klicken Sie auf "Return to Project Navigator" und speichern Sie die Datei.

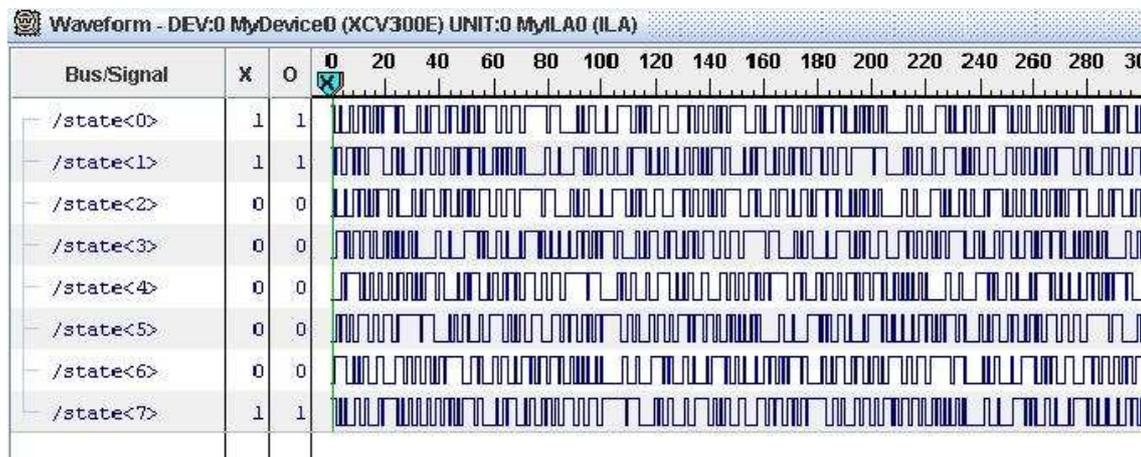
5 Test mit ChipScope vorbereiten

- Aufgabe4 im Source-Fenster auswählen und Doppel-Click auf "Generate Programming File". Das dauert wesentlich länger als in den bisherigen Beispielen, weil die Schaltung mit "integriertem Logikanalysator" viel größer ist.
- Schliessen des iMPACT-Fensters und Doppel-Click auf "Analyze Design Using ChipScope".
- Linkss-Clicken Sie auf das JTAG-Kettensymbol und "JTAG Chain Device Order" und bestätigen.



- Rechts-Click auf "Device 0, XCV300E". Das ist der zu programmierende Schaltkreis. "Configure" und "Device 0". "Select New File", dann die Bit-Datei "aufgabe4.bit" im richtigen Verzeichnis angeben und OK.

- Nachdem erfolgreichem Download abgeschlossen ist, sollte unter anderem ein Waveform-Fenster erscheinen.
- Zur übersichtlicheren Darstellung der Signalnamen Namen: "File"->"Import", dann "Select New File" und Doppel-Click auf "cs.cdc". Dann OK.
- Oben Links in der Toolbar auf "T!" klicken. BitScope beginnt daraufhin sofort, die Daten an seinen Eingängen aufzuzeichnen. Bei der vereinbarten Größe des Aufzeichnungsspeichers von 512 mal 8 Bit und 50 MHz Aufzeichnungstakt dauert das etwas 100µs. Danach wird der Inhalt des Aufzeichnungsspeichers seriell über das Programmierkabel zu PC-Übertragen und im Wave-Display angezeigt:



6 Datenauswertung mit ChipScope

Der Button "T!" wie Trigger startet die Aufzeichnung sofort. Bei jeder Wiederholung beginnt die Aufzeichnung mit einem anderen Automatenzustand. Zur Fehlersuche ist eine reproduzierbare Ausgabe, d.h. ein Aufzeichnungsstart bei einem vorgegebenen Automatenzustand erforderlich. Hierzu muss eine Triggerbedingung eingestellt werden. Stellen Sie

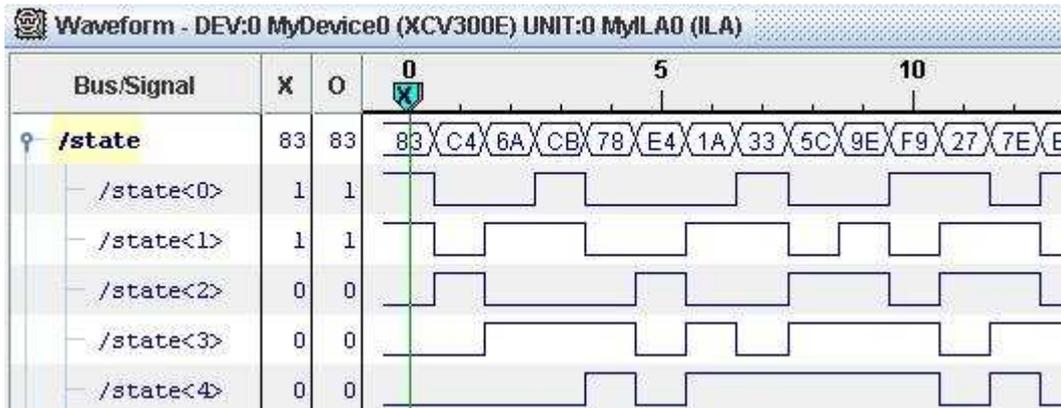
`M0:TriggerPort0 == 0001_1001` (Radix binär)

ein. Das ist der Init-Wert in der ersten Abbildung. Aufzeichnung mit "Start"-Knopf (Dreieck) auslösen. In diesem Fall wartet ChipScope, bis der Automat im Zustand "00011001" ist und zeichnet ab da auf. Kontrollieren Sie, dass die aufgezeichnete Datenfolge mit der ersten Abbildung übereinstimmt.

Wenn der Automat die Trigger-Bedingung nie erreicht, wird auch nichts aufgezeichnet. Unser Automat erreicht z.B. nie den Zustand "00000000". Probieren Sie das aus.

Tips zur Datenvisualisierung

- Rechte Mouse-Taste, "Zoom", "Zoom In" vergrößert die Darstellung.
- Markieren Sie aller 8 Signale (Links-Click 1. Signal, Shift-Links-Click letztes Signal)
- Rechts-Click und "Add to Bus"->"New Bus". Den Bus "state" nennen und in die Waveform ziehen.
- "Parken" der Maus über der neuen Signalform zur Anzeige des "Tooltips"
- Zoom bis der Inhalt des state-Busses in der Waveform zu sehen ist.



7 ChipScope-Core fest einbauen

ChipScope kann auch als Schaltung in einen Entwurf eingefügt werden.

- Erzeugen Sie ein neues Projekt (wie oben), fügen aber keinen ChipScope-Core über "New Source" hinzu.
- Rufen Sie stattdessen den "ChipScope CoreGenerator" auf (Windows-Start, Programme, ChipScope Pro, ChipScope Pro Core Generator).
- Erzeugen Sie zunächst einen "ICON" Controller, indem Sie "ICON" auswählen und mit "Next" weitergehen.
- Klicken Sie bei "Output Netlist" auf "Browse", gehen Sie dann in ihr Projektverzeichnis (keine Datei auswählen sondern icon.edn lassen) und klicken auf "Speichern".
- Die "Device Family" auf Spartan2E setzen und mit "Next" weitergehen, dann "Generate Core" klicken und danach "Start Over" wählen.
- Als nächstes einen "ILA" Core auswählen, zweimal "Next" klicken (die Einstellungen wurden vom ersten Durchgang übernommen) und im "Treffer Input"-Blatt die "Trigger Width" auf 8 setzen. Mit "Next" weitergehen und auf dem nächsten Blatt "Data Same As Trigger" aktivieren. "Next" und dann "Generate Core" anklicken. Anstelle vom "Start Over" das Fenster schließen.
- Im "Project Navigator" mit File, Open die Datei "icon_xst_example" öffnen. Von dort folgende Teile an die entsprechende Stelle in die eigene main.vhd kopieren: "component icon", "signal control0" und (vor das letzte "end") i_icon.
- Die Datei "icon_xst_example.vhd" schliessen und dafür "ila_xst_example.vhd" öffnen.
- Entsprechend dem icon die Blöcke "component ila", die Signale und i_ila kopieren und "ila_xst_example.vhd" schliessen.

Als letztes müssen noch ein paar Anpassungen vorgenommen werden:

- Bei durch das Kopieren hat ein Signal unterschiedliche Namen bekommen: "control0" und "control" sollen dasselbe Signal sein. Eines löschen, beispielsweise "control".
- In der den "port map"-Zuordnungen am Ende der Datei "control" auf der rechten Seite in in "control0" ändern.

- Jetzt fehlen nur noch die Anschlüsse der zu analysierenden Daten und der Takt. Da wir schon passende Signale haben, können die Signale "clk" und "trig0" auch gelöscht werden.
- Bei "i_ila" wird das "trig0" auf der rechten Seite durch "state" ersetzt.
- Speichern, "Generate Programming File", aber kein Download mit iMPACT. (iMPACT muss vor dem nächsten Schritt geschlossen sein.)
- ChipScope wie beim Core Inserter-Beispiel starten.
- Device0, Configure, Device 0, Select New File, aufgabe4.bit.
- File, Import, Select New File, ila.cdc
- Testen wie im Beispiel zuvor.

8 Zusatzaufgabe

Der untersuchte Zellenautomat durchläuft zyklisch eine bestimmte Zustandsfolge. Man kann diese Periodenlänge messen, indem man nach der Initialisierung des Automaten mit einem zusätzlich in die Schaltung eingebauten 8-Bit-Zähler die Schaltschritte zählt, bis der Automat zum ersten Mal wieder seinen Initialisierungszustand erreicht. Dieser Wert könnte z.B. auf led(15 downto 8) ausgegeben werden. Versuchen Sie auf diesem Wege die Zykluslänge zu bestimmen.

9 Aufräumen

- Über Menüpunkt "Project, Cleanup Project Files" automatisch generierte Design-Files löschen.
- Netzteil zur Spannungsversorgung aus der Steckdose ziehen.
- Modelsim und Projektnavigator beenden.