

Einführung in Altera Quartus II 11.0

Version 0.1

Verteiler:

Name (alphab.)	Abteilung	Ort
Laszlo Arato	EMS	NTB, Buchs
Dr. Urs Graf	INF	NTB, Buchs

Dokumentenverwaltung

Dokument-Historie

Version	Status	Datum	Verantwortlicher	Änderungsgrund
0.1	In Arbeit	9.8.2011	L. Arato	Start des Dokumentes
0.2	In Arbeit	20.2.2012	L. Arato	Neu Kapitel 5.10 PROM Programmierung

Änderungsberechtigte

Laszlo Arato	Institut EMS	NTB, Buchs
Dr. Urs Graf	Institut INF	NTB, Buchs

Dokument wurde mit folgenden Tools erstellt:

Microsoft WORD 2007

Inhaltsverzeichnis

1	Einleitung.....	3
1.1	Zweck des Dokuments.....	3
1.2	Gültigkeit des Dokuments	3
1.3	Begriffsbestimmungen und Abkürzungen.....	3
1.4	Zusammenhang mit anderen Dokumenten	3
2	Installation.....	4
2.1	Unterschiede „Subscription-Edition“ und „Web-Edition“.....	4
2.2	Download von Quartus II.....	4
2.3	Download von ModelSim-Altera	4
2.4	Installation.....	4
3	Lizenzierung	5
3.1	Arbeiten mit dem VPN und den NTB Lizenzen.....	5
3.2	Webedition Lizenz.....	6
3.3	30 Tage Evaluations-Zeit	6
3.4	Limitierungen und Hardware Unterstützung	6
4	Projekt Start mit Quartus II.....	7
4.1	Starten der Quartus II Software.....	7
4.2	Erzeugung eines neuen Projektes mit dem „New Project Wizard“	7
5	Erstes kleines Projekt in VHDL	10
5.1	Projekt Start	10
5.2	VHDL Design Eingabe	10
5.3	Erste Analyse und Synthese	11
5.4	Definition der I/O Pins	11
5.5	Erstellen eines SDC (Synopsys Design Constraint) Files	12
5.6	Kompilieren des gesamten Designs	13
5.7	Anschliessen der DE2-115 Hardware	13
5.8	Programm auf Hardware laufen lassen	14
5.9	Final Test auf der Hardware.....	14
5.10	PROM Programmierung auf dem DE2 Board	15

1 Einleitung

1.1 Zweck des Dokuments

Diese Einführung soll Studenten und anderen interessierten Personen helfen, möglichst schnell und effizient die Quartus II Software von Altera für FPGA Entwicklung zu nutzen.

1.2 Gültigkeit des Dokuments

Dieses Pflichtenheft ist für NTB internen Gebrauch.

Die Ausführungen gelten sowohl für die kostenlose Webedition Ausführung, wie auch für die Lizenzierte Vollversion. Dort wo Unterschiede bestehen, wird darauf explizit hingewiesen.

1.3 Begriffsbestimmungen und Abkürzungen

FPGA	Field Programmable Gate Array, ein programmierbarer Logikbaustein.
ALTERA	Altera Corporation ist ein Hersteller von FPGAs
Quartus II	Die offizielle Software von Altera für CPLD und FPGA Entwicklungen
VHDL	„Very High-Speed Hardware Description Language“

1.4 Zusammenhang mit anderen Dokumenten

Dieses Dokument ist das Pendant zur Einführung in die Xilinx ISE Umgebung von Urs Graf.

Folgende begleitende Dokumente sind geplant oder bereits in Arbeit:

- Einführung in ModelSim
- Einführung in VHDL Design
- Einführung in VHDL Testbench Design
- VHDL Design Guidelines

Weitere unterstützende Literatur:

- DE2_115_User_Manual.pdf

2 Installation

Die Software kann kostenlos von der Webseite von Altera heruntergeladen werden. Dazu ist eine Registrierung erforderlich.

2.1 Unterschiede „Subscription-Edition“ und „Web-Edition“

Für professionelle Anwender gibt es die Subscription-Edition, während für die meisten einfacheren Nutzer die „Web-Edition“ genügt.

Die genauen Unterschiede findet man im Alter Dokument:

http://www.altera.com/literature/po/ss_quartussevswe.pdf

2.2 Download von Quartus II

Quartus II ist der Name der Entwicklungsumgebung von Altera, sowohl für CPLD, wie auch für FPGA.

<https://www.altera.com/download/dnl-index.jsp>

Hier kann die richtige Version ausgewählt werden:

- Quartus II (Subscription Edition oder Web Edition)
- Version, z.B. 11.0, Service Pack 1
- Windows oder Linux

Das File ist ca. 2.7 GBytes gross

2.3 Download von ModelSim-Altera

ModelSim ist eine sehr starke Simulationsumgebung für digitale Schaltungen. Die ModelSim-Altera Version ist speziell eng mit Quartus II verbunden, kann aber auch einzeln genutzt werden.

Dabei ist die „Modelsim-Altera“ Version (ModelSim AE) von der kostenlosen „ModelSim Altera Starter Edition“ (ModelSim ASE) zu unterscheiden. Erstere funktioniert nur, wenn man z.B. über VPN Zugriff auf die NTB Lizenzfiles hat.

<https://www.altera.com/download/software/modelsim/11.0> oder
<https://www.altera.com/download/software/modelsim-starter/11.0>

Hier kann man auch wieder die gewünschte Version und Betriebssystem auswählen.

Das File für Windows ist 341 MBytes gross.

2.4 Installation

Man kann für Quartus und ModelSim jeweils die Installations-Files einzeln herunter laden, oder aber mit dem Altera-Installer arbeiten ... es kommt auf das selbe heraus.

Wenn man mehrere PCs mit derselben Software aufsetzen will, dann ist es vielleicht einfacher, die Files einmal herunter zu laden, um dann mehrmals zu verwenden.

3 Lizenzierung

3.1 Arbeiten mit dem VPN und den NTB Lizenzen

An der NTB haben wir 50 „floating“ Lizenzen für Quartus und Modelsim.

Wenn man diese nutzen will, muss das VPN aktiv sein, und es braucht ein lokales Lizenz-File mit einem Namen wie „Notebook1.dat“ und folgendem Inhalt:

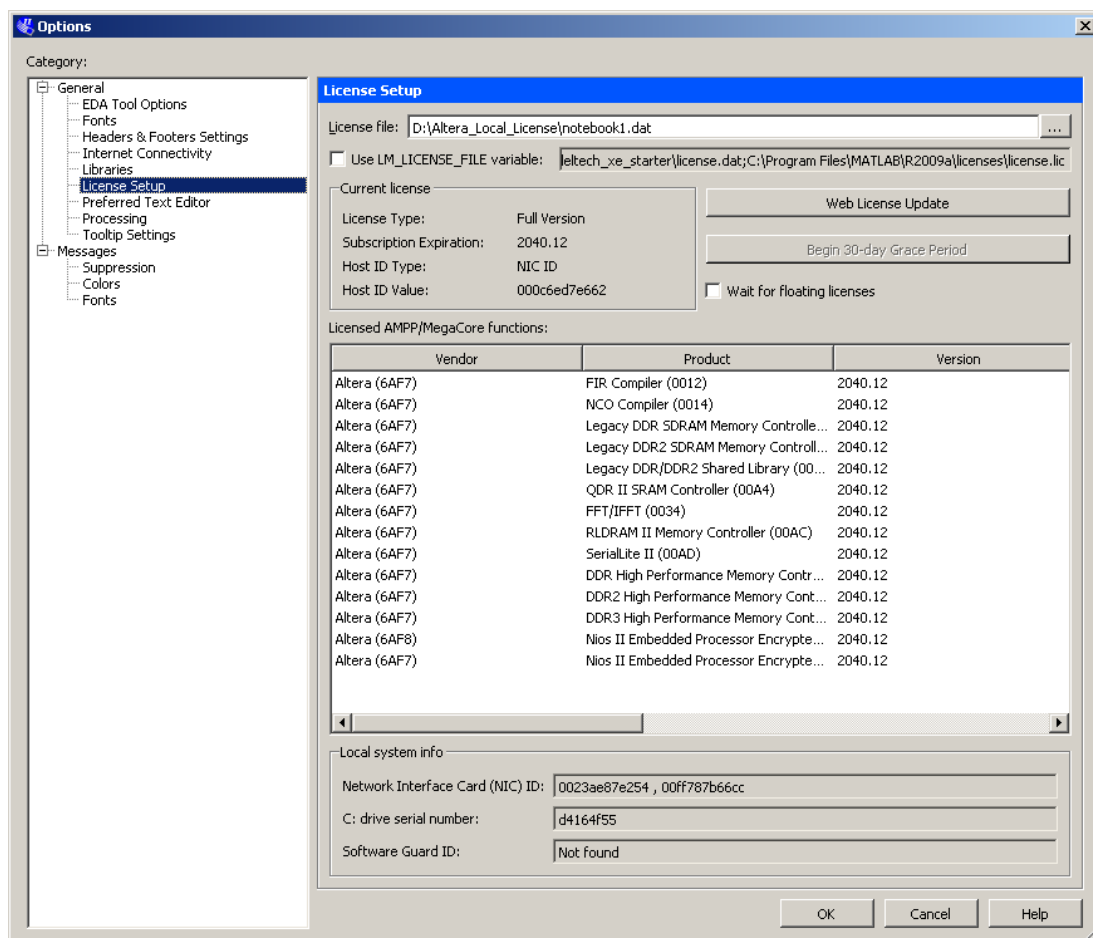
```

SERVER license-server 000c6ed7e662 1800
VENDOR alterad C:\altera\91\quartus\bin\alterad.exe
USE_SERVER
  
```

Dabei spielt die Version (hier 9.1) keine Rolle ... das gleiche File funktioniert auch für Version 11.0.

Nachdem man Quartus II gestartet hat, muss man die Lizenz einstellen:

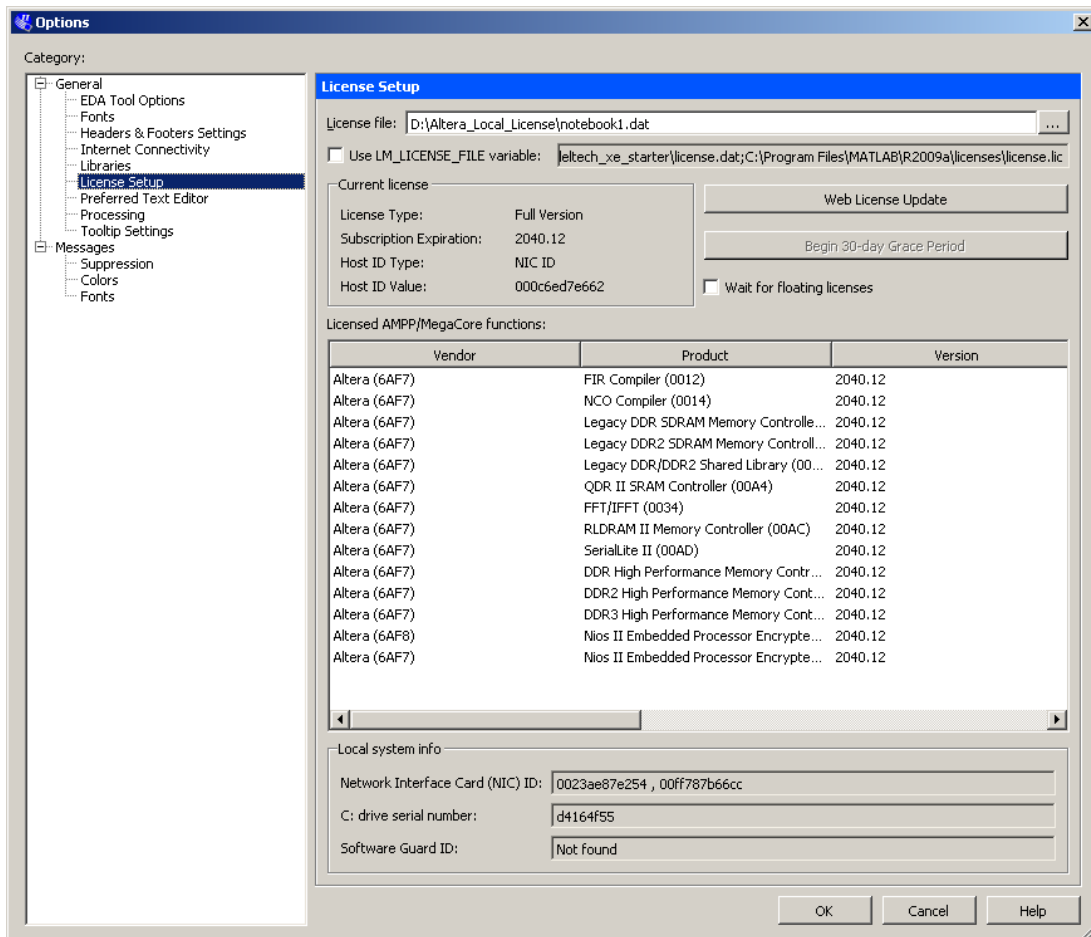
In Quartus II findet man im Menu **Tools** → **License setup** ... das folgende Fenster:



Hier muss man rechts oben den File-Namen des Lizenzfiles eingeben, oder dieses mit dem Knopf rechts oben suchen ...

3.2 Webedition Lizenz

Wenn man ohne VPN die Software brauchen will, muss man bei Altera eine „Web License“ verlangen. Dazu öffnet man in Quartus II das Lizenz-Fenster mit dem Menu **Tools** → **License setup** ... das folgende Fenster:



Hier muss man rechts oben den Knopf „Web License Update“ anklicken, und kann dann den Antrag auf eine Lizenz ausfüllen. Diese wird in der Regel innert 24 Stunden per E-Mail zugesand, und muss dann auf der lokalen Harddisk gespeichert werden. Der Pfad und Filename zu diesem File muss anschliessend im Lizenz-Fenster rechts oben eingegeben werden.

3.3 30 Tage Evaluations-Zeit

Im Lizenz Fenster von Quartus (**Tools** → **License setup**) kann eine 30-Tage Testphase gestartet werden ... aber nachher braucht es trotzdem eine Lizenz.

3.4 Limitierungen und Hardware Unterstützung

Mit der kostenlosen Web Edition Version werden nicht alle CPLD und FPGA Familien von Altera unterstützt:

Unterstützt:

MAX
 Cyclone, Cyclone II, Cyclone III
 Arria

Nicht Unterstützt:

Arria II, ausser EP2AGX45
 Stratix, ausser EP1S10, EP2S15, EP2SGX30,
 EP3SE50, EP3SL70, EP3SL50,

4 Projekt Start mit Quartus II

4.1 Starten der Quartus II Software

Es gibt verschiedene Möglichkeiten, Quartus II zu starten:

- Windows *Start* → *Programs* → *Informatik* → *Altera*
- Suche nach "Quartus" in der Startleiste von Windows 7
- Doppelclick auf ein bestehendes .QPF File eines Quartus Projektes

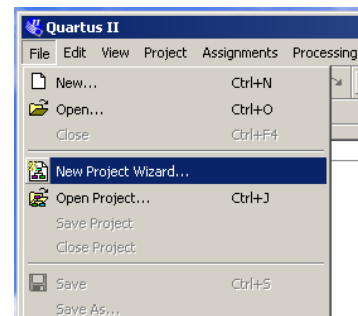
4.2 Erzeugung eines neuen Projektes mit dem „New Project Wizard“

Dies ist speziell geeignet für kleine oder ganz neue Projekte.

Im Quartus Programm wählt man dieses „Wizard“ – Hilfsprogramm mit dem Menu

File → **New Project Wizard ...** aus.

Daraufhin öffnet sich ein Dialog zur Eingabe



Schritt 1: Verzeichniss, Name und Top-Level Entity

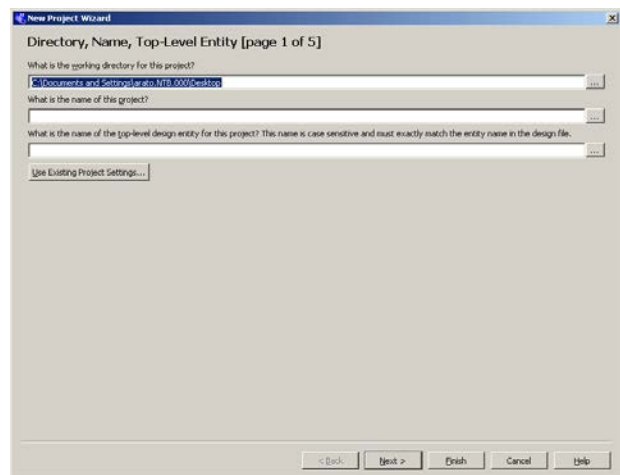
Verzeichnis: Es beschleunigt die Synthese enorm, wenn das Projekt lokal auf dem D: Laufwerk ist, und nicht immer Netz-Werk Zugriff benötigt.

Namen: Wählen sie einen starken Namen, einer der etwas aussagt, statt nur etwas wie „Projekt 1“.

Top-Level: Den Namen für die Top-Level Entity können sie jederzeit wieder ändern. (Nicht so den Projektnamen !)



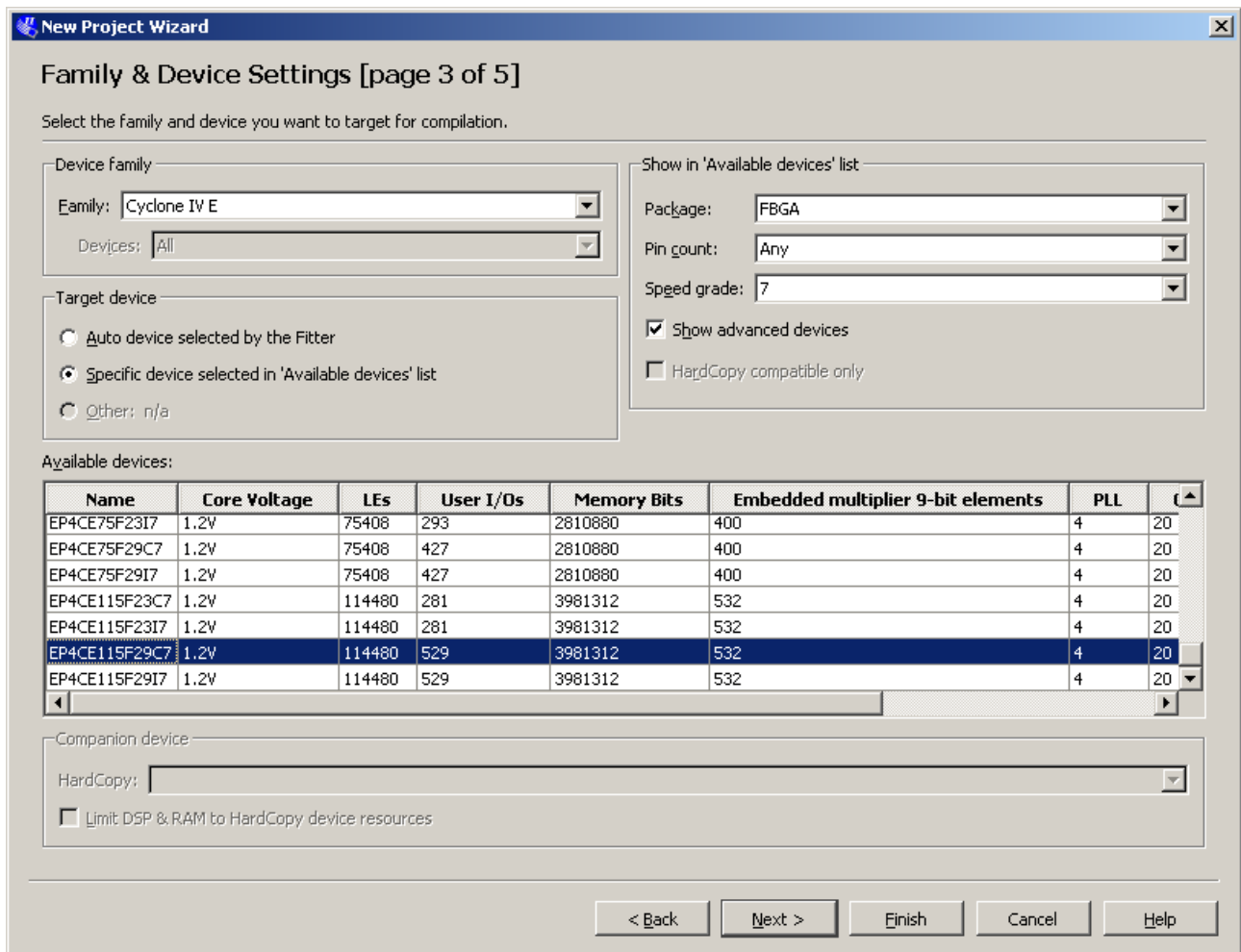
Achtung: Verzeichnis-Pfad und Projekt-Namen dürfen keine Leerzeichen enthalten!



Schritt 2: Files hinzufügen

Überspringen sie diesen Schritt ... ausser sie hätten bereits fertige Files zum einbinden ... Files können zum Projekt jederzeit später noch hinzugefügt oder entfernt werden.

Schritt 3: Chip auswählen



Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone IV E

Devices: All

Target device

Auto device selected by the Fitter
 Specific device selected in 'Available devices' list
 Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: Any

Speed grade: 7

Show advanced devices
 HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	
EP4CE75F23I7	1.2V	75408	293	2810880	400	4	20
EP4CE75F29C7	1.2V	75408	427	2810880	400	4	20
EP4CE75F29I7	1.2V	75408	427	2810880	400	4	20
EP4CE115F23C7	1.2V	114480	281	3981312	532	4	20
EP4CE115F23I7	1.2V	114480	281	3981312	532	4	20
EP4CE115F29C7	1.2V	114480	529	3981312	532	4	20
EP4CE115F29I7	1.2V	114480	529	3981312	532	4	20

Companion device

HardCopy:

Limit DSP & RAM to HardCopy device resources

In diesem Fenster bestimmen sie, welches CPLD oder FPGA sie verwenden wollen.

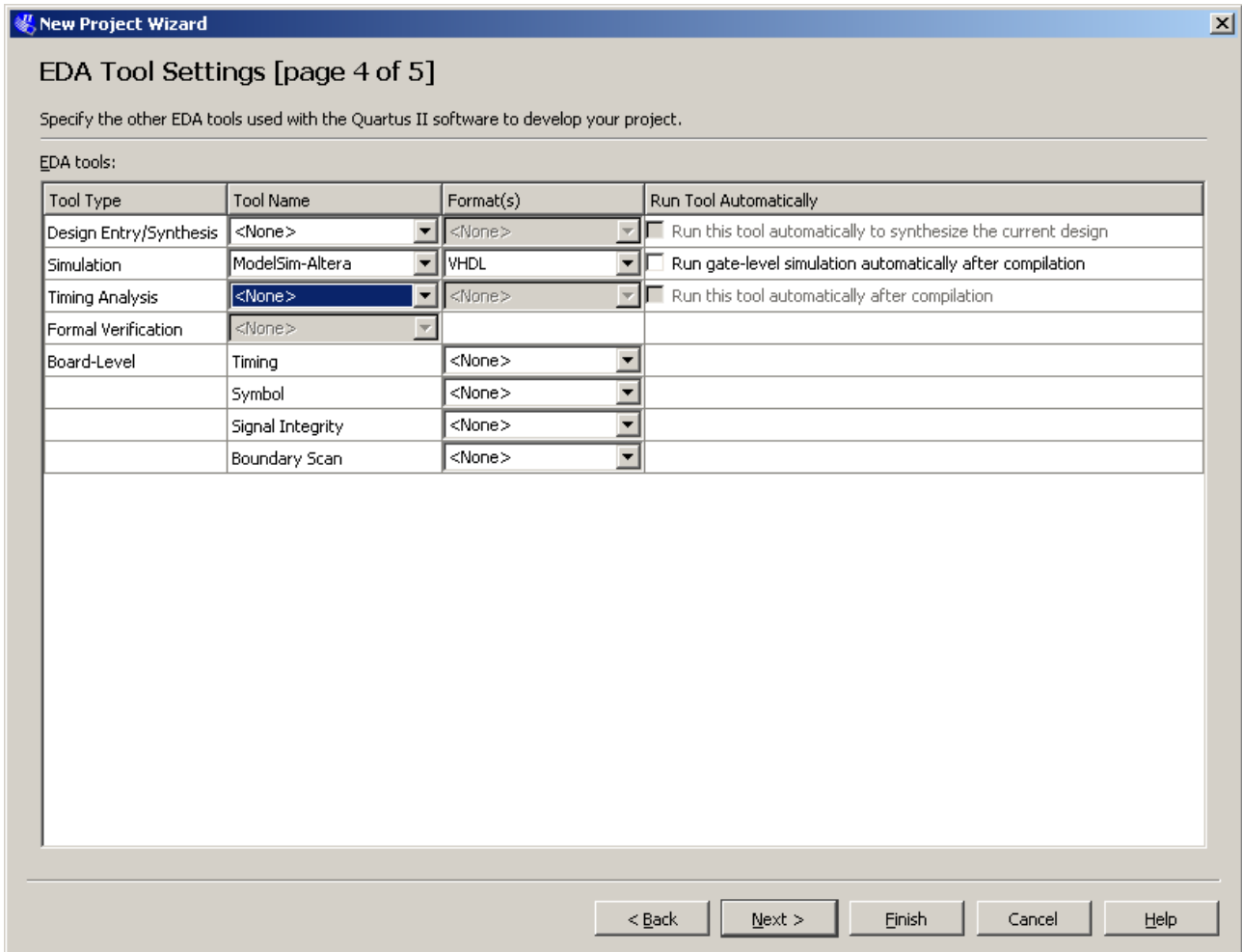
Wenn sie für das DE2-115 Board entwickeln wollen, dann wählen sie dafür das FPGA EP4C115F29C7. Dies ist ein Chip der Cyclone-IV Familie im FBGA Gehäuse (Fine-pitch Ball Grid Array).

Sie können die Auswahl mit der FPGA Familie, Gehäuse (Package), Pin-Zahl und Geschwindigkeit (Speed grade) einschränken.



Die Pin-Zahl, das Gehäuse, ja sogar die FPGA Familie kann jederzeit verändert werden. Dabei muss man jedoch beachten, dass dadurch die Pin-Positionen geändert werden müssen.

Schritt 4: EDA Werkzeug Einstellungen



EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Timing Analysis	<None>	<None>	<input type="checkbox"/> Run this tool automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

Hier können verschiedene zusätzliche 3rd-Party Werkzeuge spezifiziert und eingebunden werden.

An der NTB verwenden wir nur für die Simulation eine „spezielle“ EDA Software, nämlich ModelSim. Dabei ist zu berücksichtigen, dass wir via Altera nur eine etwas reduzierte Version haben, die als „ModelSim-Altera“ gekennzeichnet ist, und nicht die „normale“ ModelSim Version.

Dies ist die einzige Einstellung auf dieser Seite.

Schritt 5: Zusammenfassung

In diesem Schritt kann nichts mehr eingestellt werden ... es werden nur die vorherigen Eingaben und deren Konsequenzen aufgelistet.

Hier kurz kontrollieren, und dann *Finish* drücken.

5 Erstes kleines Projekt in VHDL

Dies ist ein ultra-triviales Projekt, einfach um den ganzen Fluss durch Quartus II zu zeigen. Es soll einfach für jeden der ersten 8 Schalter je ein LED ansteuern.

5.1 Projekt Start

Führen sie die Schritte im Kapitel 4 durch, nennen sie das Projekt „Quartus_Demo“ und die Top-Level Entity „quartus_demo_top“.

5.2 VHDL Design Eingabe

Erzeugen sie ein neues VHDL File mit dem Menu **File → New** und der Auswahl „VHDL File“ ein neues Design-File.

Speichern sie dieses File sogleich mit dem Menu **File → Save As** unter dem Namen „quartus_demo_top.vhd“.

Schreiben sie darin (im Quartus Editor) die folgenden Zeilen:

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;

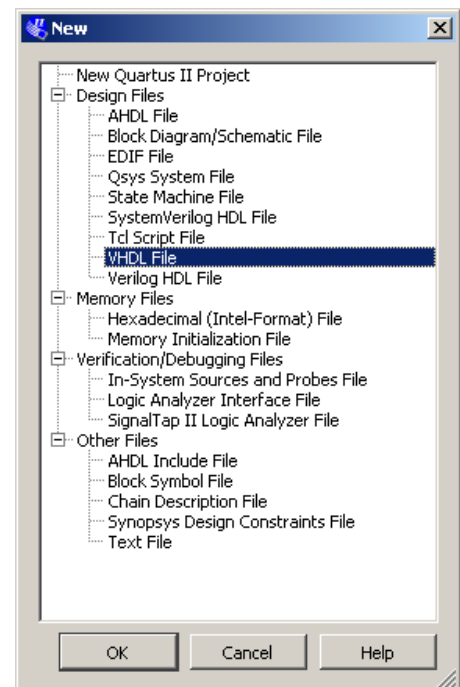
ENTITY quartus_demo_top IS
  PORT (
    Clock      : IN  std_logic;
    switch     : IN  std_logic_vector(7 DOWNTO 0);
    red_led    : OUT std_logic_vector(7 DOWNTO 0)
  );
END ENTITY quartus_demo_top;

ARCHITECTURE rtl OF quartus_demo_top IS


BEGIN

  top_reg_proc : PROCESS (clock)
  BEGIN
    IF rising_edge(clock) THEN
      red_led <= switch;
    END IF;
  END PROCESS top_reg_proc;

END ARCHITECTURE rtl;
  
```



5.3 Erste Analyse und Synthese


Starten sie eine erste Analyse und Synthese mit dem Knopf  oder im Menu mit **Processing** → **Analyze Current File**.

Wenn sie alles richtig eingegeben haben, sollte es keine Fehler und keine Warnungen geben, nur genau 10 Informationen ...

Das Projekt ist aber noch nicht fertig ... es fehlen die I/O Definitionen.

5.4 Definition der I/O Pins

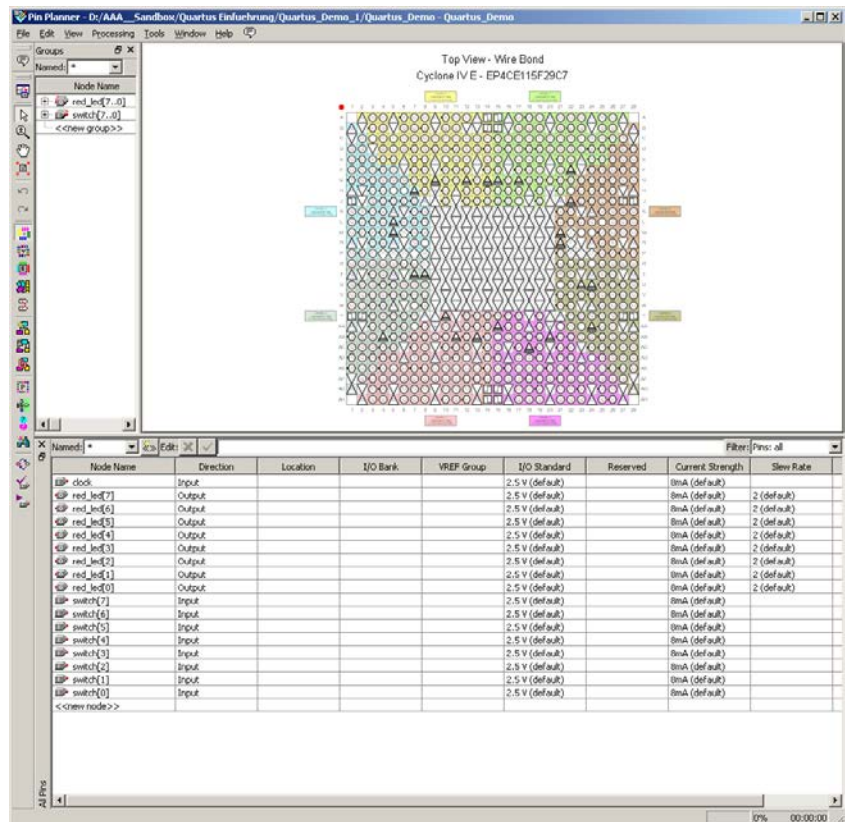
Mit dem Pin-Planner kann man relative bequem alle in der Top-Level Entity enthaltenen I/O Pins definieren, wenn das Top-Level Entity bereits einmal erfolgreich analysiert wurde. (Schritt 5.3).

Den Pin-Planner starten sie mit dem Knopf  oder mit dem Menu **Assignments** → **Pin Planner**

Klicken sie jetzt in der Tabelle in der ersten Zeile (clock) auf das dritte Feld (Location), und geben sie „Y2“ ein.

Sogleich wird daraus „PIN_Y2“, und auch I/O Bank und VRef Group werden ausgefüllt.

Damit haben sie den Takt-Eingang auf einen Pin gelegt, an dem vom DE2-115 Board her ein 50 MHz Takt anliegt.



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
clock	Input	PIN_Y2			2.5 V (default)		8mA (default)	
red_led[7]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[6]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[5]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[4]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[3]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[2]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[1]	Output				2.5 V (default)		8mA (default)	2 (default)
red_led[0]	Output				2.5 V (default)		8mA (default)	2 (default)
switch[7]	Input				2.5 V (default)		8mA (default)	
switch[6]	Input				2.5 V (default)		8mA (default)	
switch[5]	Input				2.5 V (default)		8mA (default)	
switch[4]	Input				2.5 V (default)		8mA (default)	
switch[3]	Input				2.5 V (default)		8mA (default)	
switch[2]	Input				2.5 V (default)		8mA (default)	
switch[1]	Input				2.5 V (default)		8mA (default)	
switch[0]	Input				2.5 V (default)		8mA (default)	
<new node>					2.5 V (default)		8mA (default)	

Wiederholen sie dies für alle anderen Pins mit folgenden „Locations“:

red_led 7 bis 0: H19, J19, E18, F18, F21, E19, F19 und G19
switch 7 bis 0: AB26, AD26, AC26, AB27, AD27, AC27, AC28 und AB28

Für Ausgänge (LED Pins) muss man noch die “Drive Strength” und “Slew Rate” einstellen. Dazu klickt man zweimal rasch in das entsprechende Feld, und wählt den Wert aus dem Drop-Down Menu.

Das kann mit der Zeit ziemlich mühsam werden. Einfacher geht es, wenn sie statt via Drop-Down Menu den gewünschten Wert direkt eintippen, also „8“ für 8mA und „2“ für die Slew-Rate.

Schliessen sie den Pin-Planner. Sie müssen nicht speichern, alle Änderungen werden sofort übernommen.

5.5 Erstellen eines SDC (Synopsys Design Constraint) Files

Damit die Quartus Software ihr Design richtig platzieren und auf Geschwindigkeit überprüfen kann, müssen gewisse Informationen übergeben werden.

Die Firma Synopsys hat dies mit dem Format „SDC“ (Synopsys Design Constraint“ File für die gesamte Industrie de-facto standardisiert. Auch Altera hat dieses Format übernommen.

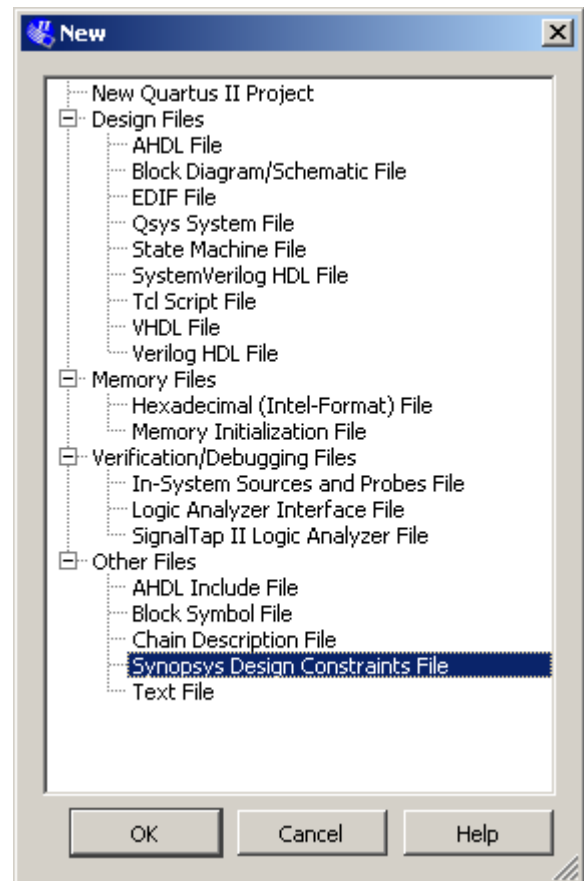
Es kann leicht mit dem Menu **File** → **New** und Auswahl (siehe rechts) erzeugt werden ...
... oder aber mit jedem beliebigen Text Editor.

Speichern sie das erzeugte neue File ab als „quartus_demo.sdc“.



Es empfiehlt sich hier, dem .sdc File den gleichen Namen zu geben, wie dem Projekt (hier „quartus_demo.sdc“.

Bei grösseren Projekten können auch einzelne Schnittstellen wie SDRAM ihre eigenen .sdc Files haben ...



Für unser kleines Projekt ist der Inhalt sehr einfach:

```
create_clock -period 20 [get_ports clock]
derive_pll_clocks
```

Die erste Zeile definiert einen Takt mit 20 ns Periode, also 50 MHz, mit Namen "clock" (letztes Argument).




Die zweite Zeile definiert, dass alle mit einem On-Chip PLL generierten Signale auch automatisch als entsprechende Takt-Signale behandelt werden, und dass deren Frequenz aus dem Eingang und der PLL-Konfiguration bestimmt wird. Wir brauchen dies zwar bei unserem Mini-Projekt noch nicht, aber es ist immer sinnvoll, diese Zeile dabei zu haben.



Die Befehle im „Synopsys Design Constraint“ File sind im TCL/TK Stil formuliert, und verwenden **nur** Kleinbuchstaben!

5.6 Kompilieren des gesamten Designs

Mit dem Knopf  oder dem Menu **Processing** → **Start Compilation** oder dem Tastenkürzel „Ctrl-L“ starten sie die Kompilation. Dabei werden jetzt nacheinander die folgenden Schritte ausgeführt:

Analysis & Synthesis:	Das Design wird analysiert und logisch interpretiert.
Fitter:	Die logischen Elemente und Port-Zuweisungen werden auf das gewählte FPGA abgebildet.
Assembler:	Das Programmierungs-File wird zusammengesetzt.
Timing Analysis:	Es wird geprüft, ob alle Timing Vorgaben erfüllt werden konnten.
EDA Netlist Writer:	Es werden .vho und .sdo Files für die Post-Fitting Timing Simulation erzeugt.

Wenn sie bis jetzt alles richtig eingegeben haben, sollte das Design ohne Fehler und nur mit einer Warnung kompilieren:

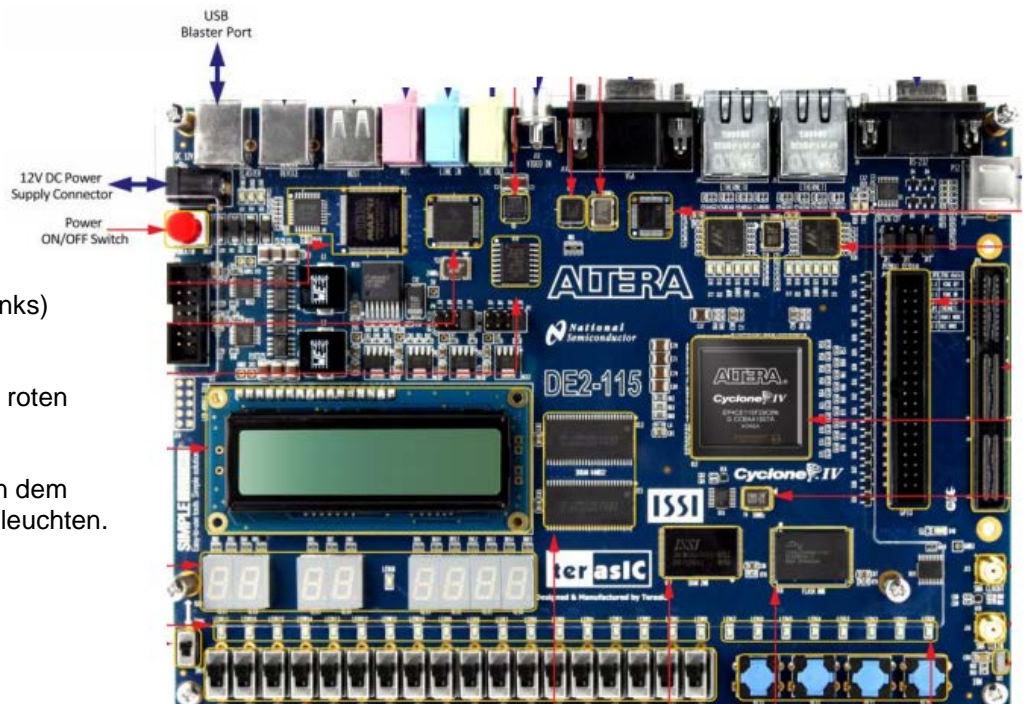
Warning: Skipped module PowerPlay Power Analyzer due tot he assignment FLOW_ENABLE_POWER_ANALYZER

Diese Warnung sagt nur aus, dass keine Schätzung der Leistungsaufnahme übersprungen wurde.


5.7 Anschliessen der DE2-115 Hardware

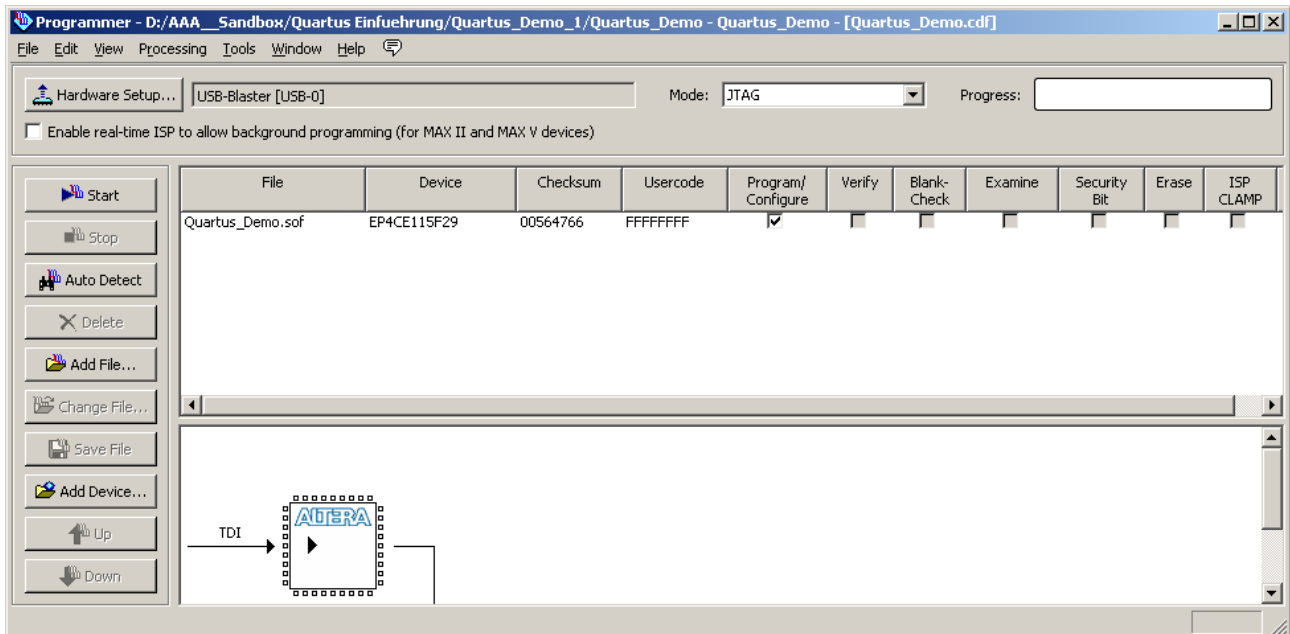
Damit sie den FPGA auf dem DE2-115 Board programmieren können, müssen sie zuerst die Hardware anschliessen.

- 1.) Schliessen sie die 12V Speisung an.
 - 2.) Schliessen sie den USB-Blaster Port (USB Stecker ganz links) an den PC an
 - 3.) Schalten sie mit dem roten Knopf das Board ein.
- Die blaue LED neben dem roten Schalter muss leuchten.



5.8 Programm auf Hardware laufen lassen


Mit dem Knopf  oder dem Menu **Tools** → **Programmer** starten Sie den Programmer.
(Selbstverständlich muss das Board eingeschaltet und über USB-Kabel mit dem PC verbunden sein ...)



Im Normalfall ist bereits die USB-Blaster Hardware auf dem DE2-115 Board erkannt (rechts neben dem Knopf „**Hardware Setup ...**“ angeschrieben als „USB-Blaster [USB-0]“.

Ist dies nicht der Fall, so muss der USB-Blaster im Menu „**Hardware Setup ...**“ ausgewählt werden.

Da dieses Programm-Teil erst nach der Kompilation geöffnet wurde, ist automatisch schon das richtige File (Quartus_demo.sof) eingestellt, und alles im JTAG Mode bereit zu Programmierung. Alles was es jetzt noch

braucht, ist den  Knopf zu drücken. Der grüne Balken oben rechts zeigt den Fortschritt.

5.9 Final Test auf der Hardware

Wenn das FPGA programmiert ist, kann man mit den Schaltern SW0 bis SW7 die LEDs LEDR0 bis LEDR7 ein- und ausschalten.

Das Programm wurde NICHT in den Flash Speicher geschrieben, sondern nur via JTAG direkt ins FPGA. Das bedeutet, dass nach einem Aus- und Einschalten (oder Stromausfall) das Programm nicht mehr im FPGA präsent ist.


Um das Programm in den Flash zu schreiben braucht es statt dem .SOF File das .POF File, sowie eine Programmier-Einstellung. Das kommt später.

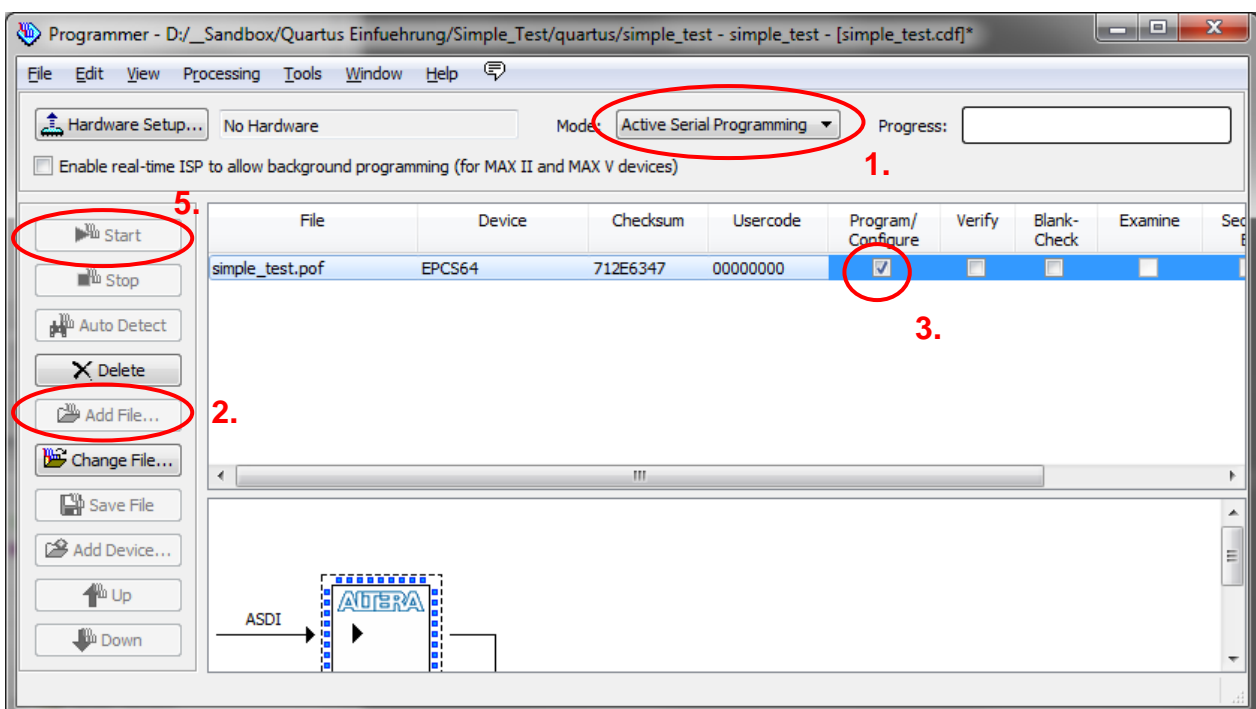
5.10 PROM Programmierung auf dem DE2 Board

Damit eine gewünschte FPGA Konfiguration nicht durch ausschalten des Boards verloren geht, kann man diese statt auf das flüchtige FPGA in einen nicht-flüchtigen externen Speicher ablegen, der dann bei jedem Einschalten das FPGA wieder konfiguriert.

Einen solchen Baustein nennt man PROM (Programmable Read Only Memory). Auf dem DE2 Board befindet sich konkret ein serielles FLASH-PROM mit der Bezeichnung EPCS64, welcher im „Active Serial Mode“ programmiert werden kann.

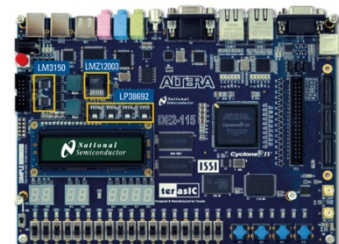
Dazu braucht es die folgenden Schritte:

Wie schon bei Kapitel 5.8 mit dem Knopf  oder Menu **Tools** → **Programmer** Programmer starten.




1. Das Feld „Mode“ im Programmer auf „Active Serial“ setzen.
Dabei werden alle bisher gewählten Konfigurations-Dateien aus der Liste entfernt, weil sie das falsche Format haben.
2. Fügen Sie mit „Add File ...“ das gewünschte Konfigurations-File im POF Format dazu.
3. Wählen Sie als Handlung „Program / Configure“ aus (Checkbox).
4. Schalten Sie den kleinen Schalter ganz unten links auf dem DE2-Board von „RUN“ auf „PROG“
5. Starten Sie die Flash-Programmierung mit „Start“
6. Stellen Sie den Schalter auf dem DE2 Board wieder auf „RUN“
7. Schalten Sie das DE2 Board aus und wieder ein. Jetzt wird die neue Konfiguration geladen.

4. + 6.



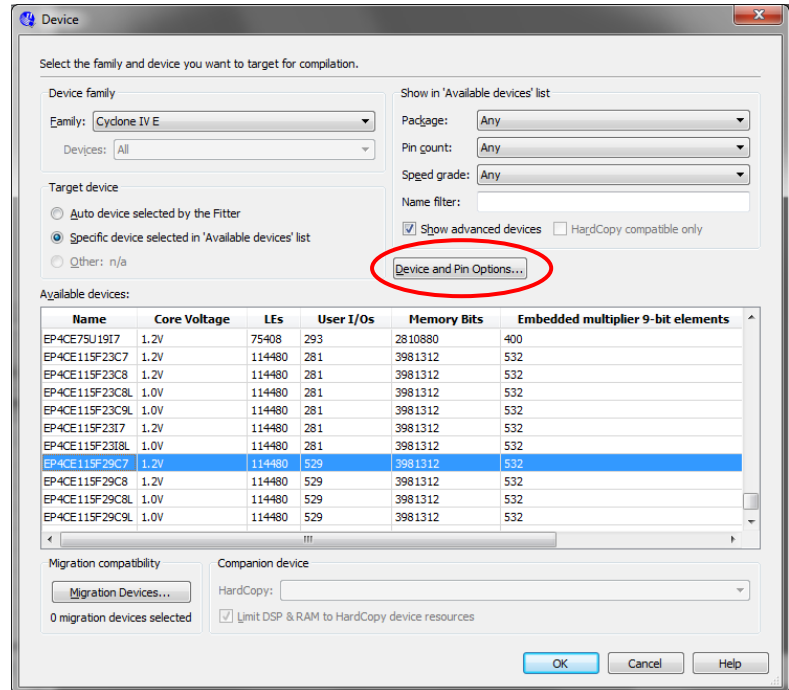


Wenn Sie vorher bei Schritt 2. (Auswahl des POF Konfigurations-Files) für Ihr Projekt kein .POF File gefunden haben, dann fehlt in der Quartus Konfiguration die Definition des entsprechenden Serial Flash Bausteins. Sie können dies leicht nachholen:

1. Öffnen Sie in Quartus mit dem Knopf  oder dem Menu **Assignments** → **Devices ...**

2. Wählen Sie den richtigen FPGA Typ aus. Für das DE2-115 Board ist dies EP4CE115F29C7.

3. Drücken Sie auf den Knopf „Device and Pin Options“ um das dazugehörige Flash Bauteil zu definieren.



4. Wählen sie im Fenster „Device and Pin Options“ im Untermenu „Configuration“ den richtigen FLASH Baustein:

Checkbox „Use configuration device“ und Wahl des Bausteins EPCS64.

5. Kompilieren Sie Ihr Projekt erneut.

