Fachgebiet Rechnerarchitektur
Fachbereich Informatik
Wolfgang Heenes

TECHNISCHE UNIVERSITÄT DARMSTADT

Kurzanleitung (Getting Started) für Quartus II 5.1

Diese Kurzanleitung beschreibt anhand eines einfachen 4-Bit Zählers die Eingabe, Simulation, Synthese und Programmierung mit der Entwicklungsumgebung Quartus II 5.1.

I. **Anlegen eines Projekts**
Nach dem Start von Quartus II in der Menüleiste unter *Files* **New Project Wizard** auswählen. Mit dem Wizard wird u. a. das Projektverzeichnis (in diesem Beispiel „zaehler"), die im Projekt benutzten Design-Files (VHDL, Verilog HDL etc.) ausgewählt. Weiterhin kann an dieser Stelle schon die *Device-Family* angegeben werden (für den Zähler wird ein Cyclone EP1C20F324C7 verwendet). Durch Drücken des Button **Finish** wird der Project Wizard geschlossen.

II. **Eingabe eines Design-Files**
In der Menüleiste wird unter ***Files*** **New** ausgewählt. Dort wählt man den File-Typ aus (*Device Design Files*, *Software Files*, *Other Files*). Bei den *Device Design Files* kann man zwischen AHDL, Block Diagram, EDIF, Verilog HDL und VHDL wählen. Nach Auswahl von z.B. Verilog HDL hat man ein Syntax Highlighting für Verilog HDL. Nach Eingabe des Codes (s. Anhang) File speichern. In der Menüleiste *Project* **Set as Top-Level Entity** auswählen um dieses File als Top-Level zu definieren.

III. **Übersetzung des Design-Files**
Der Übersetzungsvorgang (Netzlistengenerierung, Synthese) wird durch Auswahl von **Start Compilation** (Strg + L) im *Processing*- Menü ausgeführt. In der Regel wird es immer einige Warnings geben, da in der WebEdition einige Features nicht vorhanden sind.

IV. **Simulation des Desgin-Files**
Zur Simulation wird unter ***Files*** **New** die Registerkarte *Other Files* ausgewählt. Dort wird dann *Vector Waveform File* ausgewählt. Daraufhin öffnet sich ein Fenster „Waveform1.vwf". Durch drücken der rechten Maustaste im linken Teil des Fenster (Name, Value at) kommt man zum Node Finder (*Insert Node or Bus*). Im *Node Finder* kann man durch Auswahl der Filterregeln z. B. alle Input-Pins anzeigen lassen. Diese werden markiert und durch den Button >> in das Fenster „Selected Nodes" kopiert. Den gleichen Vorgang für die Output-Pins ausführen. Im Waveform-Fenster sind jetzt alle ausgewählten Pins angezeigt. Durch klick mit der linken Maustaste wird z.B. das Clock-Signal ausgewählt. Danach wird auf den Button „Count Value" geklickt. Damit kann ein Clock-Signal generiert werden. Nach speichern des Waveform Files kann die Simulation durch Auswahl von **Start Simulation** im *Processing*-Menü gestartet werden.

V. **Device- und Pin-Assignment**
Sofern nicht schon beim Anlegen des Projektes das Device ausgewählt worden ist, muß jetzt im Menü *Assignment* **Device** ausgewählt werden. Sollte der Zielbaustein in der Auswahlliste nicht vorhanden sein, muß die Filteroption von „Fastest" auf „Any" umgestellt werden. Nach Auswahl des Zielbausteins sollte das Design noch mal übersetzt werden (Strg + L). Danach kann wiederum im Menü *Assignment über* **Pins** der Assignment Editor geöffnet werden. Ein Doppelclick auf das blau unterlegte Feld <<new>> zeigt eine Liste aller zuzuweisenden Pins.

Folgende Tabelle zeigt eine mögliche Zuweisung. Für das Pining s. a. [1].

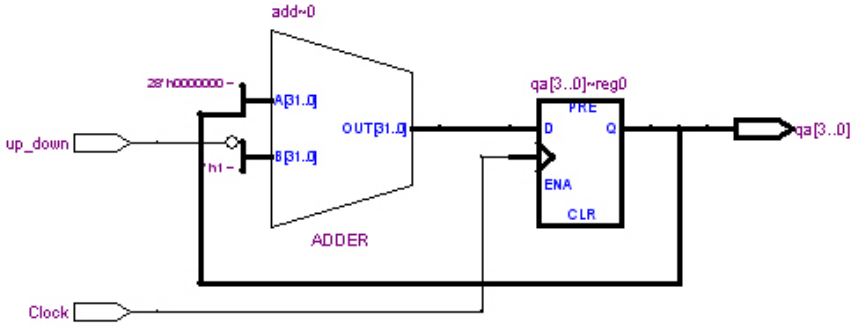
Clock	PIN_N17
qa[0]	PIN_U12
qa[1]	PIN_R11
qa[2]	PIN_T11
qa[3]	PIN_R10
up_down	PIN_R18

Die Frage nach der Übernahme des Assignments bejahen und nochmals den Übersetzungsvorgang ausführen.

VI. **Programmieren der Hardware**
Zum Programmieren der Hardware, wird im Menü *Tools* der **Programmer** aufgerufen. In der Regel sollte das Programmierfile (*.sof) bereits ausgewählt sein. Ansonsten mittels *Add File* das Programmierfile auswählen. Mit dem **Start** Button wird die Programmierung begonnen. Das Beispielprogramm benötigt einen externen Takt. Dieser wird entsprechend [1] angeschlossen.

VII. **Benutzung des RTL Viewers**

Der **RTL Viewer** befindet sich im Menü *Tools*. Nach Start des Viewers ergibt sich folgendes Bild.



Beispielprogramm:

```
module zaehler(Clock,up_down,qa);

input  Clock,up_down;
output [3:0] qa;

reg [3:0] qa;

integer direction;

always @(posedge Clock)
 begin
 if (up_down)
  begin
   direction = 1;
  end
 else
  begin
   direction = -1;
  end
 qa <= qa + direction;
 end // end always

endmodule
```

Literatur:

[1] Cyclone 324 SmartPack Rev C
[2] www.altera.com