

Tutorial für FPGA-Board und Design Tools

Übungsziel

In dieser Übung sollen Sie sich mit modernen Design-Tools vertraut machen. Der Entwurfsprozess beginnt bei der Spezifikation und endet schließlich beim Transistor. Dazwischen liegen aber noch einige Arbeitsschritte, denen wir uns in dieser Übung annähern wollen. In der ersten Aufgabe werden wir deshalb eine Schaltung auf Gatterebene erstellen und diese dann vom Synthese-Algorithmus auf Transistorebene übersetzen lassen. In der zweiten Aufgabe werden wir etwas höher gehen und eine Schaltung auf Register-Transfer-Level (RTL) betrachten.





Praktikumsboard

Auf unseren Praktikumsboards befindet sich ein Spartan-3 FPGA der Firma Xilinx, der mit 50 MHz getaktet ist. Daneben enthält es noch einen VGA Ausgang, eine RS-232 Schnittstelle, einen PS/2 Anschluss, 1MB SRAM und 2MBit Flash-RAM. An dieser Stelle wird jedoch nur auf das für den Übungsbetrieb Relevante eingegangen; weitergehende Einzelheiten finden sie in der Dokumentation zum Board, welche auf der Übungs-Webseite verfügbar ist.



Abbildung 2: Xilinx Spartan-3 Board

Auf dem FPGA-Board finden Sie rechts unten (Abbildung 2, Nummer 11) eine Reihe von Schiebeschaltern (SW7-SW0), die direkt mit Eingangsleitungen des FPGAs verbunden sind. Welcher der Name der jeweils entsprechende Eingangsleitung ist, können Sie Tabelle 2(a) entnehmen. Schalter die "oben" oder "an" sind setzen den entsprechenden Eingang auf 1, diejenigen die "unten" oder "aus" sind erzeugen eine 0 am Eingang.

Tabelle 1	: Legende zu Abbildung 2	

FPGA 1 2 Flash-RAM 10 7-Segmentanzeige 11 8 Schiebeschalter 12 8 Leuchtdioden (LED) 13 4 Taster 17 Reset Taste (lädt FPGA mit Konfiguration aus dem Flash) 18 leuchtet, wenn das FPGA fertig programmiert wurde

Tabelle 2:	Xilinx	Spartan-3	Pin-Namen
racene 2.	1 11111/1	opurtur 5	I III I (alliell

(a) Schiebeschalter							(b) Di	ruckschalte	r				
Schalter	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0	Schalter	BTN3	BTN2	BTN1	BTN0
FPGA Pin	K13	K14	J13	J14	H13	H14	G12	F12	FPGA Pin	L14	L13	M14	M13

(c) LEDs										
LED	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0		
FPGA Pin	P11	P12	N12	P13	N14	L12	P14	K12		

Die Drucktaster sind mit der Nummer 13 in Abbildung 2 beschriftet. Im gedrückten Zustand setzen sie die Eingänge am FPGA auf 1. Die entsprechenden Eingangsleitungen finden Sie in Tabelle 2(b).

Die 8 Leuchtdioden (Kurzform LED, für <u>Light-Emitting-Diode</u> - Licht aussendende Diode), welche mit der Nummer 12 beschriftet sind, werden von den Ausgängen des FPGA angesteuert. Um eine LED einzuschalten muss der zugehörige Ausgang auf *1* gesetzt werden. Die Namen der entsprechenden Ausgangsleitungen können Sie Tabelle 2(c) entnehmen.

Inbetriebnahme

- 1. Verbinden Sie das Board über den JTAG-Anschluss (22 in Abb. 2) mit dem Parallel-Port Ihres Computers. *Beachten Sie die Polung des Steckers! Die Beschriftung muss mit der auf dem Board übereinstimmen.*
- 2. Stellen Sie die Stromversorgung her (25 und Steckdose).
- 3. Sobald das Board mit Strom versorgt ist, wird die Konfiguration des FPGAs aus dem Flash-RAM geladen.

Xilinx ISE (Integrated Synthesis Environment)

 $Starten \ Sie \ die \ FPGA-Entwicklung sumgebung \ mittels \ Start \rightarrow Programme \rightarrow Xilinx \ ISE \ 9.1i \rightarrow Project \ Navigator$

Das Hauptfenster (Bild 3) ist in vier Bereiche aufgeteilt:

- Übersichtsfenster zeigt hierarchisch alle zum Projekt gehörigen Files an. Ein Doppelklick auf ein File startet das zugehörige Tool im Editierfenster.
- **Prozessfenster** Ein Doppelklick mit der linken Maustaste bzw. der Menüpunkt "Run" in dem auf einen Rechtsklick erscheinenden Pop-Up Menü startet den zugehörigen Prozess. Ein grünes Häckchen markiert einen erfolgreich abgelaufenen Prozess.

Konsole zeigt die Ausgabe des zuletzt abgelaufenen Prozesses inkl. Warnungen und Fehlermeldungen an.

Editierfenster Hier können z.B. Verilog Files bearbeitet werden.



Abbildung 3: Xilinx Project Navigator

Neues Projekt anlegen

Erstellen Sie zuerst ein neues Projekt: File \rightarrow New Project (Bild 4). Als "Top-Level Module Type" wählen Sie entweder "Schematic", wenn Sie eine Schaltung in Form von Schaltungs-Diagrammen eingeben wollen, oder "Verilog HDL", wenn sie mit Verilog arbeiten möchten.¹

Im nachfolgenden Dialog wählen Sie für das Digilent Spartan-3 Board folgende Einstellungen (Bild 5(a)):

Device Family:	Spartan 3	Synthesis Tool:	XST (VHDL/Verilog)
Device:	xc3s200	Simulator:	Modelsim XE Verilog oder ISE Simulator
Package:	ft256	Preferred Language:	Verilog
Speed Grade:	-5		

Die folgenden beiden Dialoge können Sie mit "Weiter" überspringen. Klicken Sie dann auf "Fertigstellen".

¹Speichern Sie Ihre Projekte auf einem lokalen Verzeichnis (also irgendwo auf C:), denn das Xilinx ISE hat immer wieder Probleme mit Dateien die auf einem Netzlaufwerk liegen.

Biere • Kroening Weissenbacher • Wintersteiger **Digitaltechnik**

🔤 New Project Wizard - Create New Project 📃 🗖 🔀							
Enter a Name and Location for the Project							
Project Name:	Project Location						
gatter	C:\fpga_projects\gatter						
Salact the Tune of Ten Louis Source for the Preiset							
Top-Level Source Tupe:							
Schematic							
More Info	<back next=""> Cancel</back>						

Abbildung 4: Neues Projekt erstellen

155	New Project Wizard - Device	Properties				
ſ	Select the Device and Design Flow for	the Project				
	Property Name	Value		15	New Source Wizard - Select Source Type	
	Product Category	All	~			
	Family	Spartan3	~		Normal Architecture Wizard)	
	Device	×C35200	~		Schematic	
	Package	FT256	~		Test Bench WaveForm	
	Speed	-5	~		User Document	
					V Verilog Module	<u>File name:</u>
	Top-Level Source Type	HDL	\sim		Verilog Test Fixture	logic
	Synthesis Tool	XST (VHDL/Verilog)	✓		VHDL Library	Location:
	Simulator	Modelsim-XE Verilog	~		VHDL Package	
	Preferred Language	Verilog	✓		🔚 VHDL Test Bench	C: \rpga_projects \gatter
	Enable Enhanced Design Summary					
	Enable Message Filtering					
	Display Incremental Messages					
						Add to project
C	More Info	< <u>B</u> ack <u>N</u> ext>	Cancel	(More Info	< Back Next > Cancel

(a) Einstellungen für Spartan-3 Board

(b) Neue Datei hinzufügen

Abbildung 5: Projekteinstellungen und Datei hinzufügen

Dateien hinzufügen

Fügen Sie dem Projekt nun eine neue Datei (**Project** \rightarrow **New Source**) hinzu. Es öffnet sich ein neuer Dialog, indem Sie auf der linken Seite den Dateityp auswählen können. Für Schaltungs-Diagramme wählen wir hier also "Schematic", während wir für Verilog-Dateien "Verilog Module" auswählen. Auf der rechten Seite können Sie die neue Datei noch benennen. Klicken Sie auf "Weiter" und dann "Fertigstellen" (Siehe Bild 5(b)).

Schaltungen zeichnen

Platzieren Sie dazu zuerst die Gatter indem Sie das "Add Symbol" Tool auswählen und in der Kategorie "Logic" das entsprechende Gatter markieren. Anschliessend können Sie mit "Add Wire" die Bauelemente verdrahten und mit "Add I/O Marker" die Ein- und Ausgänge festlegen. Den "I/O Markern" werden eindeutige Namen zugeordnet; um diese zu ändern wählen Sie das "Select Cursor"-Tool aus und klicken doppelt auf den entsprechenden Marker. Es öffnet sich ein Eigenschaftsdialog, in dem Sie den Namen ändern und den die Signalrichtung (Ein- oder Ausgang) festlegen können.

Biere • Kroening Weissenbacher • Wintersteiger **Digitaltechnik**



Abbildung 6: Schaltung zeichnen

FPGA-Pins zuweisen

Um I/O-Marker oder Ein- und Ausgänge eines Verilog-Modules physikalischen Anschlüssen auf dem FPGA zuzuweisen, öffnen Sie den Pin-Assignment-Editor. Dazu öffnen Sie im Prozessfenster die Kategorie **User Constraints** und klicken doppelt auf den Prozess **Assign Package Pins**. Sollten im Projekt noch keine User-Constraints-Datei vorhanden sein, so werden Sie hier gefragt ob Sie eine hinzufügen möchten. Daraufhin öffnet sich der Package-Pins-Editor.

Im linken unteren Bereich dieses Editors können Sie nun jeden Ein- oder Ausgang Ihrer Schaltung einem physikalischen Pin auf dem FPGA zuweisen (Abb. 7). Je nach gewünschter Funktion können Sie z.B. Eingänge einem Druck- oder Schiebeschalter, oder Ausgänge einer LED zuweisen. Die entsprechenden Pin-Namen sind in den Tabellen 2(a) bis 2(c) zusammengefasst. Haben Sie sich für eine Funktion entschieden, so geben sie den entsprechenden FPGA-Pin-Namen hier im Feld LOC an.

Achten Sie dabei darauf, dass Sie keinen Ausgang einem Schalter zuweisen; Ein Kurzschluss mit möglicher Beschädigung des FPGAs wäre die Folge!

Speichern Sie Ihre Änderungen und schliessen Sie das Fenster. Achten Sie darauf, dass bei den nun folgenden Schritten im *Übersichtsfenster* das Hauptmodul ihrer Schaltung ausgewählt ist!



Abbildung 7: Pin Assignment

JTAG-Clock setzen

Für Boards, welche mit einem parallelen Kabel an den Computer angeschlossen werden, sollten Sie noch die JTAG-Taktung anpassen. Dazu öffnen Sie das Eigenschaftsmenü indem Sie mit der rechten Maustaste auf den Prozess **Generate Programming File** klicken und **Properties**... wählen. Im Eigenschaftsdialog stellen Sie dann die "FPGA Start-up Clock" im Unterbereich "Startup Options" auf **JTAG Clock** (Abb. 8).

Process Properties			
<u>C</u> ategory			
General Options Configuration Options Startup Options		Startup Options	
neauback options	Property Name	Value	<u>~</u>
	FPGA Start-Up Clock	JTAG Clock	v 📃
	Enable Internal Done Pipe		
	Done (Output Events)	Default (4)	~ ~
		Property display level: Standard 💌	<u>D</u> efault
	ОК	Cancel Apply	<u>H</u> elp

Abbildung 8: JTAG-Clock setzen

Synthese

Am einfachsten starten Sie die Synthese Ihrer Schaltung, indem Sie den Prozess **Generate Programming File** starten (rechte Maustaste, **Run**). Es werden dann zuerst die Prozesse **Synthesize** und **Implement Design** ausgeführt, bevor tatsächlich ein Programming-File für den FPGA erstellt wird. Der erste dieser Prozesse wandelt Ihre Schaltung übrigens so um, dass sie mit den auf dem FPGA vorhandenen Primitiven (LUTs) implementiert werden kann. Der zweite kümmert sich danach um die physikalische Aufteilung der Schaltung auf dem FPGA.

Achten Sie bei der Synthese auf Fehler und Warnungen, die Sie in den dafür vorgesehenen Fenstern (Errors und Warnings) am unteren Rand der Entwicklungsumgebung einsehen können.

RTL Schematic

Wenn Sie nach erfolgreicher Synthese den Prozess **View RTL Schematic** in der Kategorie **Synthesize – XST** starten, so können Sie einen Blick auf die erzeugte Schaltung werfen. In der ersten dargestellten Ansicht (Abb. 9(a)) sehen Sie das *Interface* des Designs, so wie es z.B. durch eine **module** Deklaration im Verilog-Code spezifiziert wird. Mit einem Doppelklick auf dieses Symbol wird der innere Aufbau stufenweise dargestellt (Abb. 9(b)).





Programmierung des FPGA

Wenn die Synthese fehlerfrei abgeschlossen ist, können wir den FPGA programmieren. Dazu starten wir den Prozess **Configure Device (iMPACT)** in der Kategorie **Generate Programming File** (rechte Maustaste, **Run**).

Die Entwicklungsumgebung fragt Sie daraufhin nach der bevorzugten Verbindungsmethode, welche standardmäßig auf JTAG mit der Option Automatically Connect to cable and identify Boundary-Scan chain gestellt ist und auch so belassen werden sollte.

In unserem Fall erscheinen im folgenden Fenster zwei Symbol für Geräte auf der JTAG-Chain. Das linke der beiden steht dabei für das FPGA (xc3s200), während das rechte Symbol für das Flash-RAM (xcf02s) auf dem Board steht, das wir in dieser Übung nicht benötigen werden. Gleichzeitig werden Sie nach dem Configuration File für den FGPA gefragt. Wählen Sie das entsprechende Programming-File in diesem Datei-Auswahl-Dialog. In unserem Fall trägt die Datei immer den Namen des Projekts und die Erweiterung .bit. Nach einem klick auf Open werden Sie gefragt, welche Datei in das Flash-RAM auf dem Board übertragen werden soll. Da wir das Flash-RAM nicht benötigen, wählen Sie hier einfach die Option **Bypass**.

Um die eingestellten Programming Files nun endgültig auf das Board zu übertragen, klicken Sie rechts auf das FPGA-Symbol und wählen den Menüpunkt **Program**.... Es wird ein neuer Dialog gezeigt, in dem noch einige Optionen gesetzt werden können. Hier brauchen Sie allerdings nichts zu verändern – nach einem Klick auf OK werden alle Daten auf das Board übertragen.

Manchmal passiert es, dass die Übertragung fehlschlägt. Sollte dies geschehen, probieren Sie es einfach noch einmal; in der Regel liegt dabei nicht wirklich ein Fehler vor, sondern nur eine temporäre Störung.



Abbildung 10: Design auf den FPGA übertragen.

Testbenches

Nachdem wir ein Modul implementiert haben, wollen wir dieses natürlich auch testen. Üblicherweise legt man dazu eine *Testbench* an. Dabei handelt es sich um ein spezielles Modul, welches die notwendigen Ein- und Ausgabesignale (so genannte Testvektoren) für Ihre Implementierung erzeugt.

Abb. 11 zeigt eine einfache Testbench für die Implementierung eines Multiplexers mit dem Namen "Mux2tol". Es wird dort nur das zu testende Modul instantiiert und die Ein- und Ausgänge des Moduls werden auf lokale wires und regs verbunden. Innerhalb des initial-Blocks werden die Eingänge dann auf verschiedene Werte gesetzt und man kann in einer Simulation die entstehenden Ausgangswerte betrachten. Beachten Sie die Zeilen in Abb. 11, welche mit #10 beginnen – damit wird die Ausführung in der Simulation für 10 Zeiteinheiten angehalten. Wie groß eine Zeiteinheit ist legt übrigens die `timescale-Direktive in der ersten Zeile fest. Beachten Sie auch, dass die Testbench üblicherweise nicht synthetisierbar ist, denn die Kommandos zur Zeitmodellierung können nicht direkt in Hardware übersetzt werden.

Simulation

Um die Simulation eines Moduls zu starten, wählen Sie links, oberhalb des Übersichtsfensters, in der Auswahlleiste, die normalerweise Synthesis/Implementation anzeigt, den Punkt Behavioral Simulation. Stellen Sie dann sicher, dass das zu simulierende Modul (also üblicherweise eine Testbench) im Übersichtsfenster aktiviert ist. Im Prozessfenster wird daraufhin ein Prozess mit dem Namen Simulate Behavioral Model zur Verfügung gestellt, welches sich in einer Unterkategorie befindet, die den Namen des eingestellten Simulators trägt (also entweder ModelSim XE oder ISE Simulator – je nachdem, was Sie beim Anlegen des Projekts eingestellt haben).

Wenn Sie diesen Prozess gestartet haben, werden die Quelldateien kompiliert und die Simulation gestartet. Ähnlich wie in Abb. 12 werden die verschiedenen Signale angezeigt und Sie können alle Werte auf den Signalen zu den verschiedenen Zeitpunkten der Simulation ablesen.

'timescale 1ns/100ps **module** Mux2to1_tb(); // ein paar Kabel **reg** i0, i1, sel; wire o; // Eine Multiplexer-Implementierung Mux2to1 mux1(i0, i1, sel, o); // die Testwerte initial begin #10 i0 = 1'b0;i1 = 1'b0;sel = 1'b0;#10 i0 = 1'b1;i1 = 1'b0;sel = 1'b0;// ... mehr Tests hier... end endmodule

Abbildung 11: Typische Testbench



Abbildung 12: Simulation des Multiplexers