



Übungsblatt 1

Einführung in die Xilinx Vivado FPGA Design Plattform

Abgabefrist: Bearbeitung in der Übung am 16.04.2018

1. Einführung

In dieser Übung werden wir einen einfachen digitalen Entwurf als praktische Einführung implementieren. Dazu gehören die Entwicklung eines Entwurfs, welcher die vorgegebene Spezifikation erfüllt, die Umsetzung dieses Entwurfs in VHDL, die Entwurfssimulation und schließlich, die Umsetzung des Entwurfs auf einem realen FPGA. Alle Schritte werden mit den Xilinx-Tools durchgeführt.

Für die Laboraufgaben, werden Sie VHDL nutzen und mit der Entwicklungsumgebung (IDE) Vivado¹ von Xilinx arbeiten. Diese Übung führt Sie durch die Grundlagen, von der Erstellung eines neuen Projekts in Vivado, über den Entwurf eines einfachen VHDL-Moduls welches eine kombinatorische Logik implementiert, bis hin zur Überprüfung des Entwurfs in einer Simulation. Schließlich, testen Sie Ihren Entwurf auf einem realen FPGA Modul.

1.1. Schritt 1: Erstellung eines neues Projektes in Xilinx Vivado

Als ersten Schritt, werden Sie ein neues Projekt in Vivado erstellen:

1. Starten Sie Xilinx Vivado (auf allen Labor-PCs installiert).
2. Erstellen Sie ein neues Projekt mit dem Namen **lab1**. Den Zielordner können sie frei auswählen. Wählen Sie 'RTL project' als Zielprojekt und stellen Sie sicher, dass 'Do not specify sources at this time' ausgewählt ist.
3. Sie werden aufgefordert, einige Einstellungen für das Projekt vorzunehmen. Verwenden Sie die folgenden Einstellungen:

¹ <https://www.xilinx.com/support/download.html>

- Product Category: All
- Family: Artix-7
- Sub-Family: Artix-7
- Package: csg324
- Speed grade: -1
- Temp grade: C
- Part: xc7a100tcsg324-1

Nehmen Sie sich Zeit, um sich mit der Entwicklungsumgebung vertraut zu machen. Beachten Sie, dass zu diesem Zeitpunkt keine Quellen zu dem Projekt hinzugefügt werden. Verwenden Sie diese Projekteinstellungen in den folgenden Projekten für alle Übungsaufgaben!

1.2. Schritt 2: Erstellen eines Entwurfs in VHDL

Sie werden ein einfaches Modul, einen 4-Bit-Addierer mit Carry-Out-Signal in VHDL beschreiben und implementieren. Das Modul ist wie folgt spezifiziert:

- zwei 4-Bit-Eingangsports, **a** und **b**
- zwei Ausgangsports, 4-bit **sum** und 1-bit **carry_out**

Erstellen Sie eine VHDL-Quelldatei für das Projekt:

1. Klicken Sie auf *Flow Navigator > Project Manager > Add Sources* und wählen Sie *Add or create design sources*. Klicken Sie auf *Next*.
2. Klicken Sie auf *Create File* und wählen Sie *File Type > VHDL*. Geben Sie den Namen **adder.vhd** unter *File Name* ein und klicken Sie auf *Finish*.
3. Ein neues Fenster *Define Module* wird geöffnet. Geben Sie die *I/O Ports* gemäß der Spezifikation an und klicken Sie auf *Ok*. Die Datei **adder.vhd** wird daraufhin im Fenster *Sources* angezeigt. Öffnen Sie die Datei mit einem Doppelklick auf den Dateinamen.
4. Wie auch bei vielen anderen Programmiersprachen beginnt der VHDL-Code mit einer Liste von benutzten Bibliotheken. Der generierte Code sollte bereits einen Import für *ieee.std_logic_1164.all* enthalten. Fügen Sie einen weiteren Import für *ieee.std_logic_unsigned.all* hinzu. Die spätere Bibliotheksdefinition ermöglicht die Anwendung der arithmetischen Operation "Addition" auf "std_logic" Typen der Signale, wodurch die Signaltypkonvertierung nicht mehr nötig ist.
5. Anschließend beschreiben Sie den Architekturteil des erzeugten VHDL-Moduls. Definieren Sie ein temporäres 5-Bit-Signal **temp**, dem das Ergebnis der Addition der an den Eingangs-Ports anliegenden Signale zugewiesen werden soll. Die entsprechenden

Bits des internen Signals **temp** werden den Ausgangsanschlüssen des Addierer-Moduls zugeordnet. Verwenden Sie den Verkettungsoperator **&**, um die Größe des Eingangs-Ports intern auf 5 Bits zu erweitern.

6. Wenn der Entwurf der VHDL-Datei abgeschlossen ist, speichern Sie die Datei und prüfen Sie nach Syntaxfehlern. Prüfen Sie das Design auf Synthetisierbarkeit. Klicken Sie dazu auf *Flow Navigator > RTL Analysis > Open Elaborated Design*. Das Design wird jetzt übersetzt und die RTL-Darstellung Ihres Moduls anschließend geöffnet (Abbildung 1.1).

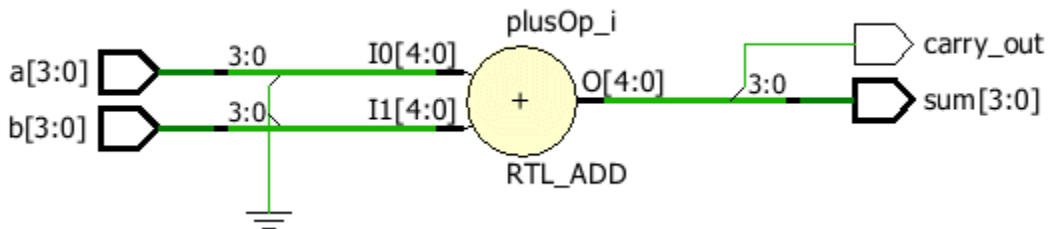


Abbildung 1.1. RTL Schema des Addierer-Moduls.

1.3. Schritt 3: Durchführen einer einfachen Simulation

Es gibt zwei Möglichkeiten den VHDL-Entwurf im Vivado Simulator zu überprüfen. Entweder durch die Verwendung einer VHDL-Testbench die im einfachsten Fall eine Abfolge von Signaländerungen definiert oder durch Änderung der Eingangssignale direkt im Simulator. In dieser Übung werden wir die zweite Methode verwenden.

1. Klicken Sie auf *Flow navigator > Simulation > Simulation settings*. Wählen Sie *VHDL* als Simulationssprache und *Vivado Simulator* als Zielsimulator. Ändern Sie den Wert der *xsim.simulation.runtime* Variable im Tab *Simulation* auf 0 ns und klicken Sie auf *Ok*.
2. Klicken Sie auf *Run Simulation > Run Behavioral Simulation*, um die Simulation zu starten. Neuen Fenster werden geöffnet (*Scopes*, *Objects* und *Waveform*). **Nehmen Sie sich Zeit, um sich mit der Simulationsumgebung vertraut zu machen.**
3. Legen Sie Eingangssignale für die Simulation an. Klicken Sie im Fenster *Objects* mit der rechten Maustaste auf die Signalnamen **a** und **b** und wählen Sie *Force constant*. Setzen Sie das Eingangssignal **a** auf den Wert 0101 und das Eingangssignal **b** auf den Wert 1010.
4. Lassen Sie die Simulation 10 us laufen, indem Sie **run 10 us** in die *Tcl Console* eingeben. Überprüfen Sie die Ergebnisse im *Waveform*-Fenster. Wie sind die Werte von **sum** und **carry_out**?
5. Jetzt stellen Sie den Wert von Signal **b** auf 1011 um. Lassen Sie die Simulation weitere 10 us laufen. Überprüfen Sie die Ergebnisse. Hat sich das **carry_out** Signal geändert? Wie ist der Wert von **sum**?

Die Ergebnisse der Simulation sind in der Abbildung 1.2 dargestellt.

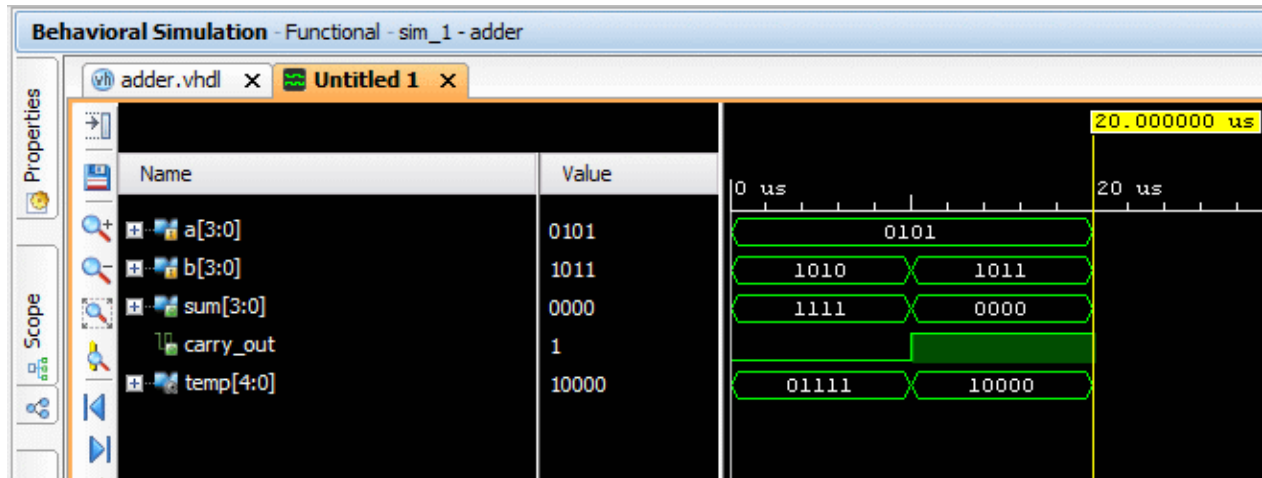


Abbildung 1.2. Ergebnisse der Simulation.

1.4. Schritt 4: Testen des Entwurfs auf einem FPGA

Sobald das korrekte Verhalten des Entwurfs überprüft wurde, können Sie die Funktionalität direkt auf der FPGA-Hardware testen. Sie werden die **Nexsys-4-Entwicklungsboards** von Digilent verwenden. Das FPGA Board ist mit einem **Xilinx Artix-7 FPGA-Modul** und mit einigen zusätzlichen Komponenten wie Switches, Sieben-Segment-LCD-Displays und LED-Dioden ausgestattet.

1. Bevor Sie mit der FPGA-Implementierung anfangen können, muss folgendes beachtet werden. Die LEDs und die Schalter des FPGAS sind mit bestimmten Pins des Artix-7 Chips auf dem Board verbunden. Um die Modul-Ports mit den gewünschten Pins zu verbinden, müssen Sie eine Verbindungsspezifikation bereitstellen. Klicken Sie auf *Flow Navigator* > *Project Manager* > *Add Sources* und wählen Sie *Add or create constraints*. Klicken Sie auf *Next*. Jetzt klicken Sie auf *Create file* und geben sie **adder.xdc** als *Filename* ein. Ihr Ziel ist es den Eingangsport **a** mit den Schaltern SW15-SW12 und den Eingangsport **b** mit den Schaltern SW11-SW8 zu verbinden. Der Ausgangsport **sum** wird mit den Leds LD3-LD0 verbunden und der Ausgangsport **carry_out** mit der Led LD4. Die Verdrahtung zwischen den FPGA-Pins und den I/O Modulen auf dem Nexsys-4 Board ist im Dokument *Nexsys-4 FPGA Board Reference Manual* zu finden (Seite 18). Vervollständigen Sie die **adder.xdc** wie angegeben und speichern Sie die Datei. Verwenden Sie das mitgelieferte Beispiel in Abbildung 1.3 als Referenz.

```
# map port a[3] to the related pin of SW15 (P4)
set_property PACKAGE_PIN P4 [get_ports {a[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
```

```

# map port a[2] to the related pin of SW14 (P3)
set_property PACKAGE_PIN P3 [get_ports {a[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
....

```

Abbildung 1.3: Beispiel der Pin-Zuordnung in der XDC-Datei

- Jetzt kann die Implementierung erfolgen. Der erste Schritt in der Umsetzung ist die Synthese. Dabei wird die RTL-Beschreibung des Designs zu logischen Gattern konvertiert. Um die Synthese zu starten, klicken Sie auf *Flow-Navigator* > *Synthese* > *Run Synthesis*. Ein Fenster öffnet sich und zeigt, ob der Syntheseschritt erfolgreich beendet wurde. Klicken Sie auf "Cancel", um das Fenster zu schließen. Ein neues Menü *Open Synthesized Design* wird unter dem Tab *Synthesis* angezeigt. Klicken Sie auf *Schematic* und schauen Sie sich die schematische Darstellung des synthetisierten Designs an. Der erstellte Schaltplan sollte ähnlich zu dem in Abbildung 1.4 sein.

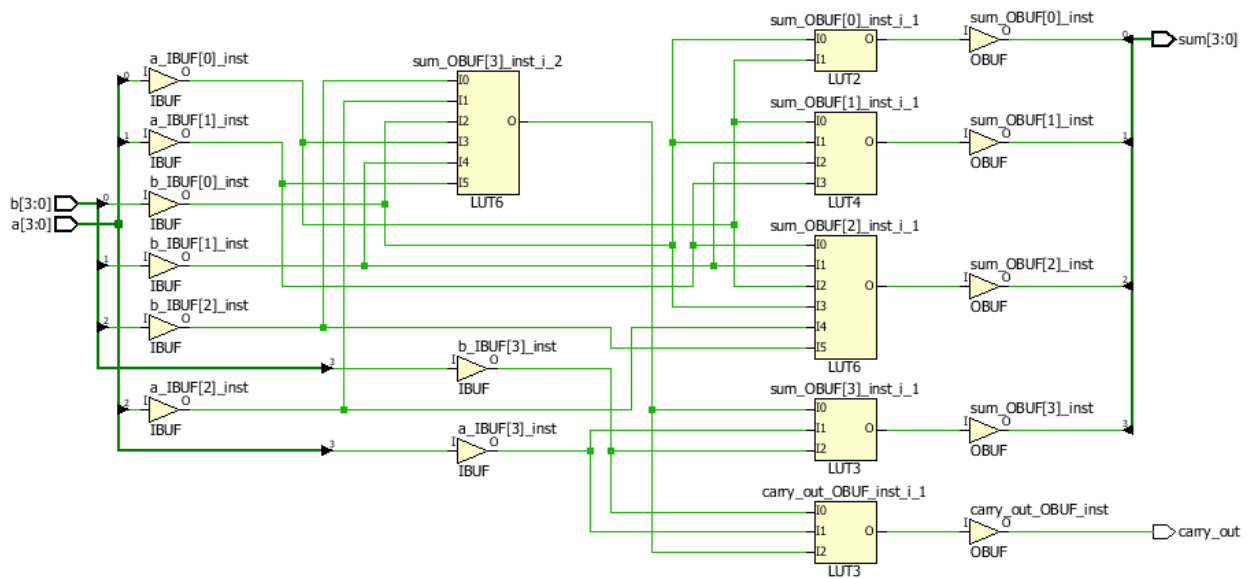


Abbildung 1.4. Schaltplan nach der Synthese.

- Führen Sie nun den *Implementations*-Schritt durch. Dabei wird das *Place & Route* des synthetisierten Designs auf dem Ziel-FPGA-Module ausgeführt. Klicken Sie auf *Flow Navigator* > *Implementation* > *Run Implementation*. Wenn der *Implementations*-Schritt erfolgreich abgeschlossen ist, wird ein neues Fenster geöffnet. Sie werden gefragt, ob Sie mit der *Bitstream Generation* fortfahren möchten.
- In dem geöffneten Fenster wählen Sie *Generate Bitstream* aus und klicken Sie auf *Ok*. Das Tool wird eine Datei **adder.bit** erzeugen, die für die Programmierung des FPGA verwendet

wird. Ein neues Fenster öffnet sich und informiert Sie, ob *Bitstream Generation* erfolgreich abgeschlossen wurde.

5. Bevor Sie fortfahren, prüfen Sie, ob der Netzschalter an der Nexsys-4 Board eingeschaltet ist und ob alle Schalter auf der Platine in der Nullstellung sind. Wählen Sie nun 'Open Hardware Manager' im neu geöffneten Fenster aus und klicken Sie auf *OK*. Das Fenster *Hardware-Manager* öffnet sich und zeigt dass noch kein *Hardware Target* geöffnet wurde. Klicken Sie auf den *Open Target* Link auf dem grün-farbigem Balken des Fensters *Hardware-Manager* und wählen Sie *Auto Connect*. Das Programm findet das Zielgerät automatisch und zeigt den Namen des Geräts im Fenster *Hardware*. Klicken Sie auf der Link *Program Device* an der Oberseite des Fensters *Hardware Manager* oder klicken Sie mit der rechten Maustaste auf den Gerätenamen und wählen Sie *Program Device* aus. Ein neues Fenster wird geöffnet. Die Bitstrom-Datei (*adder.bit*) und die Debug-Probes-Datei (*debug_nets.ltx*) werden im Fenster gezeigt. Klicken Sie auf *Program*. Der Addierer wird dadurch auf dem Zielmodul auf dem FPGA-Board realisiert.
6. Um das Addierer-Modul zu testen, ändern Sie die Werte der Eingangssignale durch Umschalten der Schalter SW8 bis SW15. Die Leds (LD0-LD4) sollten jetzt leuchten und das Ergebnis der Additionsooperation anzeigen. Ändern Sie die Eingangssignale und beobachten und prüfen Sie die korrekte Änderungen der **sum-** (LD3 zu LD0) und **Carry-Out**-Signale (LD4).