

# Übung 1 Erstes Projekt

Jan Grapengeter

May 1, 2019

## 1 Anlegen eines Projektes in Vivado

- Starten Sie Vivado
- Wählen Sie "Create New Project"
- "Next"
- Project Name: Wählen Sie einen Namen für das Projekt und legen Sie einen Installationsordner für das Projekt auf Ihrer Festplatte an.
- "Next"
- Project Type: Wählen Sie "RTL Project" (voreingestellt), "Do not specify sources at this time" sollte nicht angewählt sein.
- "Next"
- Add Sources: Hier legen Sie ihre VHDL-Dateien an
- Klicken Sie auf "Create File"
- Wählen Sie VHDL als Dateityp (File type) und wählen Sie "main" als Namen der Datei, "File location" sollte auf "Local to Project" stehen (voreingestellt).
- "Next"
- "Add Existing IP" kann ignoriert werden. - "Next"
- Add Constraints: Hier legen Sie ein Constraints-File an, dessen Bedeutung wird später erklärt
- Wählen Sie "Create File"
- Wählen Sie XDC als Dateityp (File type) und wählen Sie "cons" als Namen der Datei, "File location" sollte auf "Local to Project" stehen (voreingestellt)
- "Next"
- Default Part: Hier wählen Sie den von Ihnen verwendeten FPGA-Chip aus
- Wählen Sie "xc7atcpg236-1" aus, dies ist die Bezeichnung des FPGA-Chips auf dem Basys3.
- New Project Summary: Hier wird eine Zusammenfassung des gerade angelegten Projektes angezeigt. Wenn Sie alles richtig gemacht haben, sollte diese aussehen wie folgt:

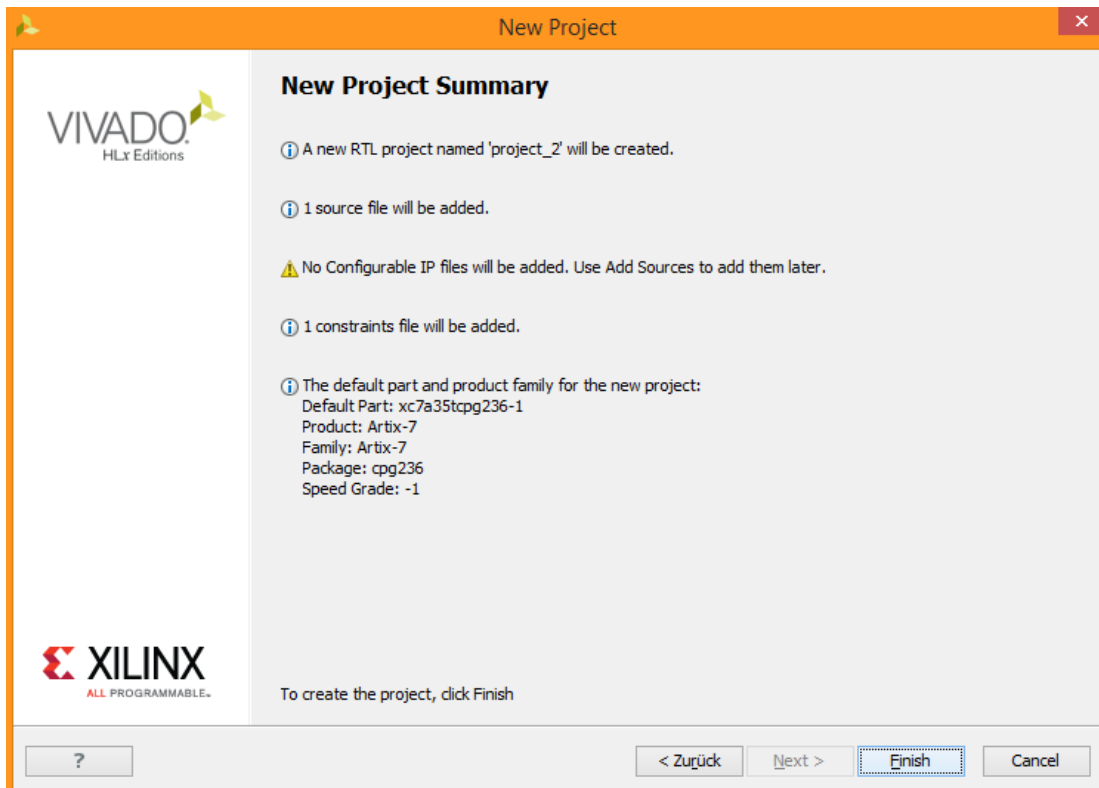


Figure 1: Projektzusammenfassung

- "Finish"

- Define Module: Hier tragen Sie die Ein- und Ausgänge ihres Projektes ein. Über diese Ports werden später Signale in das FPGA hinein und hinausgeführt.

- Wählen Sie "main" als "Entity name" und "Behavioral" als "Architecture Name" (voreingestellt).

- Unter "Port Name" tragen Sie "sw0" als Namen ein

- "Direction" sollte auf "in" stehen und "Bus" nicht aktiviert sein (voreingestellt).

- Klicken Sie auf das grüne Kreuz, um einen weiteren Port hinzuzufügen, tragen Sie "ld0" als Namen ein und ändern Sie "Direction" auf "out", Bus sollte nicht aktiviert sein.

- "Ok"

- Wenn Sie alles richtig gemacht haben, sollten Sie nun im Projekt Manager sein und folgendes Bild sehen:

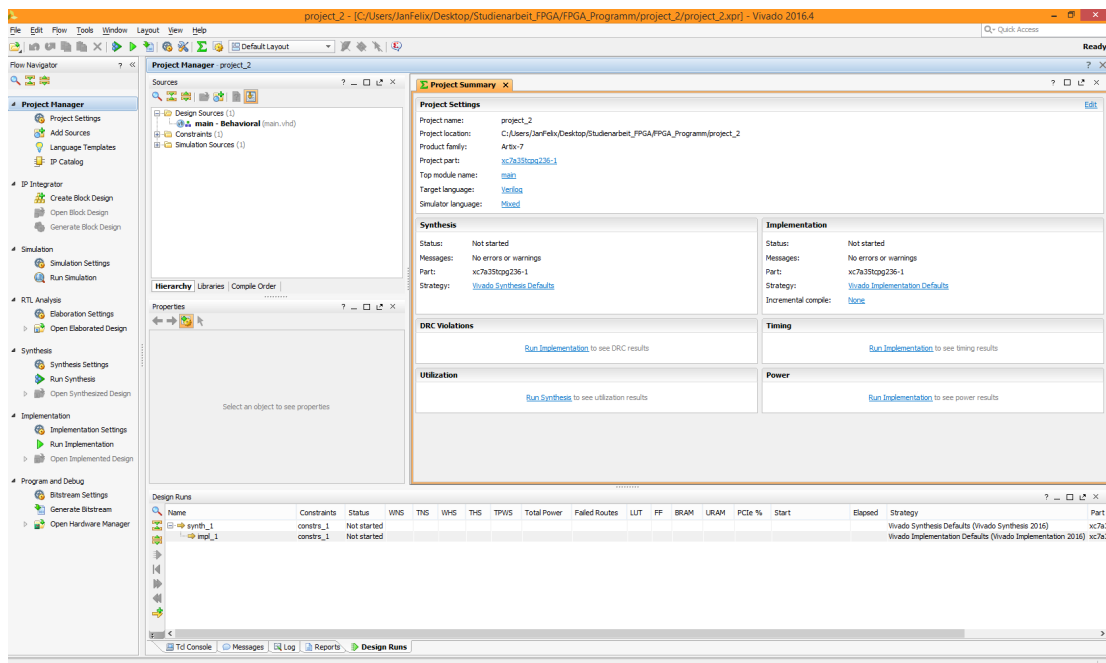


Figure 2: Projektmanager

-Damit haben Sie das Projekt erfolgreich angelegt.

## 2 Anlegen der VHDL-Main

-Doppelklicken Sie auf "main - Behavioral(main.vhd)" unter "Sources" (sollte orange umrandet sein)

-Im nun geöffneten Fenster "main.vhd" sollten Sie Ihre Main-Datei mit den beiden von Ihnen angelegten Ports sehen.

```

22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity main is
35     Port ( sw0 : in STD_LOGIC;
36           ld0 : out STD_LOGIC);
37 end main;
38
39 architecture Behavioral of main is
40
41 begin
42
43
44 end Behavioral;

```

Figure 3: VHDL main

-Unter "begin" tragen Sie:

`ld0<=sw0;`

ein. Damit wird der Wert des Eingangs "sw0" auf den Ausgang "ld0" geschrieben.

-Speichern Sie "main.vhd" ab.

### 3 Anlegend der Constraints-Datei

-Wieder im Reiter "Project Manager" doppelklicken Sie auf "Constraints"

-Doppelklicken Sie dann auf "cons.xdc", dies ist Ihre vorher angelegte Constraints-Datei.

-Sie sollten nun das noch leere Fenster "cons.xdc" sehen.

-Tragen Sie hier folgende Zeilen ein:

```

set_property -dict { PACKAGE_PIN V17    IOSTANDARD LVCMOS33 } [get_ports { sw0}];
set_property -dict { PACKAGE_PIN U16    IOSTANDARD LVCMOS33 } [get_ports { ld0}];

```

-Damit weisen Sie Ihren in der VHDL-Datei festgelegten Ein- und Ausgängen physikalische Ports auf dem FPGA-Chip zu. Hier wird dem Eingang "sw0" der Port "v17" zugewiesen und dem Ausgang "sd0" der Port "u16". Auf dem Basys3 sind die Ein- und Ausgänge des FPGA mit bestimmter Peripherie auf dem Board fest verdrahtet. Sie sollten, wenn Sie auf das Board

schauen, auch die Namen der Peripherie und der angeschlossenen Ports finden. "sw0" mit Port "v17" und "ld0" mit Port "u16" sind im rechten unteren Eck. (Im Bild bei Nummer 6)

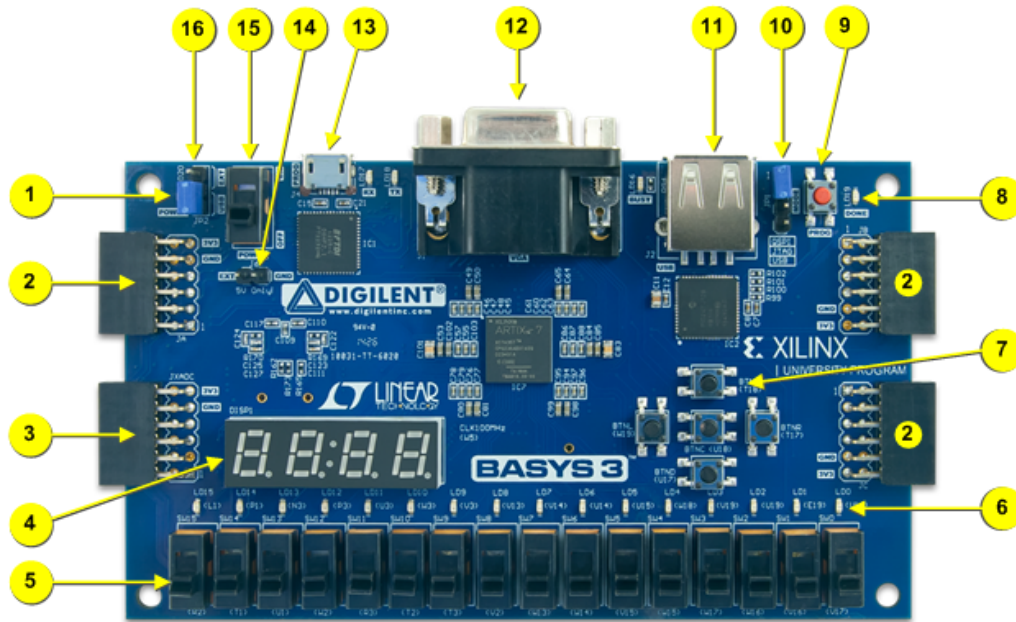


Figure 1. Basys3 board features

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod connector(s)	10	Programming mode jumper
3	Analog signal Pmod connector (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Figure 4: Basys3 Frontansicht

## 4 Programmierung des Boards

-Um nun Ihr Projekt auf das Board zu überspielen, klicken Sie in der rechten Spalte (Flow Navigator) auf "Generate Bitstream", damit wird aus Ihrem Projekt ein für das FPGA lesbarer Code gemacht.

-No Implementation Results available: "Yes"

-Launch runs:

-Wählen Sie "Default directory" als "Launch directory"

-Wählen Sie "Launch runs on local host". Unter "Number of Jobs" können Sie die Anzahl der Prozessorkerne auswählen, die Vivado dafür einsetzen darf. Wählen Sie so viele wie möglich, da

dieser Vorgang ansonsten einige Minuten dauern kann, insbesondere bei größeren Projekten.

- In der rechten oberen Ecke können Sie die einzelnen Schritte sehen, die Vivado zur Bitstream-erstellung durchläuft.
- Bitstream Generation Completed:
- Wählen Sie "Open Hardware Manager"
- Unter "Flow Navigator" finden Sie nun ganz unten unter "Program and Debug" die Schaltfläche "Open Target", klicken Sie auf diese und wählen Sie "auto connect"
- Vivado sollte nun automatisch das angeschlossene Basys3 board finden.
- Wo vorher Ihr Reiter "Project Manager" war, sollte nun der "Hardware Manager" geöffnet sein.
- Hier können Sie das FPGA Chip-Modell sehen.
- Unter "Flow Navigator - Program and Debug" wählen Sie nun "Program Device" und klicken Sie auf "xc7a35t\_0" (Der Basys3 FPGA Chip).
- Program Device: Hier wählen Sie den von Ihnen generierten Bitstream aus.
- Da dieser automatisch ausgewählt sein sollte, klicken Sie auf "Program".
- Das FPGA sollte nun mit Ihrem VHDL Code programmiert worden sein.
- Testen Sie dies, indem Sie den Schalter 1 (sw0) umschalten, die LED0 (ld0) sollte an und ausgehen, abhängig von der Schalterposition.

## 5 Aufgaben

- 1.: Ändern Sie die Constraints so, dass nun Schalter "sw1" die Led "ld1" umschaltet. Nehmen Sie keine Änderungen an der VHDL Datei vor.
- 2.: Legen Sie in Ihrer Main zwei zusätzliche Ports an und verbinden Sie diese wie "sw0" und "ld0". Ändern Sie die constraints Datei danach so, dass Sie über den Knopf "BTND" (zu finden direkt oberhalb der LEDs) die LED 5 (ld5) umschalten können.
- 3.: Drücken Sie den Knopf "PROG" (zu finden rechts oben) auf Ihrem Board, dieser dient in der Standard-Stellung als Reset. Welches Programm läuft nun auf Ihrem FPGA? Was sagt Ihnen das über die interne Speicherstruktur eines FPGAs?