

# Anlegen eines Projekts in Quartus

Quartus starten

Menü File

New Project Wizard ...

Eingabe des Arbeitsverzeichnisses

New Project Wizard

**Directory, Name, Top-Level Entity**

What is the working directory for this project?

C:/intelFPGA/DE0-CV/VHDL

What is the name of this project?

bsp\_01

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

bsp\_01

Use Existing Project Settings...

< Back Next > Finish Cancel Help

Eingabe des Projektnamens (Name der ENTITY)

⇒ Button NEXT

Project Type: Empty Project

⇒ Button NEXT

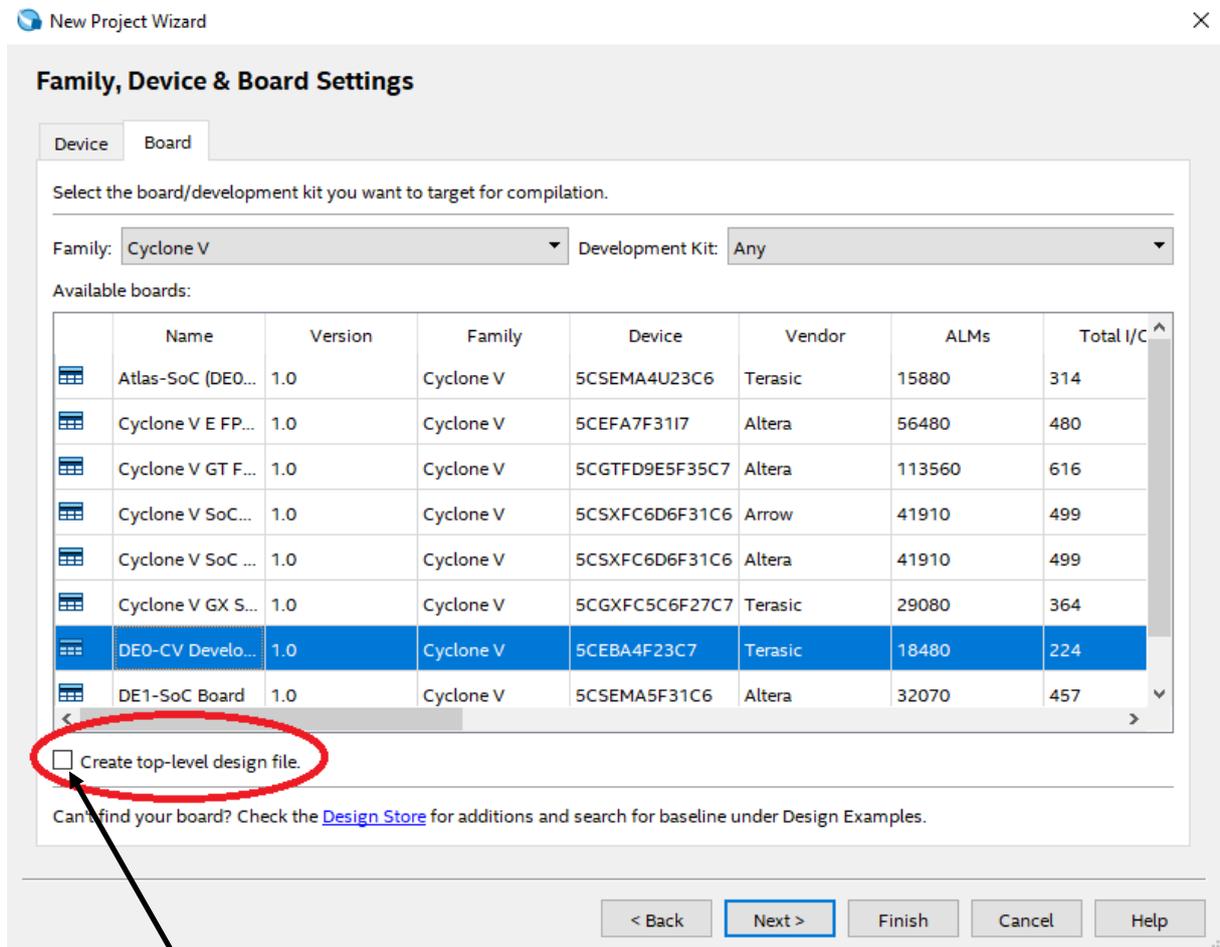
Add Files: keine Eingabe

⇒ Button NEXT

## Family, Device &amp; Board Settings

Registerkarte BOARD

„DE0-CV Development Board“ auswählen



**WICHTIG: Häkchen bei „Create Top-level design file“ entfernen**

⇒ Button NEXT

EDA Tool Settings: keine Eingabe

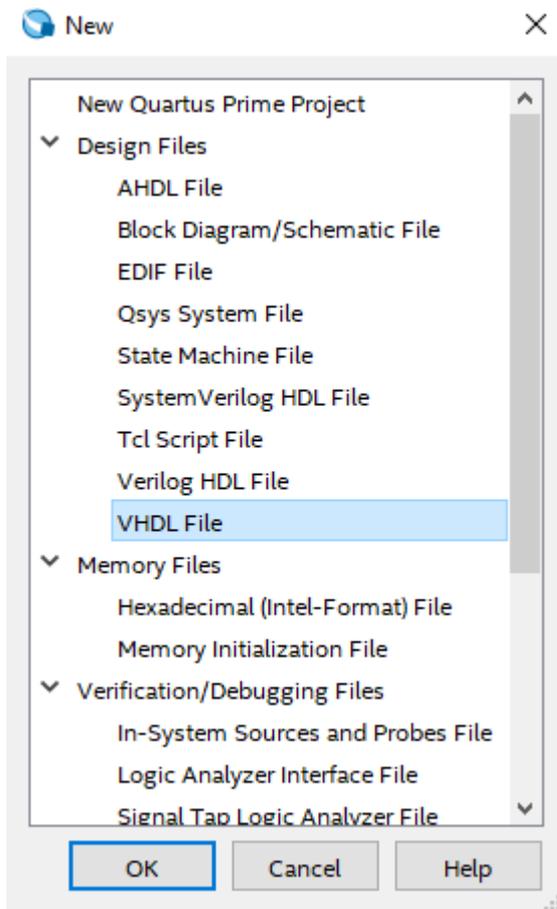
⇒ Button NEXT

⇒ Finish

Quartus legt nun das neue Projekt an

## Menü FILE

New ...



Button OK

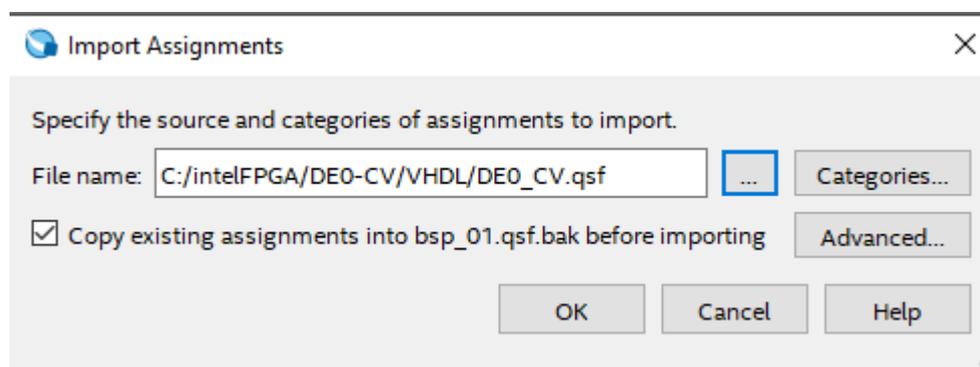
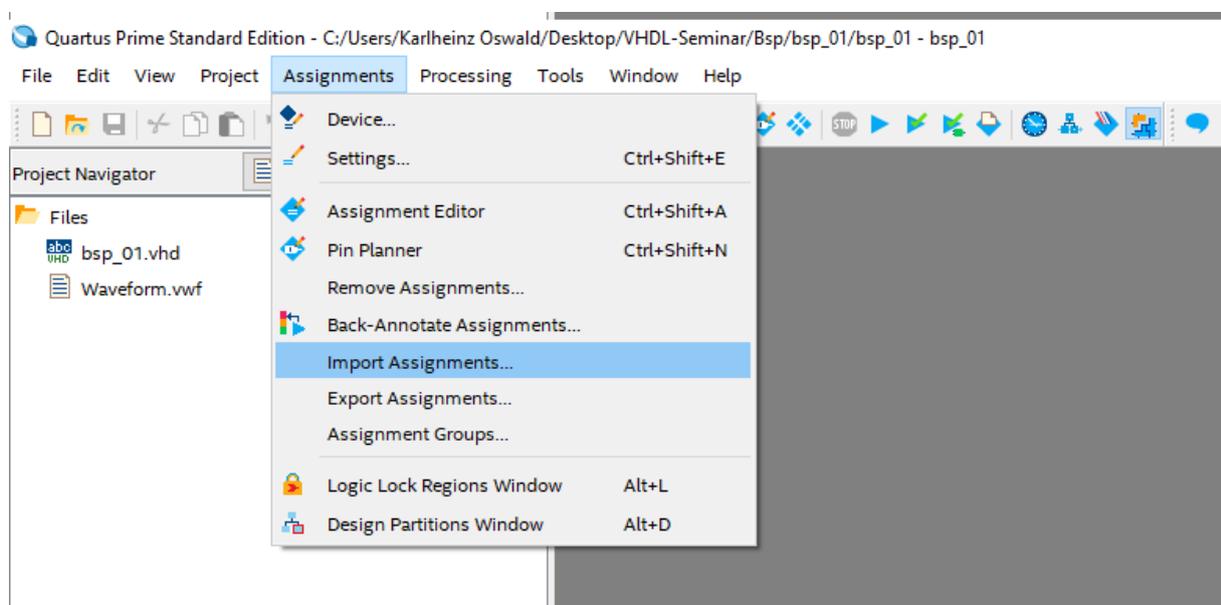
Dann öffnet Quartus den Texteditor und ist bereit zur Eingabe

Nach dem Eingeben von **LIBRARY**, **ENTITY** und **ARCHITECTURE** ist der nächste Schritt das Vorbereiten für das Übersetzen des eingegebenen Codes.

Mit der Auswahl des Boards haben wir bereits den richtigen FPGA-Baustein (5CEBA4F23C7) ausgewählt.

Fehlt noch die Zuordnung der Pins des Bausteins zu Signalnamen, unter denen die angeschlossenen Komponenten am Board angesprochen werden können.

Dazu können wir ein vorbereitetes File DE0\_CV.qsf importieren:



Den Pfad entsprechend dem Speicherort des Files anpassen.

**Clock-Signale, f = 50MHz**

CLOCK\_50

CLOCK2\_50

CLOCK3\_50

CLOCK4\_50

**Signale für das DRAM**

DRAM\_ADDR[12..0]

DRAM\_BA[1..0]

DRAM\_CAS\_N

DRAM\_CKE

DRAM\_CLK

DRAM\_CS\_N

DRAM\_DQ[15..0]

DRAM\_LDQM

DRAM\_RAS\_N

DRAM\_UDQM

DRAM\_WE\_N

**Signale der beiden Stiftleisten**

GPIO\_0[35..0]

GPIO\_1[35..0]

**Signale der 6 Sieben-Segment-Anzeigen**

HEX0[6..0]

HEX1[6..0]

HEX2[6..0]

HEX3[6..0]

HEX4[6..0]

HEX5[6..0]

**Signale der 5 Taster**

KEY[3..0]

FPGA\_RESET\_N

**Signale der roten LEDs**

LEDR[9..0]

**Signale der Micro-SD-Karte**

SD\_CLK

SD\_CMD

SD\_DATA[3..0]

**Signale der PS2-Buchse**

PS2\_CLK

PS2\_DAT

PS2\_CLK2

PS2\_DAT2

**Signale der 10 Schiebeschalter**

SW[9..0]

**Signale der 9pol. HD SubD-Buchse (VGA)**

VGA\_B[3..0]

VGA\_G[3..0]

VGA\_R[3..0]

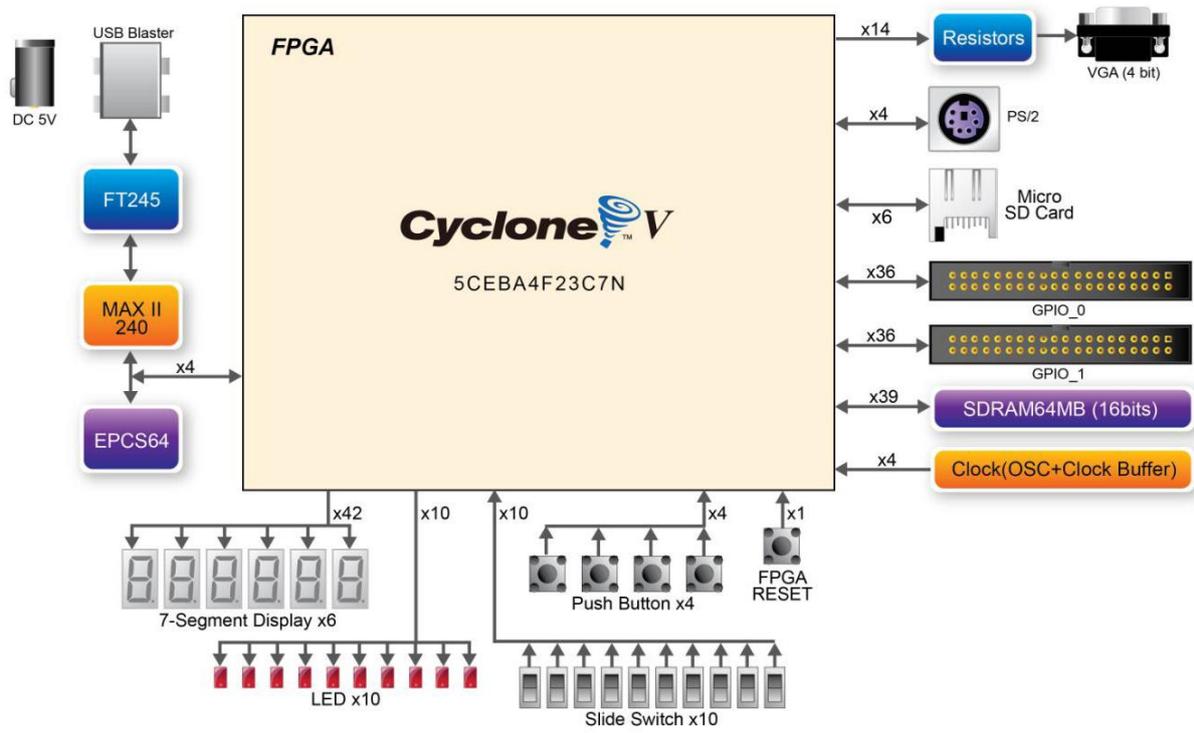
VGA\_HS

VGA\_VS

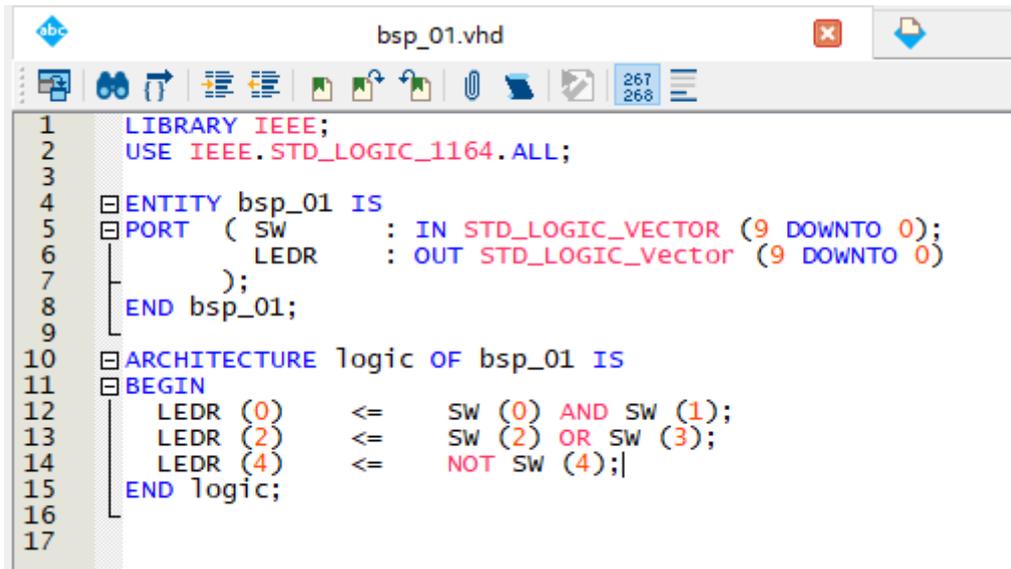
Für das erfolgreiche Funktionieren eines Projekts für das TERASIC DE0-CV Board unter Verwendung des Files DE0\_CV.qsf **MÜSSEN** die oben angeführten Signalnamen verwendet werden!



**Blockdiagramm des TERCASIC DE0-CV Boards**

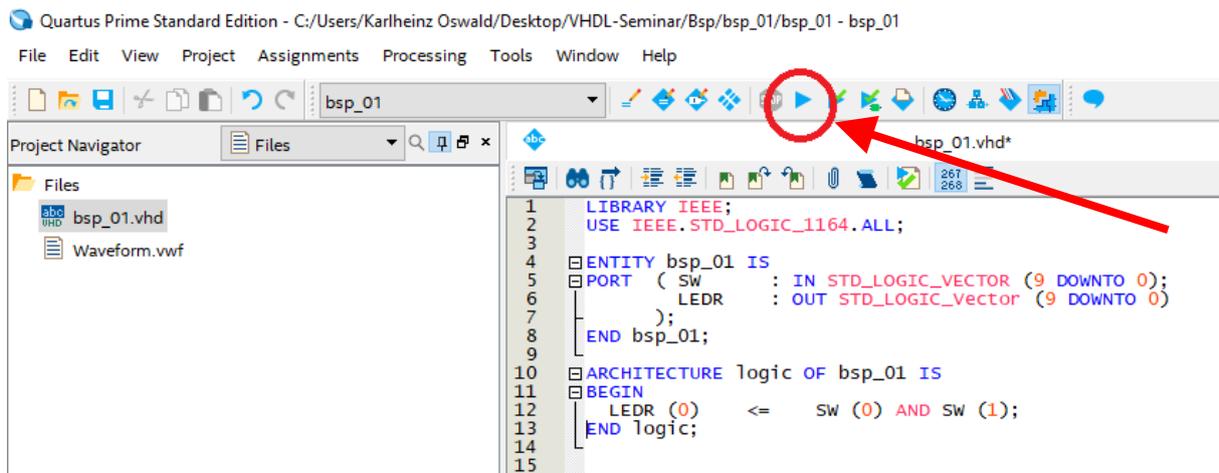


## Erstes Projekt:



```
1  LIBRARY IEEE;
2  USE IEEE.STD_LOGIC_1164.ALL;
3
4  ENTITY bsp_01 IS
5  PORT ( SW      : IN STD_LOGIC_VECTOR (9 DOWNTO 0);
6        LEDR     : OUT STD_LOGIC_Vector (9 DOWNTO 0)
7        );
8  END bsp_01;
9
10 ARCHITECTURE logic OF bsp_01 IS
11 BEGIN
12     LEDR (0)  <=  SW (0) AND SW (1);
13     LEDR (2)  <=  SW (2) OR SW (3);
14     LEDR (4)  <=  NOT SW (4);
15 END logic;
16
17
```

## Compilieren



Starten des Compilers geschieht mit Hilfe des oben gezeigten Icons.

Bei fehlerfreiem Durchlauf kann das Projekt in den Baustein geladen werden.