Lauflicht

Inhalt

[2 Abbildungsverzeichnis 1](#_Toc8601147)

[3 Allgemeines 2](#_Toc8601148)

[4 Hardware - Schnittstellen 3](#_Toc8601149)

[4.1 Lauflicht PINS 3](#_Toc8601150)

[5 Software Schnittstelle 4](#_Toc8601151)

[5.1.1 Registerbeschreibung 4](#_Toc8601152)

[6 Implementierung 6](#_Toc8601153)

[6.1 Erstellen des Projekts in Quartus 6](#_Toc8601154)

[6.2 Pin-Belegung FPGA 8](#_Toc8601155)

[6.3 Komponenten des Lauflichts 9](#_Toc8601156)

[6.3.1 Übersicht der einzelnen Komponenten des Lauflichts 9](#_Toc8601157)

[6.3.2 TOP-Entity (Lauflicht) 9](#_Toc8601158)

[6.3.3 I2C\_Target 10](#_Toc8601159)

[6.3.4 I2C\_Shell 13](#_Toc8601160)

[6.3.5 Light 15](#_Toc8601161)

[6.3.6 Top Level Entity 16](#_Toc8601162)

[7 Synthese Ergebnis 16](#_Toc8601163)

[7.1.1 Schreib/Lese Zyklen der I2C – Avalon - Bridge 18](#_Toc8601164)

[8 Testbench 19](#_Toc8601165)

# Abbildungsverzeichnis

[Abbildung 3 Toplevel Blockschaltbild Toplevel design 9](#_Toc8510718)

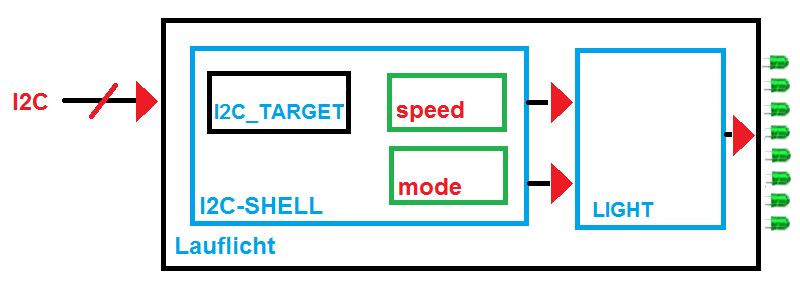
[Abbildung 5 Konfiguration der I2C Avalon Bridge 11](#_Toc8510719)

[Abbildung 6 I2C Bus Zyklus 18](#_Toc8510720)

# Allgemeines

Entwicklung eines Lauflichts welches über I2C konfiguriert werden kann.

* 8 LED’s
* Die Lauflicht Geschwindigkeit kann in 16 Stufen von 0,1s Schrittdauer bis 1,6s Schrittdauer eingestellt werden
* Konfiguration von 8 verschiedenen Lauflicht Mustern.

****

I2C\_TARGET wird mit Hilfe des Quartus IP-Catalogs erstellt.

# Hardware - Schnittstellen

## Lauflicht PINS

FPGA / CYC1000 TRENZ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | DIR | FPGA-PIN | BOARD ANSCHUSS | Beschreibung |
| res\_n\_i | in | N6 | - | System Reset |
| clk\_i | in | M2 | - | System Clock **12.5 MHz** |
|  | | | | |
| *I2C* | | | | |
| scl\_i | in | R14 | 32 | I2C Clock |
| sda\_io | in / out | T15 | 31 | I2C Daten |
|  | | | | |
| * *LED-OUTPUTS* | | | | |
| LED0 | out | M6 | PRINT LED | TEST LED |
| LED1 | out | T4 | PRINT LED | TEST LED |
| LED2 | out | T3 | PRINT LED | TEST LED |
| LED3 | out | R3 | PRINT LED | TEST LED |
| LED4 | out | T2 | PRINT LED | TEST LED |
| LED5 | out | R4 | PRINT LED | TEST LED |
| LED6 | out | N5 | PRINT LED | TEST LED |
| LED7 | out | N3 | PRINT LED | TEST LED |

ESP32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | DIR | PIN | ESP - BOARD | Beschreibung |
| I2C INTERFACE | | | | |
| scl\_o | out | SCL / 22 | PIN 32 | I2C Clock |
| sda\_io | in/out | SDA / 21 | PIN 31 | I2C DATA |
| CONFIGS | | | | |
| CONF\_1 | in | A7 / 35 | PIN 8 | SE SW SPEC |
| CONF\_2 | in | A18 / 25 | PIN 9 | SE SW SPEC |
| CONF\_3 | in | A19 / 26 | PIN 10 | SE SW SPEC |
|  |  |  |  |  |

# Software Schnittstelle

### Registerbeschreibung

#### SPEED- Register

I2C Register Address: 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Bit 7* | *Bit 6* | *Bit 5* | *Bit 4* | Bit(3:0) |
| *-* | *-* | *-* | *-* | speed(3:0) |
| *-* | *-* | *-* | *-* | 0: step-size: 0,1s  1: step-size: 0,2s  …  15: step-size:1,5s |
|  |  |  |  | Default: 0x02 |

#### MODE- Register

I2C Register Address: 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Bit 7* | *Bit 6* | *Bit 5* | *Bit 4* | *Bit 3* | Bit(2:0) |
| *-* | *-* | *-* | *-* | *-* | MODE(2:0) |
| *-* | *-* | *-* | *-* | *-* | Mode 0 … Mode 7 |
|  |  |  |  |  | Default: 0x0 |

|  |  |
| --- | --- |
| *MODE* | Bit(2:0) |
| *0* | 0 0 0 0 0 0 0 1  0 0 0 0 0 0 1 0  …  1 0 0 0 0 0 0 0 |
| *1* | 0 0 0 0 0 0 1 1  0 0 0 0 0 1 1 0  …  1 1 0 0 0 0 0 0 |
| *2* | 1 0 0 0 0 0 0 1  0 1 0 0 0 0 1 0  0 0 1 0 0 1 0 0  0 0 0 1 1 0 0 0  1 0 0 0 0 0 0 1  0 1 0 0 0 0 1 0  … |
| *3* | 0 1 0 1 0 1 0 1  1 0 1 0 1 0 1 0 |
| *4* | 1 1 0 0 1 1 0 0  0 1 1 0 0 1 1 0  0 0 1 1 0 0 1 1 |
| *5* | - |
| *6* | - |
| *7* | - |

# Implementierung

## Erstellen des Projekts in Quartus

Projektverzeichnis: <…/Lauflicht\_cyc100/synthese>

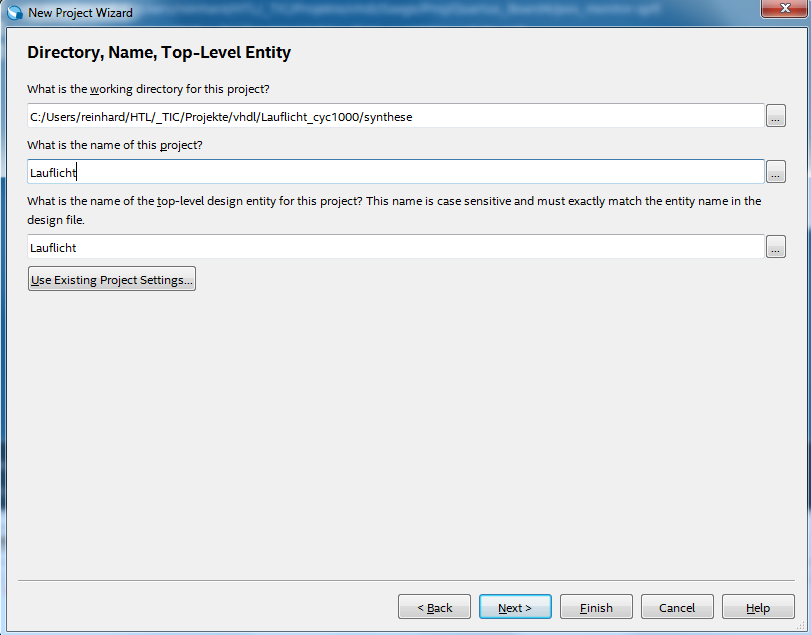


Abbildung 1 Synthese Verzeichnis + Top-Level Entity

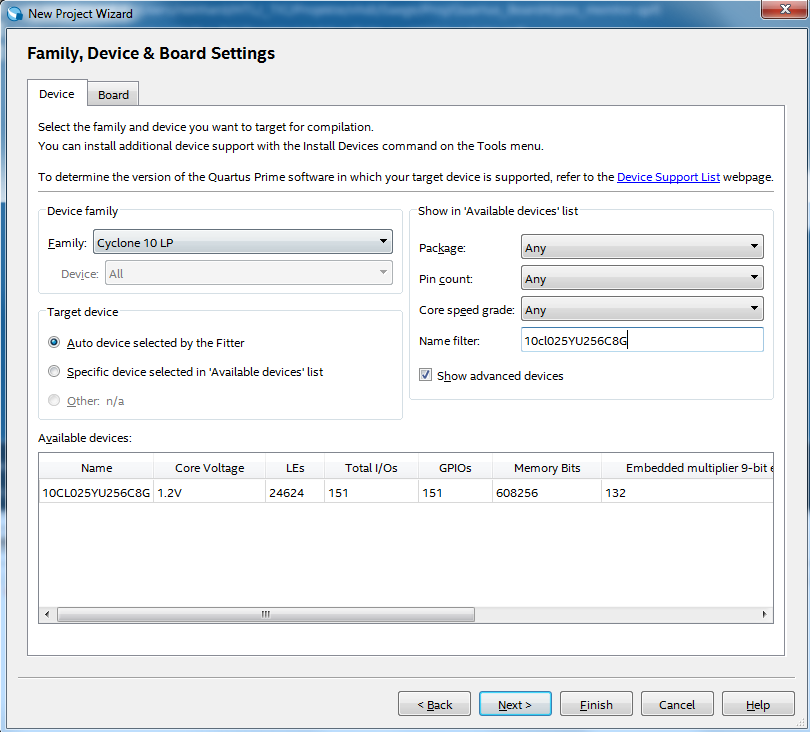


Abbildung 2 Auswahl des FPGA für das TRENZ Board CYC100 <10CL025YU256C8G>

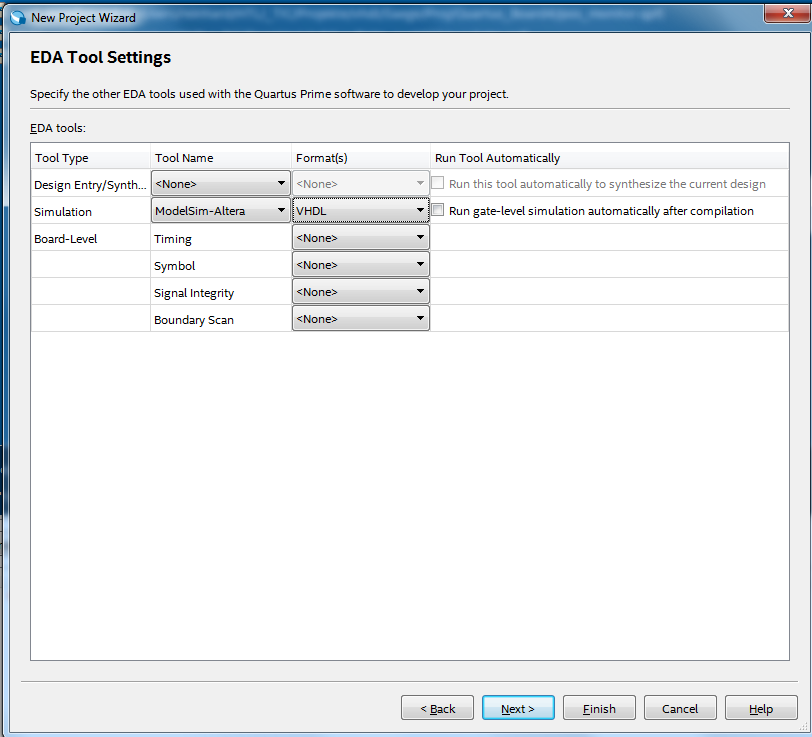


Abbildung 3 Simulation-Einstellung: Modelsim Altera

Summary

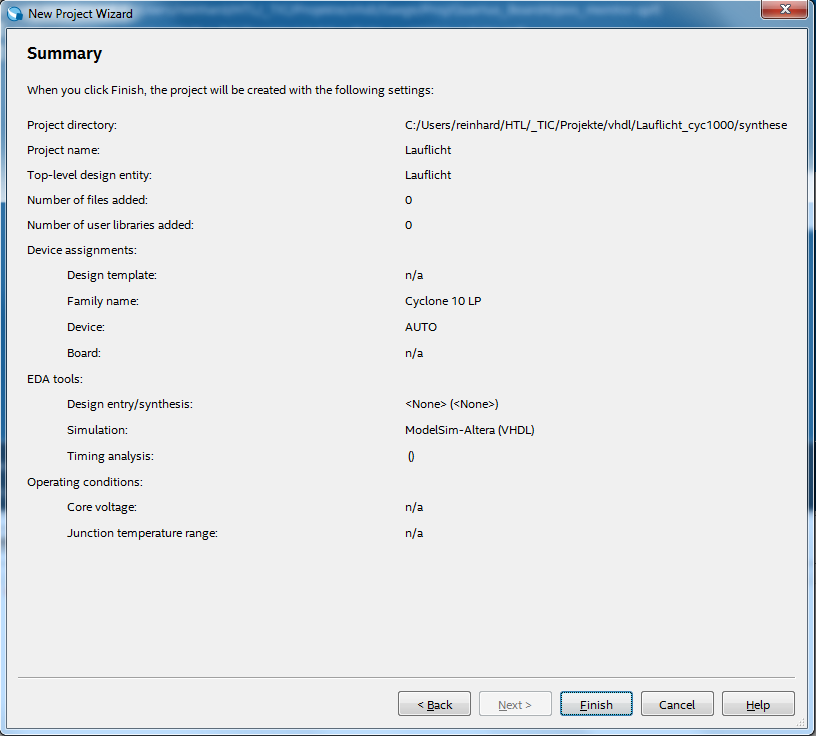


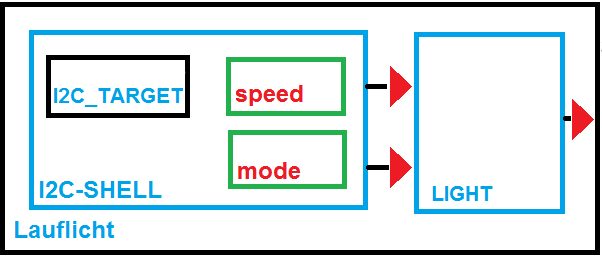
Abbildung Summary

## Pin-Belegung FPGA

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | DIR | FPGA-PIN | Beschreibung |
| res\_n\_i | in | N6 | System Reset |
| clk\_i | in | M2 | System Clock **12.5 MHz** |
|  | | | |
| scl\_i | in | R14 | I2C Clock |
| sda\_io | in / out | T15 | I2C Daten |
|  | | | |
| LED0 | out | M6 | TEST LED |
| LED1 | out | T4 | TEST LED |
| LED2 | out | T3 | TEST LED |
| LED3 | out | R3 | TEST LED |
| LED4 | out | T2 | TEST LED |
| LED5 | out | R4 | TEST LED |
| LED6 | out | N5 | TEST LED |
| LED7 | out | N3 | TEST LED |

## Komponenten des Lauflichts

### Übersicht der einzelnen Komponenten des Lauflichts



* Top-Entity Lauflicht
* I2C\_TARGET (Erstellen eines I2C Slave Device mit Hilfe des Quartus IP CATALOG)
* I2C\_SHELL (Lauflicht Register über I2C schreibbar/lesbar)
* Light (LED Signalgenerierung)

### TOP-Entity (Lauflicht)

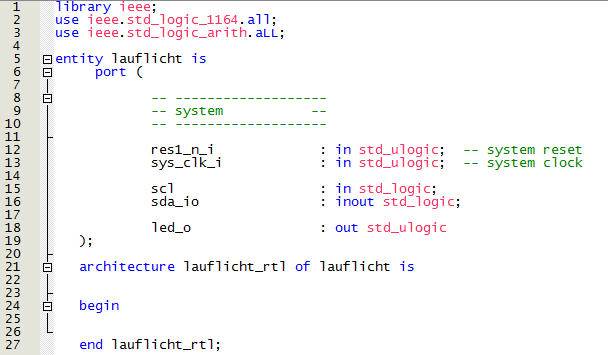
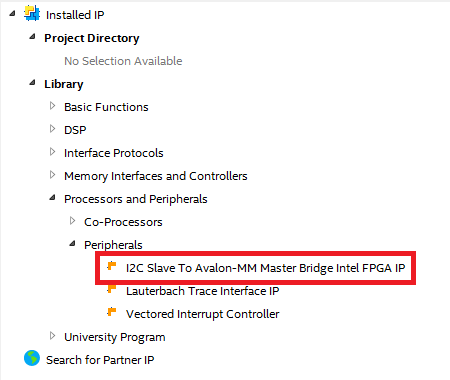


Abbildung Signale der TOP-LEVEL Entity

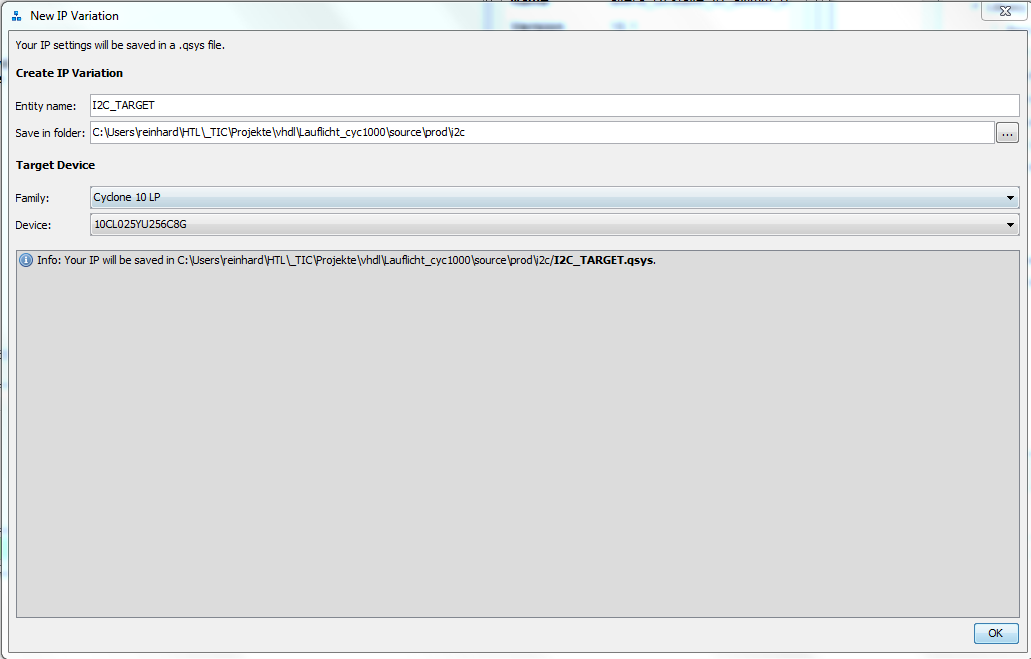
### I2C\_Target

#### Erstellen der I2C-Avalon Bridge (Quartus IP Katalog)

Die Lauflicht Register werden über die I2C Schnittstelle beschrieben bzw. sind auch über I2C Lesbar (Erweiterung). Ab Quartus 16.X wird die I2C-Memory master – Avalon Bridge angeboten.



* Verzeichnis und FPGA festlegen
* **Es sollte die Bridge in einem eigenen Verzeichnis erstellt werden, um die Übersichtlichkeit des Projekts nicht zu gefährden.**



* Erstellen der I2C - Avalon Memory Bridge (Slave!)

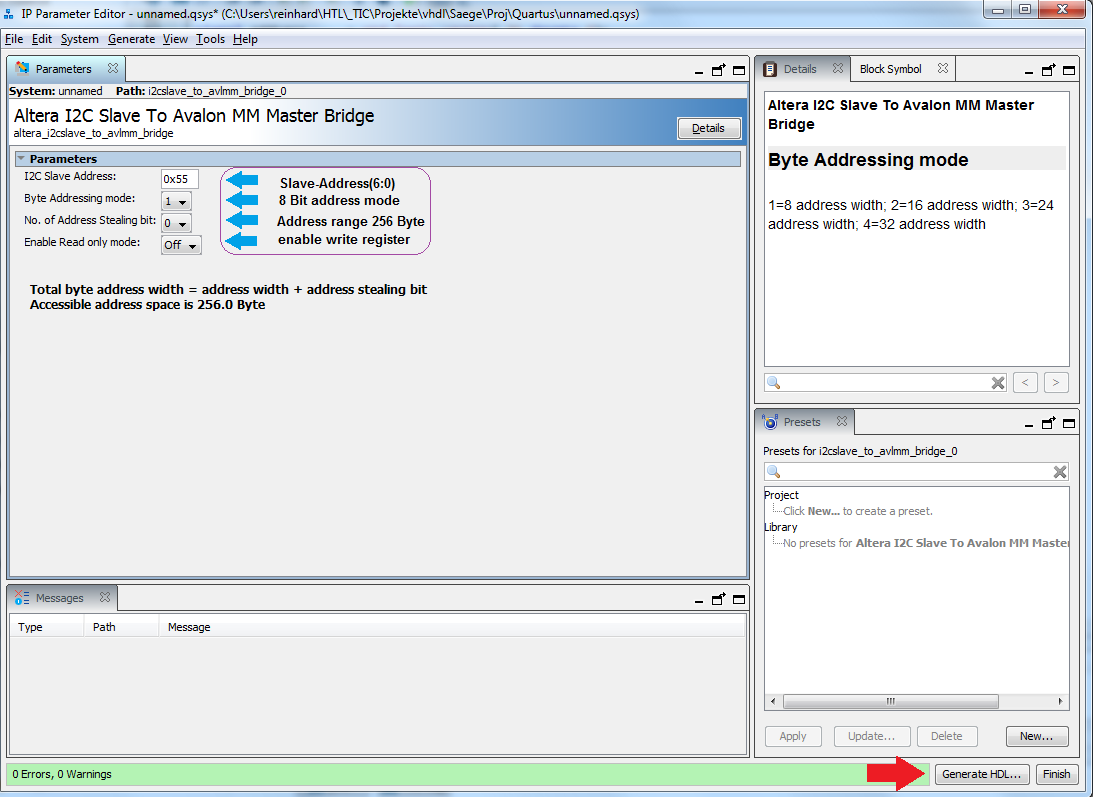


Abbildung Konfiguration der I2C Avalon Bridge

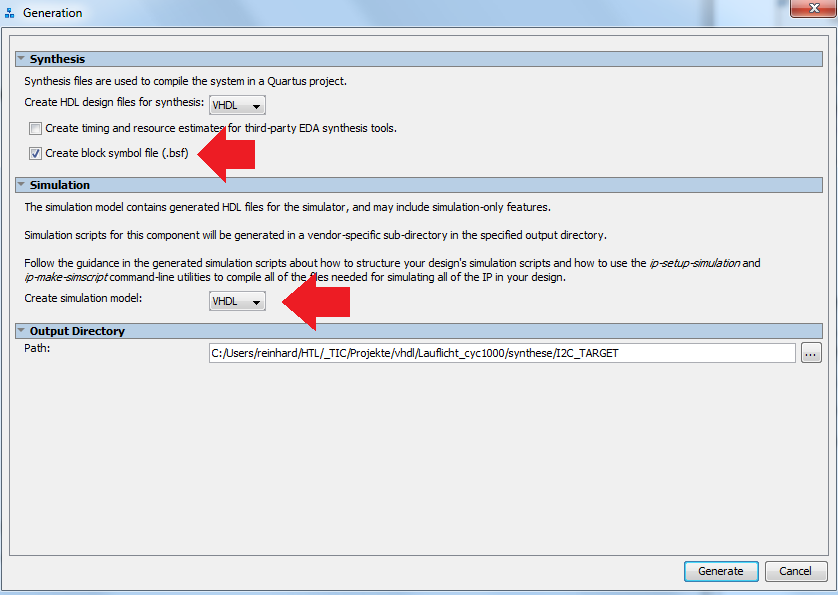
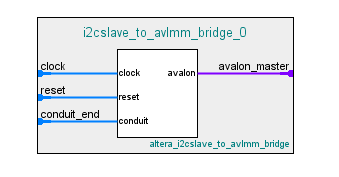


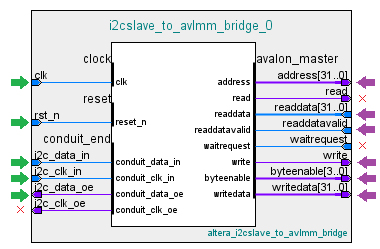
Abbildung Generieren eines Simulationsmodells

* Erstellte Komponenten



* Schnittstellen

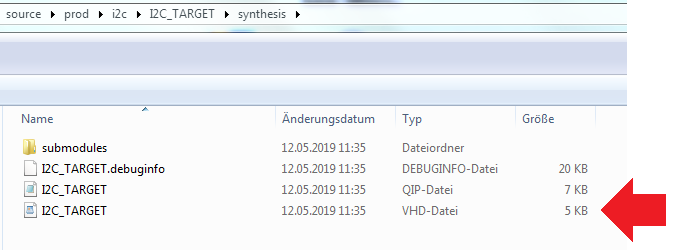
Für das Lauflicht benötigen wir die mit Pfeilen markierten Signale

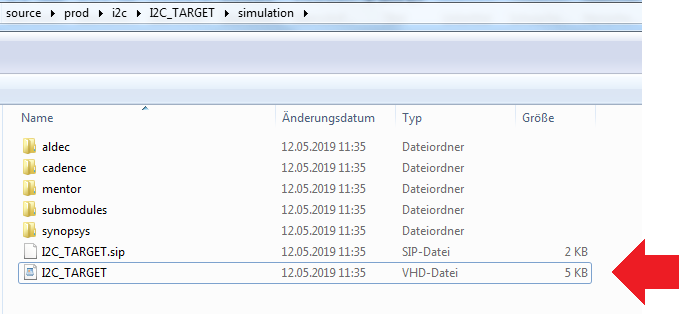


#### I2C Bridge zum Projekt hinzufügen.

Für die Synthese muss nur das File I2C\_TARGET.qsys includiert werden. In der I2C Shell (Entity I2C\_SHELL) wird die Komponente I2C\_TARGET hinzugefügt.

QSYS erzeugt folgendes vhdl File im Verzeichnis Synthese und Simulation:





In beiden Verzeichnissen befinden sich zwei identische Files <I2C\_TARGET.VHD>.



In den Verzeichnissen <submodules> können die verilog Files unterschiedlich sein. Quartus findet die erforderlichen submoduls selbstständig.

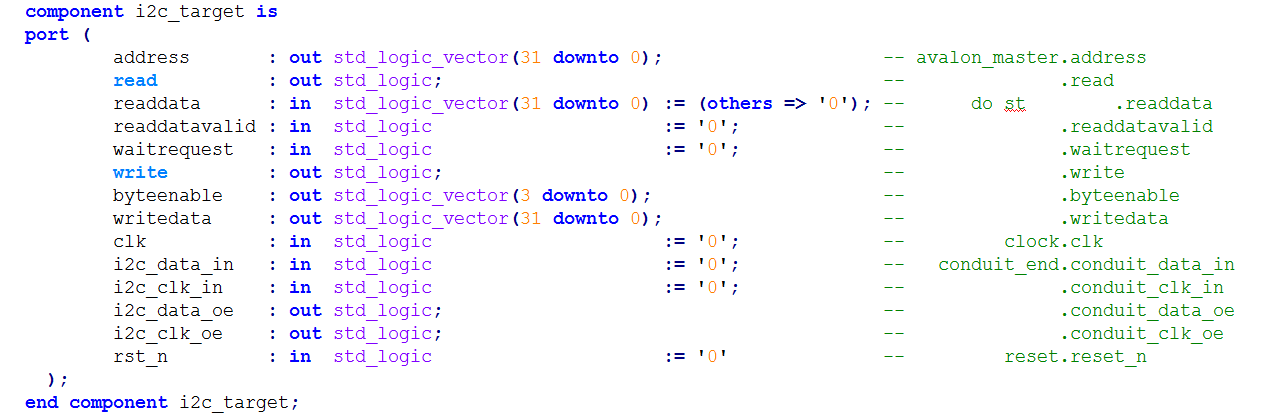
### I2C\_Shell

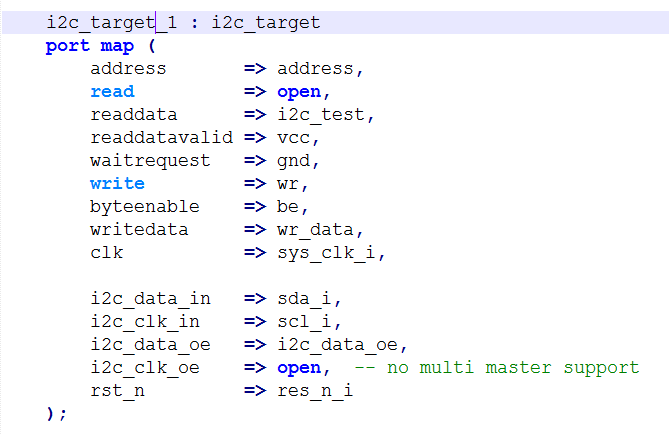
Funktion:

Schreiben / *Lesen* der Register speed(3:0), mode(3:0) über I2C. Die Shell verwendet keinen TRI Bus, generiert aber das Signal sda\_oe für den TRI Buffer in der Top Entity.

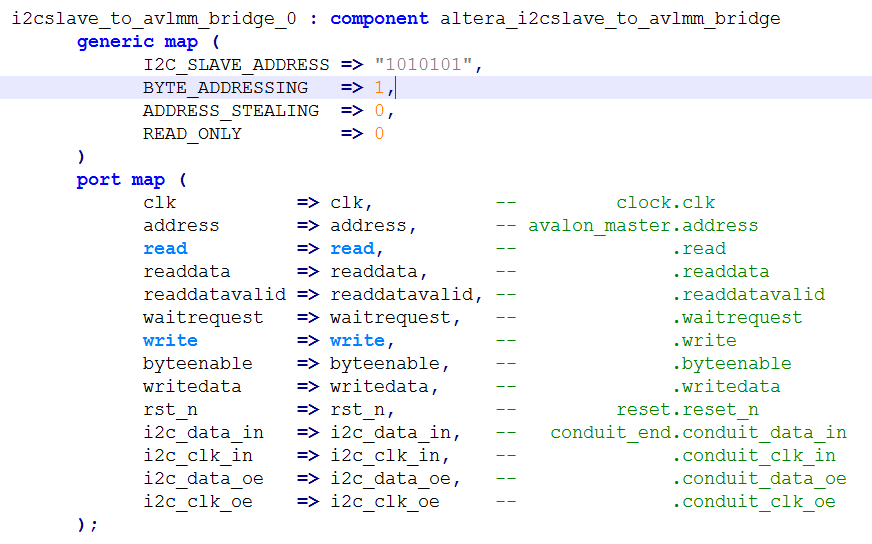
|  |  |  |
| --- | --- | --- |
| Signal | *DIR* | Beschreibung |
| res\_n\_i | *in* | System Reset |
| clk\_i | *in* | System Clock **12.5 MHz** |
|  | | |
| scl\_i | *in* | I2C Clock |
| sda\_i | *in* | I2C Empfangs Daten |
| sda\_o | *out* | I2C Sende Daten |
| sda\_oe | *out* | IC2 Data OE |
|  | | |
| mode\_o(3:0) | *out* | mode Register |
| *mode\_valid\_o* | *out* | *mode data valid* |
| speed\_o(3:0) | *in* | speed Register |
| *speed\_valid\_o* | *out* | *speed data valid* |

Die I2C\_SHELL verwendet die Komponente I2C\_TARGET

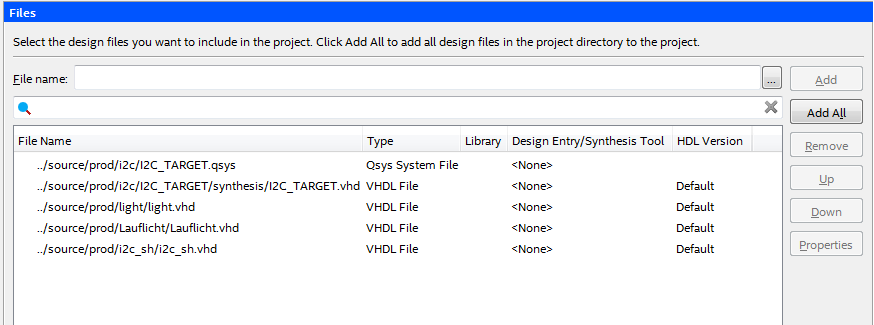




I2C\_TARGET instanziiert die altera\_i2cslave\_to\_avlmm\_bridge.



Für die Synthese müssen folgende Komponenten hinzugefügt werden:

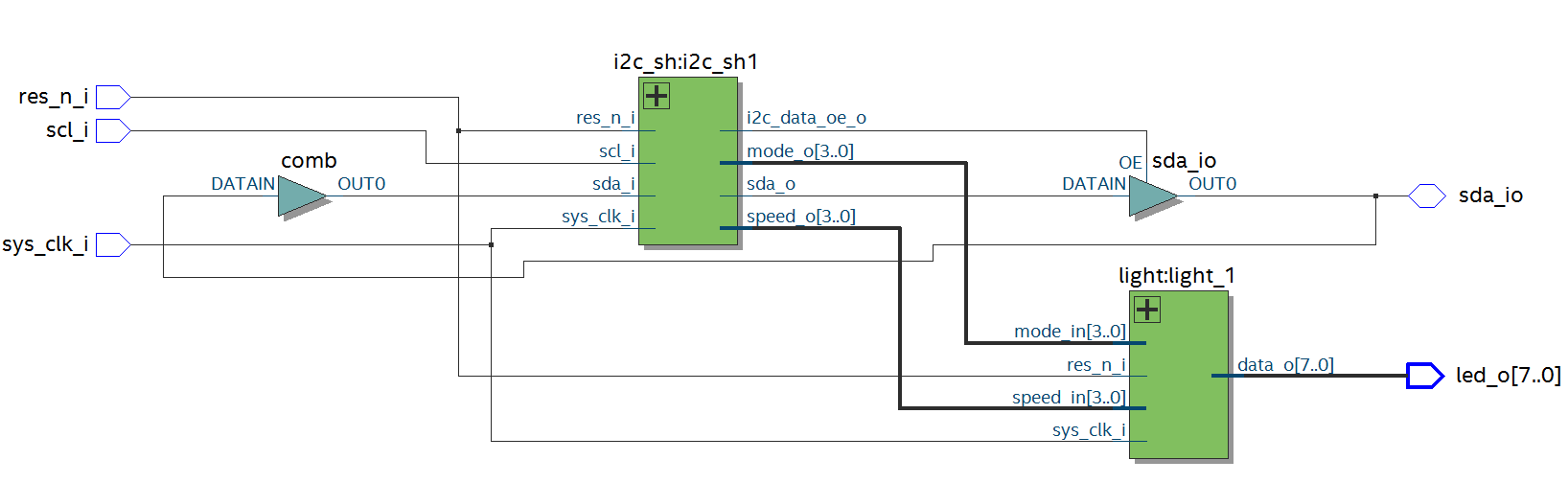


### Light

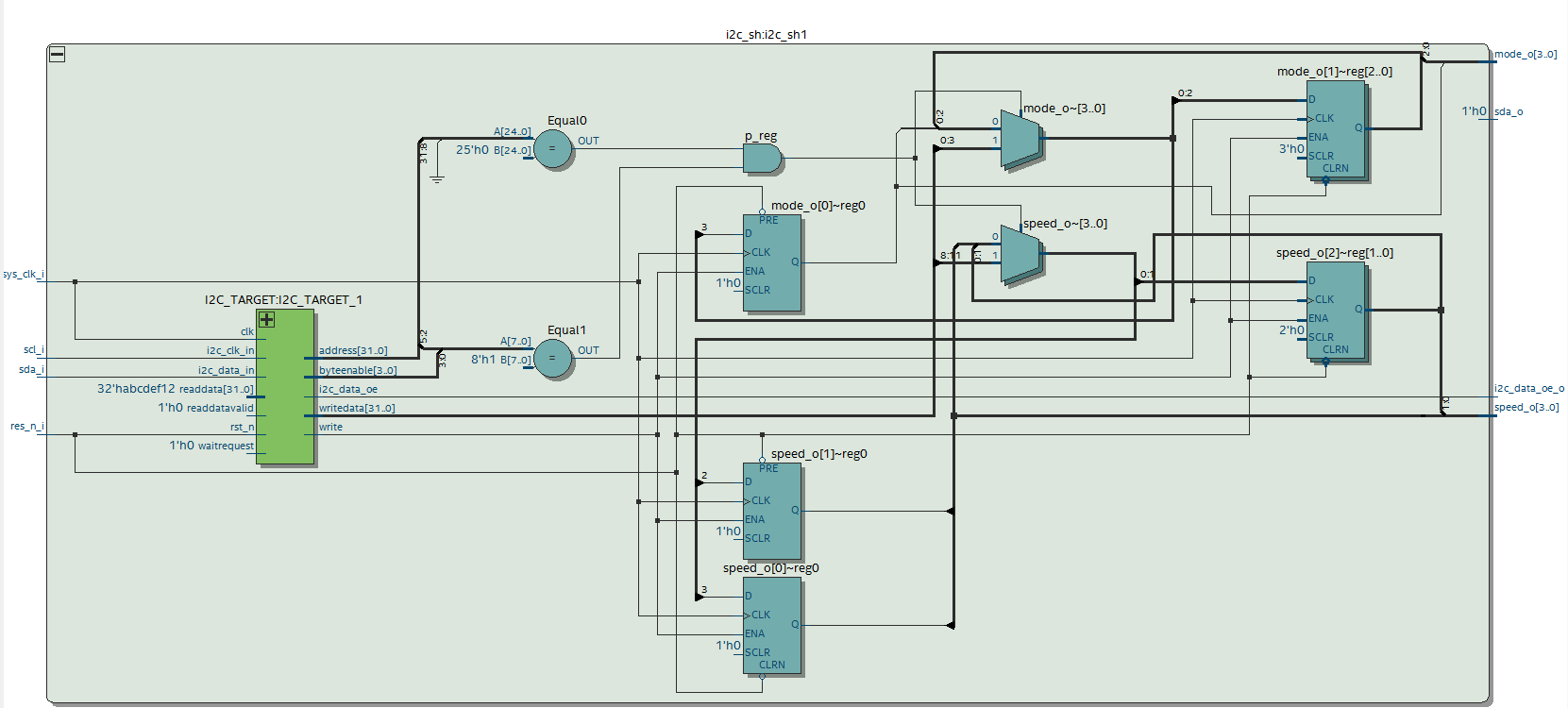
### Top Level Entity

# Synthese Ergebnis

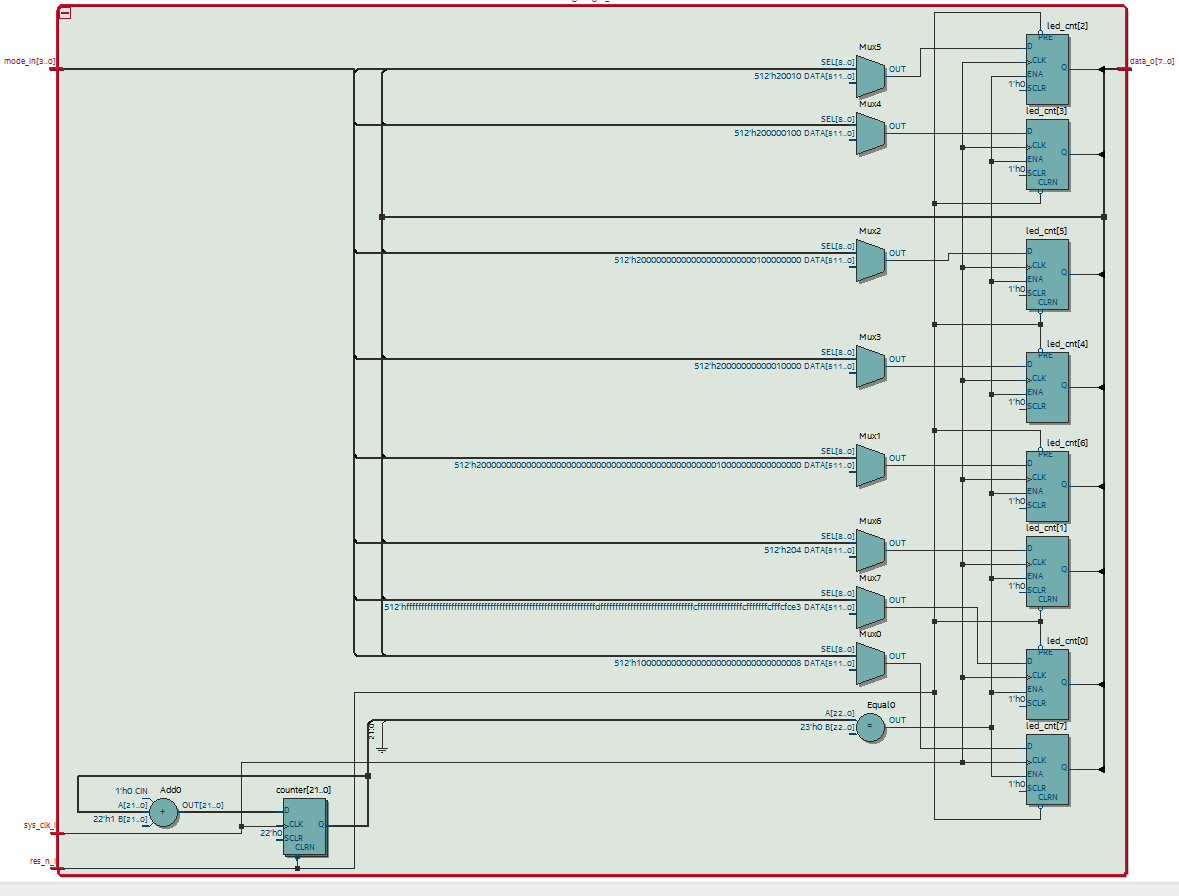
* Top Level (lauflicht)



* I2c\_shell



* Light



### Schreib/Lese Zyklen der I2C – Avalon - Bridge

Device Adresse(**6:0**): „101-0101“ (0x55)

* Schreiben: **Device\_ID(6:0)** & R/!W = 101 0101 & 0 = „0xAA“ (101 0101 0 = 0xAA)
* Lesen: **Device\_ID(6:0)** & R/!W = 101 0101 & 1 = „0xAB“ (101 0101 1 = 0xAB)

**Timing**

Abbildung I2C Bus Zyklus

# Testbench