

# DE2-115 Board Support Package

Version 0.2

## Kurzfassung

Dieses Board Support Package soll helfen, VHDL Projekte auf dem DE2-115 Board von Terrasic effizient zu starten und die Hardware-Umgebung schnell zu konfigurieren.

Dabei werden die folgenden Schritte empfohlen:

- 1.) Kopieren aller Verzeichnisse für dieses Board Support Package in das Projektverzeichnis.
- 2.) Umbenennen aller File-Namen mit „de2\_115\_start“ in einen Projekt-spezifischen Namen
 

orig_qsf_files/de2_115_start.qpf	Das Quartus II Projekt File
orig_qsf_files /de2_115_start.qsf	Das Quartus II Settings File
src/de2_115_start.p.vhd	Das Top-Level Package File
src/de2_115_start.e.vhd	Das Top-Level Entity File
src/de2_115_start.a_dummy.vhd	Das Top-Level Architektur File
src/de2_115_start.sdc	Das Synopsys Design Constraint File
sim/de2_115_start.tb.vhd	Ein Testbench File
- 3.) Öffnen ALLER Files aus den Verzeichnissen „orig\_qsf\_files“, „src“ und „sim“ in Notepad++
- 4.) Ersetzen von „de2\_115\_start“ durch den Projekt-spezifischen Namen in ALLEN geöffneten Dateien.
- 5.) Im Package File (src/<projekt\_name>.p.vhd) und im Entity File (src/<projekt\_name>.e.vhd) alle nicht benötigten Schnittstellen zur Hardware entfernen bzw. löschen.
- 6.) Im Quartus Setting File (orig\_qsf\_files//<projekt\_name>.qsf) alle „source“ Referenzen auf nicht benötigte Pin Files der nicht verwendeten Hardware Schnittstellen löschen oder auskommentieren.
- 7.) Alle Dateien in Notepad++ speichern und schliessen
- 8.) Alle Dateien aus dem Verzeichnis „orig\_qsf\_files“ in das Verzeichnis „quartus“ kopieren

Jetzt ist das Projekt aufgesetzt, und man kann es im Quartus Verzeichnis mit Doppel-Klick auf das .qpf File starten.

## Dokument-Historie

Version	Status	Datum	Verantwortlicher	Änderungsgrund
0.1	In Arbeit	3.11.2011	L. Arato	Start des Dokumentes
0.2	Release	1.3.2012	L. Arato	Rechtschreibfehler und Fusszeile korrigiert

## Änderungsberechtigte

Laszlo Arato	Institut EMS	NTB, Buchs
Dr. Urs Graf	Institut INF	NTB, Buchs

## Inhaltsverzeichnis

<b>Kurzfassung</b> .....	<b>1</b>
<b>Inhaltsverzeichnis</b> .....	<b>2</b>
<b>1 Einleitung</b> .....	<b>3</b>
1.1 Zweck des Dokuments .....	3
1.2 Gültigkeit des Dokuments .....	3
1.3 Begriffsbestimmungen und Abkürzungen .....	3
1.4 Zusammenhang mit anderen Dokumenten .....	3
<b>2 Verzeichnisse</b> .....	<b>3</b>
<b>3 Lizenz</b> .....	<b>4</b>
<b>4 Quartus Projekt File de2_start_115.qpf</b> .....	<b>5</b>
<b>5 Quartus Settings File de2_start_115.qsf</b> .....	<b>5</b>
<b>6 Pin Definition Files</b> .....	<b>7</b>
6.1 Pin Definitionen für System-Funktionen .....	7
6.2 Pin Definitionen für 7-Segment Display Pins .....	9
6.3 Pin Definitionen für Audio Codec Pins .....	10
6.4 Pin Definitionen für EEPROM Pins .....	10
6.5 Pin Definitionen für Ethernet Pins .....	11
6.6 Pin Definitionen für Extended IO Pins .....	11
6.7 Pin Definitionen für Flash Pins .....	11
6.8 Pin Definitionen für GPIO Pins .....	11
6.9 Pin Definitionen für GPIO auf HSMC Pins .....	11
6.10 Pin Definitionen für Infrarot Receiver Pin .....	11
6.11 Pin Definitionen für LCD Pins .....	12
6.12 Pin Definitionen für LVDS auf HSMC Pins .....	12
6.13 Pin Definitionen für PS2 Pins .....	12
6.14 Pin Definitionen für SD-Card Pins .....	12
6.15 Pin Definitionen für SDRAM Pins .....	13
6.16 Pin Definitionen für SRAM Pins .....	13
6.17 Pin Definitionen für TV Decoder Pins .....	13
6.18 Pin Definitionen für UART Pins .....	13
6.19 Pin Definitionen für USB Pins .....	13
6.20 Pin Definitionen für VGA Pins .....	13
<b>7 Definition der Design Files</b> .....	<b>14</b>
<b>8 SDC Synopsys Design Constraint File</b> .....	<b>14</b>
<b>9 Top-Level Package File</b> .....	<b>15</b>
<b>10 Top-Level Entity File</b> .....	<b>19</b>
<b>11 Top-Level Dummy Architektur File</b> .....	<b>20</b>
<b>12 Testbench File</b> .....	<b>21</b>

## 1 Einleitung

### 1.1 Zweck des Dokuments

Diese Einführung soll Studenten und anderen interessierten Personen helfen, möglichst schnell und effizient das DE2-115 Board von Altera / Terasic für FPGA Entwicklung zu nutzen.

### 1.2 Gültigkeit des Dokuments

Dieses Pflichtenheft ist für NTB internen Gebrauch.

### 1.3 Begriffsbestimmungen und Abkürzungen

FPGA	Field Programmable Gate Array, ein programmierbarer Logikbaustein.
ALTERA	Altera Corporation ist ein Hersteller von FPGAs
Quartus II	Die offizielle Software von Altera für CPLD und FPGA Entwicklungen
VHDL	„Very High-Speed Hardware Description Language“

### 1.4 Zusammenhang mit anderen Dokumenten






Folgende begleitende Dokumente sind geplant oder bereits in Arbeit:

- Einführung in ModelSim
- Einführung in VHDL Design
- Einführung in VHDL Testbench Design
- VHDL Design Guidelines

Weitere unterstützende Literatur:

- DE2\_115\_User\_Manual.pdf

## 2 Verzeichnisse

 doku	doku	für User Manual, Hardware und Projektbeschreibungen
 orig_qsf_files	orig_qsf_files	Back-up der Quartus-II Setting Files, vor Quartus selbst geschützt
 quartus	quartus	Das Quartus-II Arbeitsverzeichnis
 sim	sim	Verzeichnis für Testbenches, Waveforms und ModelSim .do Files
 src	src	Verzeichnis für alle Design-Files

### 3 Lizenz

Die in diesem Packet zusammengefassten Files werden unter der Boost-Software-Lizenz 1.0 zur Verfügung gestellt (BSL-1.0). Dabei wird für jede Person oder Organisation gratis die Erlaubnis gegeben, diese Software zu verwenden, zu verändern und weiterzuverteilen, solange die folgenden Auflagen erfüllt werden:

- Die Copyright Markierung, der Text der BSL Lizenz sowie der Hinweise auf die Ablehnung der Haftung müssen in jeder Kopie und abgeleiteten Arbeit enthalten sein
- Die Software wird so zur Verfügung gestellt, wie sie ist, ohne jegliche Garantie oder Haftung.

Massgebend ist der englische Originaltext der BSL-1.0 Lizenz:

#### **Boost Software License 1.0 (BSL-1.0)**

**Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:**

**The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.**

**THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.**

## 4 Quartus Projekt File de2\_start\_115.qpf

File: de2\_start\_115.qpf

Verzeichnis: orig\_qsf\_files

Beschreibung: Projekt Definition ... enthält im Wesentlichen nur den Projektnamen

```
QUARTUS_VERSION = "11.0"
DATE = "13:45:52 Oct 09, 2011"

# Revisions

PROJECT_REVISION = "de2_115_start"
```

## 5 Quartus Settings File de2\_start\_115.qsf

File: de2\_start\_115.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Quartus-II Projekt Einstellungen ... enthält ALLES über FPGA-Typ, Compiler-, Simulation- und Timing-Konfiguration, Liste der Design-Files und alle Pin-Definitionen.  
Durch den „Source“ Befehl können Teile davon zur Übersichtlichkeit ausgelagert werden.

```
set_global_assignment -name FAMILY "Cyclone IV E"
set_global_assignment -name DEVICE EP4CE115F29C7
set_global_assignment -name TOP_LEVEL_ENTITY de2_115_start
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 11.0
set_global_assignment -name PROJECT_CREATION_TIME_DATE "13:45:52 JUNE 09, 2011"
set_global_assignment -name LAST_QUARTUS_VERSION 11.0

set_global_assignment -name USE_CONFIGURATION_DEVICE ON
set_global_assignment -name CYCLONEIII_CONFIGURATION_DEVICE EPCS64
set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA

set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
# set_instance_assignment -name FAST_INPUT_REGISTER ON -to *
# set_instance_assignment -name FAST_OUTPUT_REGISTER ON -to *
set_instance_assignment -name TSU_REQUIREMENT "10 ns" -from * -to *
set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -rise
set_global_assignment -name OUTPUT_IO_TIMING_NEAR_END_VMEAS "HALF VCCIO" -fall
set_global_assignment -name OUTPUT_IO_TIMING_FAR_END_VMEAS "HALF SIGNAL SWING" -
rise
set_global_assignment -name OUTPUT_IO_TIMING_FAR_END_VMEAS "HALF SIGNAL SWING" -
fall
```

Generelle FPGA  
Definitionen

Config Device

Timing Configuration

```

set_global_assignment -name STRATIX_DEVICE_IO_STANDARD "2.5 V"
set_global_assignment -name CRC_ERROR_OPEN_DRAIN OFF
set_global_assignment -name CYCLONEIII_RESERVE_NCEO_AFTER_CONFIGURATION "USE AS
REGULAR IO"

```

General I/O Spec.

```

set_global_assignment -name FLOW_ENABLE_POWER_ANALYZER OFF

```

Power Analyzer

Partition Defintions

```

set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -
section_id Top
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL
PLACEMENT_AND_ROUTING -section_id Top
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top

```

Logic Lock Definitions for partial compilation

```

set_global_assignment -name LL_ROOT_REGION ON -section_id "Root Region"
set_global_assignment -name LL_MEMBER_STATE LOCKED -section_id "Root Region"

```

Simulation Settings

```

set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim-Altera (VHDL)"
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT VHDL -section_id
eda_simulation
set_global_assignment -name EDA_TEST_BENCH_ENABLE_STATUS TEST_BENCH_MODE -
section_id eda_simulation
set_global_assignment -name EDA_NATIVELINK_SIMULATION_TEST_BENCH
de2_115_start_tb -section_id eda_simulation
set_global_assignment -name EDA_TEST_BENCH_NAME de2_115_start_tb -section_id
eda_simulation
set_global_assignment -name EDA_DESIGN_INSTANCE_NAME NA -section_id
de2_115_start_tb
set_global_assignment -name EDA_TEST_BENCH_MODULE_NAME de2_115_start_tb -
section_id de2_115_start_tb
set_global_assignment -name EDA_TEST_BENCH_FILE ../sim/de2_115_start.tb.vhd -
section_id de2_115_start_tb

```

```

source de2_115_system_pins.qsf
source de2_115_hex_pins.qsf
source de2_115_lcd_pins.qsf
source de2_115_gpio_pins.qsf
source de2_115_vga_pins.qsf
# ... and many more ...
source de2_115_eeprom_pins.qsf
source de2_115_audio_pins.qsf
source de2_115_sd_card_pins.qsf
source de2_115_lvds_on_hsmc_pins.qsf

```

Long list of I/O Pin  
Definition Files

```

source design_files.qsf

```

External Design File List

## 6 Pin Definition Files

Für jeden Input und LVDS Pin braucht es zwei Zeilen:

- set\_location\_assignment : Zuweisung des I/O Pin zum Signalnamen
- set\_instance\_assignment -name IO\_STANDARD : Setzen des I/O Typs

Für reguläre Output Pins braucht es noch zwei zusätzliche Definitionen:

- set\_instance\_assignment -name SLEW\_RATE : Setzen des Flankensteilheit
- set\_instance\_assignment -name CURRENT\_STRENGTH\_NEW : Setzen der Strombegrenzung

### 6.1 Pin Definitionen für System-Funktionen

File: de2\_start\_system\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Definition der Pins für die System-Funktionen: Takt, Knöpfe, Schalter und LEDs

```
##      Clock Input
##      =====
set_location_assignment PIN_Y2           -to is1_clock_50
set_location_assignment PIN_AG14        -to is1_clock2_50
set_location_assignment PIN_AG15        -to is1_clock3_50
set_location_assignment PIN_AE23        -to osl_sma_clkout
set_location_assignment PIN_AH14        -to is1_sma_clkin

set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to is1_clock_50
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to is1_clock2_50
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to is1_clock3_50
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to osl_sma_clkout
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to is1_sma_clkin

set_instance_assignment -name SLEW_RATE 2 -to osl_sma_clkout
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to osl_sma_clkout

##      Pushbutton Keys
##      =====
set_location_assignment PIN_M23          -to islv4_key_n[0]
set_location_assignment PIN_M21          -to islv4_key_n[1]
set_location_assignment PIN_N21          -to islv4_key_n[2]
set_location_assignment PIN_R24          -to islv4_key_n[3]

##      Attention: Voltage Standard depends on position of Jumper JP7
set_instance_assignment -name IO_STANDARD "2.5 V" -to islv4_key_n[0]
set_instance_assignment -name IO_STANDARD "2.5 V" -to islv4_key_n[1]
set_instance_assignment -name IO_STANDARD "2.5 V" -to islv4_key_n[2]
set_instance_assignment -name IO_STANDARD "2.5 V" -to islv4_key_n[3]
```

```

##      Slide Switch
##      =====
set_location_assignment PIN_AB28          -to islv18_switch[0]
set_location_assignment PIN_AC28          -to islv18_switch[1]
...

## Attention: Voltage Standard depends on position of Jumper JP7
set_instance_assignment -name IO_STANDARD "2.5 V"      -to islv18_switch[0]
set_instance_assignment -name IO_STANDARD "2.5 V"      -to islv18_switch[1]
...

##      Red LEDs
##      =====
set_location_assignment PIN_G19           -to oslv18_red_led[0]
set_location_assignment PIN_F19           -to oslv18_red_led[1]
...

set_instance_assignment -name IO_STANDARD "2.5 V"      -to oslv18_red_led[0]
set_instance_assignment -name IO_STANDARD "2.5 V"      -to oslv18_red_led[1]
...

set_instance_assignment -name SLEW_RATE 2             -to oslv18_red_led[0]
set_instance_assignment -name SLEW_RATE 2             -to oslv18_red_led[1]
...

set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to oslv18_red_led[0]
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to oslv18_red_led[1]

##      Green LEDs
##      =====
set_location_assignment PIN_E21           -to oslv9_green_led[0]
set_location_assignment PIN_E22           -to oslv9_green_led[1]
...

set_instance_assignment -name IO_STANDARD "2.5 V"      -to oslv9_green_led[0]
set_instance_assignment -name IO_STANDARD "2.5 V"      -to oslv9_green_led[1]
...

set_instance_assignment -name SLEW_RATE 2             -to oslv9_green_led[0]
set_instance_assignment -name SLEW_RATE 2             -to oslv9_green_led[1]
...

set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to oslv9_green_led[0]
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to oslv9_green_led[1]
...

```



## 6.2 Pin Definitionen für 7-Segment Display Pins

File: de2\_start\_hex\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Pins für die 8 Stück 7-Segment Anzeigen.

```
## DE2-115 Hex Pins
##
#####

## Pin Mapping to 7-Segment LED elements
##
##          Pin 0:  Top Element      |          |
##          Pin 1:  Right Upper Element 5          1
##          Pin 2:  Right Lower Element |          |
##          Pin 3:  Bottom Element     -- 6  --
##          Pin 4:  Left Lower Element  |          |
##          Pin 5:  Left Upper Element  4          2
##          Pin 6:  Middle Element     |          |
##                                     -- 3  -

## Hex segment 0
## =====

set_location_assignment PIN_G18          -to or_hex_display.slv7_segment_0[0]
set_location_assignment PIN_F22          -to or_hex_display.slv7_segment_0[1]
...

set_instance_assignment -name IO_STANDARD "2.5 V" -to
                                     or_hex_display.slv7_segment_0[0]
set_instance_assignment -name IO_STANDARD "2.5 V" -to
                                     or_hex_display.slv7_segment_0[1]
...

set_instance_assignment -name SLEW_RATE 2 -to or_hex_display.slv7_segment_0[0]
set_instance_assignment -name SLEW_RATE 2 -to or_hex_display.slv7_segment_0[1]
...

set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to
                                     or_hex_display.slv7_segment_0[0]
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to
                                     or_hex_display.slv7_segment_0[1]
...

## Hex segment 1 - 7
## =====
...
```

## 6.3 Pin Definitionen für Audio Codec Pins

File: de2\_start\_system\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Serielle I2S Schnittstelle zum Audio-Codec.  
Die ausgehenden Signale sind in einem Record zusammengefasst.

```
## DE2-115 AUDIO Codec Pins
##
#####

set_location_assignment PIN_D1 -to r_audio_out.sl_dac_data
set_location_assignment PIN_E3 -to r_audio_out.sl_dac_lrclk
set_location_assignment PIN_C2 -to r_audio_out.sl_adc_lrclk
set_location_assignment PIN_E1 -to r_audio_out.sl_xclk
set_location_assignment PIN_F2 -to r_audio_out.sl_bclk

set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to r_audio_out.sl_dac_data
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to r_audio_out.sl_dac_lrclk
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to r_audio_out.sl_adc_lrclk
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to r_audio_out.sl_xclk
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to r_audio_out.sl_bclk

set_instance_assignment -name SLEW_RATE 2 -to r_audio_out.sl_dac_data
set_instance_assignment -name SLEW_RATE 2 -to r_audio_out.sl_dac_lrclk
set_instance_assignment -name SLEW_RATE 2 -to r_audio_out.sl_adc_lrclk
set_instance_assignment -name SLEW_RATE 2 -to r_audio_out.sl_xclk
set_instance_assignment -name SLEW_RATE 2 -to r_audio_out.sl_bclk

set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to r_audio_out.sl_dac_data
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to r_audio_out.sl_dac_lrclk
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to r_audio_out.sl_adc_lrclk
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to r_audio_out.sl_xclk
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to r_audio_out.sl_bclk

set_location_assignment PIN_D2 -to isl_audio_adc_data
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to isl_audio_adc_data
```

## 6.4 Pin Definitionen für EEPROM Pins

File: de2\_start\_eeprom\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies ist eine einfache I2C Schnittstelle zu einem kleinen EEPROM Chip.  
Im Gegensatz zum Flash Chip ist dies ein NOR-Flash und daher kleiner aber individuell pro Byte lese- und schreibbar.

## 6.5 Pin Definitionen für Ethernet Pins

File: de2\_start\_ethernet\_0\_pins.qsf  
de2\_start\_ethernet\_1\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies sind die Pin-Definitionen für die beiden Gigabit-Ethernet Schnittstellen. Eingang- und Ausgang-Pins sind jeweils richtungsgetrennt in je einem Record zusammen.

## 6.6 Pin Definitionen für Extended IO Pins

File: de2\_start\_extended\_io\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies ist ein zusätzlicher Stecker für I/O Pins, daher der Name Extended-IO. Konkret sind dies nur 7 bidirektionale Signale auf einem 14-Pin Stecker.

## 6.7 Pin Definitionen für Flash Pins

File: de2\_start\_flash\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Der 8-Bit Datenbus ist bidirektional geführt. Alle anderen Pins sind nur Ausgang und in einem Record zusammengefasst.

## 6.8 Pin Definitionen für GPIO Pins

File: de2\_start\_gpio\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies sind 36 bidirektionale I/O Pins in einem zweireihigen Stecker, welcher auch zu den DE2-35 und DE2-70 Boards kompatibel ist.

## 6.9 Pin Definitionen für GPIO auf HSMC Pins

File: de2\_start\_gpio\_on\_hsmc\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Diese Schnittstelle kann statt als LVDS auch als ganz „normale“ GPIO Reihe verwendet werden. Dabei sind dann allerdings 5 Pins „nur Input“, während die anderen bidirektional dienen können.

Insgesamt stehen so  $5 + 76 = 82$  Pins zur Verfügung.

## 6.10 Pin Definitionen für Infrarot Receiver Pin

File: de2\_start\_ir\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies ist nur ein einziger Pin für den IR Empfänger.

### 6.11 Pin Definitionen für LCD Pins

File: de2\_start\_lcd\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Alle Signale sind zu einem Record zusammengefasst. Das LCD-Display hat eigene Intelligenz und wird per 8-Bit Befehle angesteuert.

### 6.12 Pin Definitionen für LVDS auf HSMC Pins

File: de2\_start\_lvds\_on\_hsmc\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Diese Schnittstelle ist absichtlich auskommentiert, da die LVDS Pin-Definitionen mit den Definitionen der Hex 7-Segment Anzeige kollidieren. Zumindest kennt man zur Zeit am NTB noch keine Konfiguration welche ohne Fehler beides gleichzeitig unterstützt. Die HSMC Schnittstelle kann auch ähnlich wie die GPIO Schnittstelle einfach mit „normalen“ Input- oder Output Signalen verwendet werden.

Im LVDS Mode sind pro Richtung (Ein- und Ausgang) die folgenden Signale vorgesehen:

- 1 x „normales“ Taktsignal ohne LVDS
- 2 x LVDS Taktsignale
- 17 LVDS Datensignale

Zusätzlich ist ein „normaler“ bidirektionaler 4-Bit Datenbus definiert.

### 6.13 Pin Definitionen für PS2 Pins

File: de2\_start\_ps2\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Die PS2 Schnittstelle ist doppelt ausgeführt für je eine Maus und eine Tastatur. Alle 4 Pins sind zu einem Record zusammengefasst.

### 6.14 Pin Definitionen für SD-Card Pins

File: de2\_start\_sd\_card\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies ist ein 4-Bit bidirektionaler Bus, sowie 2 Ausgangs-Pins für Takt und Command, sowie ein Eingangs-Pin für Write-Protection.

## 6.15 Pin Definitionen für SDRAM Pins

File: de2\_start\_sdram\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Dies ist eine 32-Bit Breite Schnittstelle zu einem SDRAM (eigentlich 2x 16-Bit). Ausser dem bidirektionalen Datenbus sind alle anderen Signale zu je einem Eingangs- und einem Ausgangs-Record zusammengefasst.

## 6.16 Pin Definitionen für SRAM Pins

File: de2\_start\_sram\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Ähnlich wie schon bei der SDRAM Schnittstelle ist auch hier der 16-Bit breite Datenbus bidirektional, während die anderen Pins zu einem Ausgangs-Record zusammengefasst sind.

## 6.17 Pin Definitionen für TV Decoder Pins

File: de2\_start\_tv\_decoder\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Abgesehen von den Eingängen die in einem Record zusammengefasst sind, hat diese Schnittstelle nur noch einen Ausgang-Pin und einen Bidirektionalen Pin für den I2C Config Port.

## 6.18 Pin Definitionen für UART Pins

File: de2\_start\_uart\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Ein Tx- und ein Rx-Pin, jeweils noch mit einem Handshake-Pin, paarweise in Records.

## 6.19 Pin Definitionen für USB Pins

File: de2\_start\_usb\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Alle Ausgangs-Signale sind in einem Record zusammengefasst. Ebenso alle Eingangssignale. Hinzu kommen noch die bidirektionalen Pins für den 16-Bit Datenbus. Der Standard entspricht dem USB on-the-go Vorschlag, deshalb wird es bei Altera auch oft als OTG bezeichnet.

## 6.20 Pin Definitionen für VGA Pins

File: de2\_start\_vga\_pins.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Pins für 3x8-Bit Farbe VGA Ausgang ... alle Signale in einem Record zusammengefasst.

## 7 Definition der Design Files

File: design\_files.qsf

Verzeichnis: orig\_qsf\_files

Beschreibung: Liste der Design-Files. Diese Liste ist zur Übersichtlichkeit ausgelagert, könnte aber auch direkt im <projekt\_namen>.qsf File stehen. Diese Liste kann entweder manuell erweitert werden oder über das GUI in Quartus-II. Falls zusätzliche Files mit dem Quartus-II GUI hinzugefügt werden, ist es nicht garantiert, dass die neuen Definitions-Zeilen an diese Stelle geschrieben werden.

```
set_global_assignment -name VHDL_FILE ../src/de2_115_start.p.vhd
set_global_assignment -name VHDL_FILE ../src/de2_115_start.e.vhd
set_global_assignment -name VHDL_FILE ../src/de2_115_start.a_dummy.vhd
set_global_assignment -name SDC_FILE ../src/de2_115_start.sdc
```

## 8 SDC Synopsys Design Constraint File

File: de2\_start.sdc

Verzeichnis: src

Beschreibung: Definition der Timing-Zusammenhänge im Synopsys Design Constraint File-Format. Die Kommentare im File sind schon recht gut und selbsterklärend.

```
# Constrain clock port clk with a 10-ns requirement and 0.3ns jitter
create_clock -period 20 [get_ports isl_clock_50]
create_clock -period 20 [get_ports isl_clock2_50]
create_clock -period 20 [get_ports isl_clock3_50]

# Specify clock uncertainty (jitter)
set_clock_uncertainty 0.3ns -to isl_clock_50
set_clock_uncertainty 0.3ns -to isl_clock2_50
set_clock_uncertainty 0.3ns -to isl_clock3_50

# Automatically apply a generate clock on the phase-locked loops (PLLs) outputs
# This command can be safely left in the SDC even if no PLLs exist in the design
derive_pll_clocks

# Constrain the input I/O path
set_input_delay -clock isl_clock_50 -max 3 [all_inputs]
set_input_delay -clock isl_clock_50 -min 2 [all_inputs]

# Constrain the output I/O path
set_output_delay -clock isl_clock_50 2 [all_outputs]
```

## 9 Top-Level Package File

File: de2\_start.p.qsf  
 Verzeichnis: src

Beschreibung: Top-Level Package File. Kann nach Belieben gekürzt oder ergänzt werden.  
 Einfach auskommentieren oder löschen was nicht gebraucht wird.

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

PACKAGE de2_115_start_pkg IS

  TYPE t_hex_display IS RECORD
    slv7_segment_0      : std_logic_vector(6 DOWNTO 0);    --  -- 0  --
    slv7_segment_1      : std_logic_vector(6 DOWNTO 0);    --  |      |
    slv7_segment_2      : std_logic_vector(6 DOWNTO 0);    --  5      1
    slv7_segment_3      : std_logic_vector(6 DOWNTO 0);    --  |      |
    slv7_segment_4      : std_logic_vector(6 DOWNTO 0);    --  -- 6  --
    slv7_segment_5      : std_logic_vector(6 DOWNTO 0);    --  |      |
    slv7_segment_6      : std_logic_vector(6 DOWNTO 0);    --  4      2
    slv7_segment_7      : std_logic_vector(6 DOWNTO 0);    --  |      |
  END RECORD t_hex_display;

  TYPE t_audio_out IS RECORD
    sl_dac_data          : std_logic;
    sl_dac_lrclk         : std_logic;
    sl_adc_lrclk         : std_logic;
    sl_xclk              : std_logic;
    sl_bclk              : std_logic;
  END RECORD t_audio_out;

  TYPE t_ethernet_out IS RECORD
    sl_gtx_clk          : std_logic;
    sl_tx_ena           : std_logic;
    sl_tx_err           : std_logic;
    slv4_tx_data        : std_logic_vector(3 DOWNTO 0);
    sl_mdc               : std_logic;
    sl_reset_n          : std_logic;
  END RECORD t_ethernet_out;

  TYPE t_ethernet_in IS RECORD
    sl_tx_clk           : std_logic;
    sl_rx_clk           : std_logic;
    sl_rx_dv            : std_logic;
    sl_rx_err           : std_logic;
    sl_rx_crs           : std_logic;
    sl_rx_col           : std_logic;
    slv4_rx_data        : std_logic_vector(3 DOWNTO 0);
    sl_int_n            : std_logic;
    sl_link100          : std_logic;
  END RECORD t_ethernet_in;
  
```

Definition der Signale für 7-Segment Hex Display:  
 Jede Anzeige hat 6 Leucht balken, nummeriert wie unten angezeigt.  
 Achtung: Die Bits sind low-aktiv.

Definition der Signale für Audio Codec ... alles ausser den seriellen IN Daten!

Definition der Signale für Gigabit Ethernet Out

Definition der Signale für Gigabit Ethernet In

```

TYPE t_flash_ctrl IS RECORD
  slv23_addr      : std_logic_vector(22 DOWNT0 0);
  sl_ce_n         : std_logic;
  sl_oe_n         : std_logic;
  sl_we_n         : std_logic;
  sl_reset_n     : std_logic;
  sl_wp_n         : std_logic;
END RECORD t_flash_ctrl;
  
```

Definition der Signale für Flash Memory ... alles ausser den INOUT Daten!

```

TYPE t_lcd_display IS RECORD
  slv8_lcd_data   : std_logic_vector(7 DOWNT0 0);
  sl_lcd_enable   : std_logic;
  sl_lcd_read_write_n : std_logic;
  sl_lcd_data_command_n : std_logic;
  sl_lcd_power_on_off_n : std_logic;
  sl_lcd_light_on_off_n : std_logic;
END RECORD t_lcd_display;
  
```

Definition der Signale für LCD Anzeige mit eigener Intelligenz ...

```

TYPE t_ps2_in IS RECORD
  sl_ps2_clk      : std_logic;
  sl_ps2_clk2     : std_logic;
  sl_ps2_dat      : std_logic;
  sl_ps2_dat2     : std_logic;
END RECORD t_ps2_in;
  
```

Definition der Signale für doppelte PS2-Schnittstelle für Keyboard und Maus

```

TYPE t_sdr_ctrl IS RECORD
  sl_cke          : std_logic;
  sl_clk          : std_logic;
  sl_cs_n         : std_logic;
  sl_we_n         : std_logic;
  sl_cas_n        : std_logic;
  sl_ras_n        : std_logic;
  slv2_ba         : std_logic_vector(1 DOWNT0 0);
  slv4_dqm        : std_logic_vector(3 DOWNT0 0);
  slv13_addr      : std_logic_vector(12 DOWNT0 0);
END RECORD t_sdr_ctrl;
  
```

Definition der Signale für SDRAM... alles ausser den INOUT Daten!

```

TYPE t_tv_decoder IS RECORD
  slv8_data       : std_logic_vector(7 DOWNT0 0);
  sl_vsync        : std_logic;
  sl_hsync        : std_logic;
  sl_clk27        : std_logic;
END RECORD t_tv_decoder;
  
```

Definition der Signale für TV Decoder ... nur Eingangssignale!

```

TYPE t_sram_ctrl IS RECORD
  slv20_addr      : std_logic_vector(19 DOWNT0 0);
  sl_ce_n         : std_logic;
  sl_oe_n         : std_logic;
  sl_we_n         : std_logic;
  sl_ub_n         : std_logic;
  sl_lb_n         : std_logic;
END RECORD t_sram_ctrl;
  
```

Definition der Signale für SRAM... alles ausser den INOUT Daten!

```

TYPE t_uart_out IS RECORD
  
```



```

sl_txd                                : std_logic;
sl_cts                                : std_logic;
END RECORD t_uart_out;

TYPE t_uart_in IS RECORD
sl_rxd                                : std_logic;
sl_rts                                : std_logic;
END RECORD t_uart_in;

TYPE t_usb_out IS RECORD
slv2_addr                             : std_logic_vector(1 DOWNTO 0);
sl_cs_n                               : std_logic;
sl_wr_n                               : std_logic;
sl_rd_n                               : std_logic;
sl_reset_n                            : std_logic;
slv2_dack_n                            : std_logic_vector(1 DOWNTO 0);
END RECORD t_usb_out;

TYPE t_usb_in IS RECORD
slv2_int                               : std_logic_vector(1 DOWNTO 0);
slv2_dreq                              : std_logic_vector(1 DOWNTO 0);
sl_lspeed                              : std_logic;
sl_hspeed                              : std_logic;
END RECORD t_usb_in;

TYPE t_vga IS RECORD
usig8_vga_red                         : unsigned(7 DOWNTO 0);
usig8_vga_green                       : unsigned(7 DOWNTO 0);
usig8_vga_blue                        : unsigned(7 DOWNTO 0);
sl_sl_vga_clock                       : std_logic;
sl_vga_blank_n                        : std_logic;
sl_vga_hsync                          : std_logic;
sl_vga_vsync                          : std_logic;
sl_vga_sync_n                         : std_logic;
END RECORD t_vga;

COMPONENT de2_115_start IS
PORT (
----- Clock Input -----
isl_clock_50                          : IN  std_logic ;           -- 50 MHz
isl_clock2_50                         : IN  std_logic ;           -- 50 MHz
isl_clock3_50                         : IN  std_logic ;           -- 50 MHz
osl_sma_clkout                        : OUT std_logic ;           -- Ext.
isl_sma_clkin                         : IN  std_logic ;           -- Ext.
----- Push Button and Swicht -----
islv4_key_n                           : IN  std_logic_vector(3 DOWNTO 0);
islv18_switch                         : IN  std_logic_vector(17 DOWNTO 0);
----- LED -----
oslv18_red_led                        : OUT std_logic_vector(17 DOWNTO 0);
oslv9_green_led                      : OUT std_logic_vector(8 DOWNTO 0);
----- 7-SEG Dispaly -----
or_hex_display                        : OUT t_hex_display;

```

Definition der Signale für  
UART mit einfachen  
Handshake-Signalen

Definition der Signale für  
USB... nur Ausgangssignale!

Definition der Signale für  
USB... nur  
Eingangssignale!

Definition der Signale für  
VGA: 3 x 8-Bit Farben,  
plus Sync Signale

```

----- Audio Codec -----
r_audio_out          : OUT t_audio_out;
isl_audio_adc_data   : IN  std_logic;
----- EEPROM I2C -----
osl_eeprom_sclk      : OUT std_logic;
bsl_eeprom_sdat      : INOUT std_logic;
----- Ethernet 0 -----
r_enet_0_out         : OUT t_ethernet_out;
r_enet_0_in          : IN  t_ethernet_in;
sl_enet_0_mdio       : INOUT std_logic;
----- Ethernet 1 -----
r_enet_1_out         : OUT t_ethernet_out;
r_enet_1_in          : IN  t_ethernet_in;
sl_enet_1_mdio       : INOUT std_logic;
----- Extended IO -----
bslv7_ex_io          : INOUT std_logic_vector(6 DOWNT0 0);
----- FLASH -----
r_flash_ctrl         : OUT t_flash_ctrl;
sl_flash_ry          : IN  std_logic;
bslv8_flash_data     : INOUT std_logic_vector(7 DOWNT0 0);
----- GPIO -----
bslv36_gpio          : INOUT std_logic_vector(35 DOWNT0 0);
----- GPIO on HSMC -----
islv5_hsmc_gpio      : IN  std_logic_vector(4 DOWNT0 0);
bslv77_hsmc_gpio     : INOUT std_logic_vector(76 DOWNT0 0);
----- IR Receiver -----
isl_ir_rxd           : IN  std_logic;
----- LCD Display -----
or_lcd_display       : OUT t_lcd_display;
----- LVDS on HSMC --- conflicts with Hex Display -----
-- osl_hsmc_clk_out   : OUT  std_logic_vector( 2 DOWNT0 0);
-- isl_hsmc_clk_in    : IN   std_logic_vector( 2 DOWNT0 0);
-- oslv17_hsmc_data_out : OUT  std_logic_vector(16 DOWNT0 0);
-- islv17_hsmc_data_in : IN   std_logic_vector(16 DOWNT0 0);
-- bslv4_hsmc_data     : INOUT std_logic_vector( 3 DOWNT0 0);
----- PS/2 -----
ir_ps2               : IN  t_ps2_in;           -- Keyboard / Maus
----- SD Card -----
bslv4_sd_card_data   : INOUT std_logic_vector(3 DOWNT0 0);
osl_sd_card_clk      : OUT  std_logic;
bsl_sd_card_cmd      : INOUT std_logic;
isl_sd_card_wp_n     : IN   std_logic;
----- SDRAM -----
r_sdr_am_ctrl        : OUT  t_sdr_am_ctrl;
bslv32_sdr_am_data   : INOUT std_logic_vector(31 DOWNT0 0);
----- SRAM -----
r_sram_ctrl          : OUT  t_sram_ctrl;
bslv16_sram_data     : INOUT std_logic_vector(15 DOWNT0 0);
----- TV Decoder -----
ir_tv_decoder        : IN  t_tv_decoder;
osl_td_reset_n       : OUT  std_logic;
osl_td_i2c_sclk      : OUT  std_logic;
bsl_td_i2c_sdat      : INOUT std_logic;

```

```

----- UART -----
or_uart_out      : OUT t_uart_out;           -- UART Transmitter
ir_uart_in       : IN  t_uart_in;           -- UART Receiver
----- USB -----
bslv16_usb_data  : INOUT std_logic_vector(15 DOWNT0 0);
or_usb_out       : OUT  t_usb_out;
ir_usb_in        : IN   t_usb_in;
----- VGA -----
or_vga           : OUT  t_vga
) ;

END COMPONENT de2_115_start;
END PACKAGE de2_115_start_pkg;

```

## 10 Top-Level Entity File

File: de2\_start.e.qsf  
Verzeichnis: src

Beschreibung: Top-Level Entity File. Praktisch identisch zum Package File, einfach ohne die Definition der Records, und ENTITY statt COMPONENT. Kann nach Belieben gekürzt oder ergänzt werden. Einfach auskommentieren oder löschen was nicht gebraucht wird.

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY de2_115_start IS
  PORT (
    ----- Clock Input -----
    isl_clock_50      : IN  std_logic ;           -- 50 MHz
    isl_clock2_50     : IN  std_logic ;           -- 50 MHz
    isl_clock3_50     : IN  std_logic ;           -- 50 MHz
    ...
    ...
    ...
    ----- VGA -----
    or_vga            : OUT  t_vga
  ) ;

END ENTITY de2_115_start;

```

## 11 Top-Level Dummy Architektur File

File: de2\_start.a\_dummy.qsf  
Verzeichnis: src

Beschreibung: Top-Level Dummy-Architektur. Diese dient nur dazu, dass wenigstens einige Signale miteinander verbunden sind ... um die ersten Kompilationen zu testen.  
Man könnte dies beliebig erweitern, so dass alle Signale irgendwie verbunden sind, und dass es keine Warnungen mehr gibt.

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

USE work.de2_115_start_pkg.ALL;

ARCHITECTURE dummy OF de2_115_start IS

BEGIN
    main_proc : PROCESS (isl_clock_50)
    BEGIN
        IF rising_edge(isl_clock_50) THEN
            oslv18_red_led      <= islv18_switch;
        END IF;
    END PROCESS main_proc;

    oslv9_green_led(3 DOWNT0 0)      <= islv4_key_n;
    oslv9_green_led(8 DOWNT0 6)      <= islv4_key_n(2 DOWNT0 0);
    osl_sma_clkout                    <= isl_sma_clk;
    oslv9_green_led(4)                <= isl_clock2_50;
    oslv9_green_led(5)                <= isl_clock3_50;

    or_uart_out.sl_txd                <= ir_uart_in.sl_rxd;
    or_uart_out.sl_cts                <= ir_uart_in.sl_rts;

END ARCHITECTURE dummy;
```

## 12 Testbench File

File: de2\_start.tb.vhd

Verzeichnis: sim

Beschreibung: Das ist einfach nur der Anfang für ein gutes Testbench-File, damit Quartus im QSF-File schon etwas zum Referenzieren hat.

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

USE work.de2_115_start_pkg.ALL;

ENTITY simple_test_tb IS
END ENTITY simple_test_tb;

ARCHITECTURE Testbench OF simple_test_tb IS

SIGNAL clock          : std_logic := '0';
SIGNAL i_count        : integer   := 0;

BEGIN
  --      ##      Unit Under Test Instantiation
  u_dut : de2_115_start PORT MAP (
    --
    --
  );

  --      ##      Clock Process
  tb_clock_proc : PROCESS
  BEGIN
    WAIT FOR 10 ns; clock <= NOT clock;
  END PROCESS tb_clock_proc;

  ----      ##      TB Main Process
  tb_main_proc : PROCESS
  BEGIN
    -- Generate stimulus
    -- ...
  END PROCESS tb_main_proc;

  ---- ## End simulation when enough samples are checked
  ASSERT i_count < 20 REPORT "End of simulation" SEVERITY FAILURE;

END ARCHITECTURE Testbench;

```

Instanziierung der DUT (Device Under Test) mit allen Signalen

Einfacher Takt-Prozess für 50 MHz.

Hier sollten dann die Stimuli erzeugt werden ... und kontrollieren werden.

Eine Art, die Simulation zu beenden, wenn i\_count den Wert 20 erreicht ...