



FPGA für Anfänger

Labortage 2017

Mark Hoffmann

Inhaltsverzeichnis

- ① Beispielanwendung
 - LED Lauflicht
- ② Was ist ein FPGA?
 - Hardware Interna
- ③ Entwicklungsboard
 - Cyclone IV E FPGA
 - Development Kit

- ④ Entwicklungssoftware
 - Quartus Prime
 - ModelSim
 - VHDL & Verilog
- ⑤ Demonstration
 - Live!

LED Lauflicht

LED Lauflicht

Möglichst einfache
Beispielanwendung

Lauflicht mit vier LEDs (rot)
und einer Status LED (grün)



Box mit Lauflicht

FPGA

Ein **Field Programmable Gate Array** ist eine im Feld, also vor Ort, beim Kunden, reprogrammierbare Logikgatter Anordnung



Xilinx XC2064 FPGA

FPGA

Logikgatter

Ein Logikgatter ist ein Bauelement, das mehrere digitale Eingangssignale zu einem digitalen Ausgangssignal umsetzt - eine Boole'sche Funktion realisiert

NICHT	
A	Y
1	0
0	1

UND		
A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

NAND		
A	B	Y
1	1	0
1	0	1
0	1	1
0	0	1

XNAND (Ungleichheit)		
A	B	Y
1	1	0
1	0	1
0	1	1
0	0	0

Oder		
A	B	Y
1	1	1
1	0	1
0	1	1
0	0	0

NOR		
A	B	Y
1	1	0
1	0	0
0	1	0
0	0	1

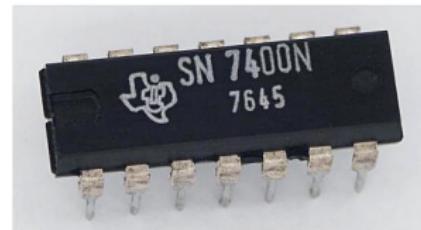
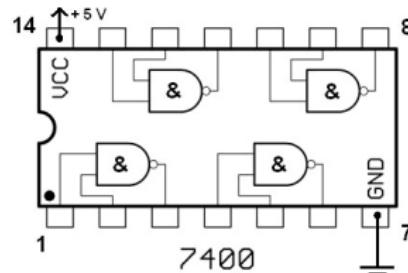
XNOR (Gleichheit)		
A	B	Y
1	1	1
1	0	0
0	1	0
0	0	1

Wahrheitstabellen für Gatter

FPGA

Entwicklung zum FPGA

- 7400er Serie
- PAL / GAL

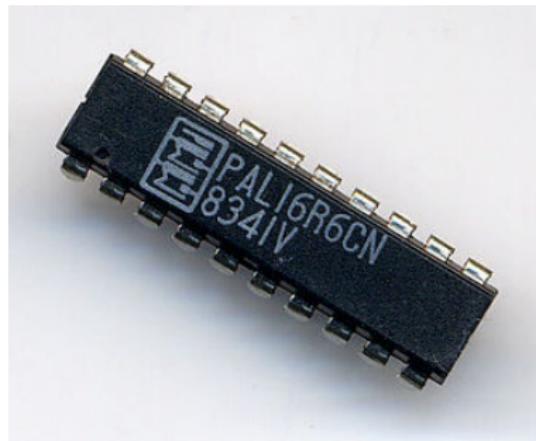


TI SN7400N

FPGA

Entwicklung zum FPGA

- 7400er Serie
- PAL / GAL
- CPLD

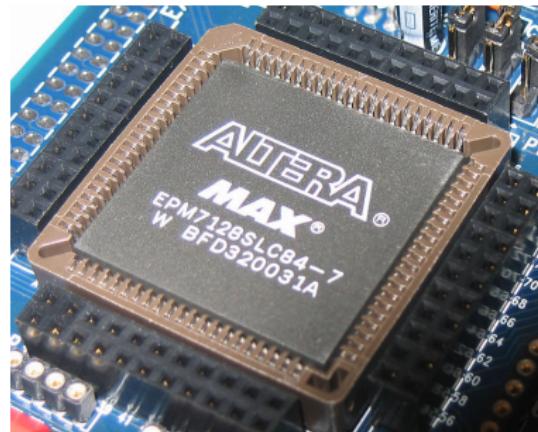


MMI 16R6 PAL

FPGA

Entwicklung zum FPGA

- 7400er Serie
- PAL / GAL
- CPLD
- FPGA



Altera MAX 7000 CPLD

FPGA

Entwicklung zum FPGA

- 7400er Serie
- PAL / GAL
- CPLD
- FPGA
- ASIC

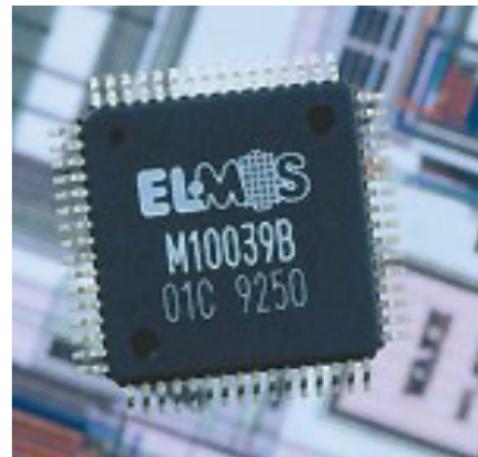


Altera Stratix IV FPGA

FPGA

Entwicklung zum FPGA

- 7400er Serie
- PAL / GAL
- CPLD
- FPGA
- ASIC

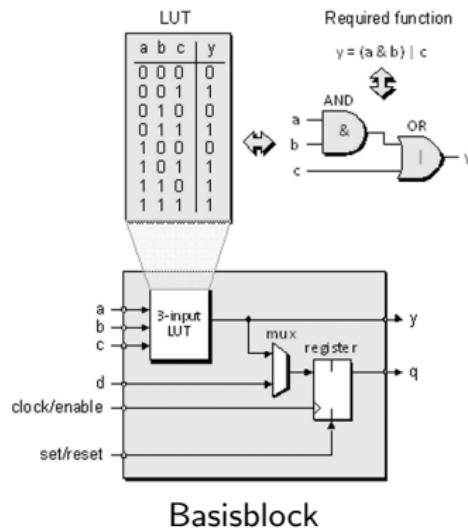


Elmos Asic

FPGA

FPGA Grundstruktur

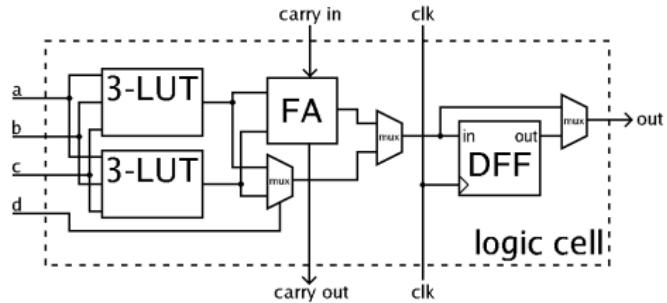
- Basisblock (LE)
- CLB



FPGA

FPGA Grundstruktur

- Basisblock (LE)
- CLB
- M9K Block RAM

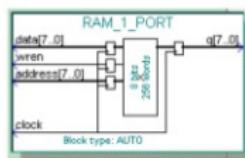


Configurable Logic Block mit Full Adder

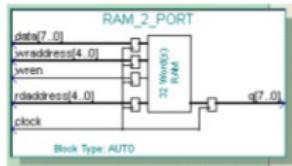
FPGA

FPGA Grundstruktur

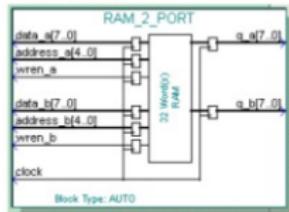
- Basisblock (LE)
- CLB
- M9K Block RAM
- DSP Blocks



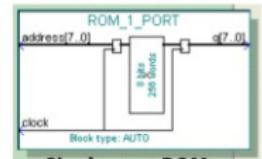
Single-port RAM



Simple dual-port RAM



True dual-port RAM



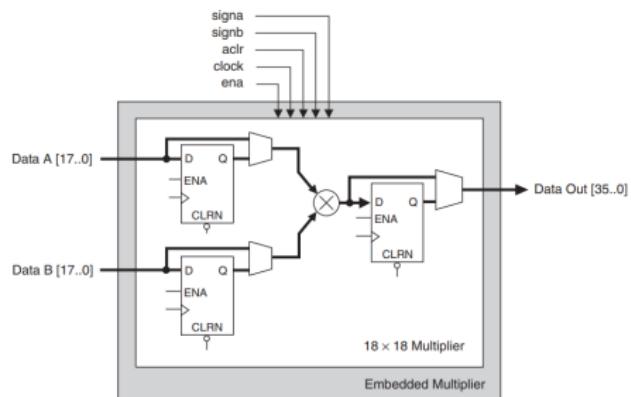
Single-port ROM

M9K Block RAM

FPGA

FPGA Grundstruktur

- Basisblock (LE)
- CLB
- M9K Block RAM
- DSP Blocks
- PLL

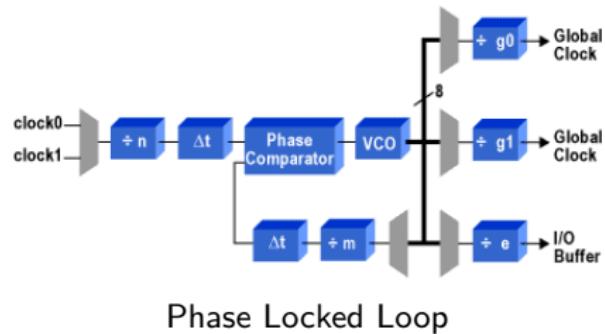


Embedded Multiplier (18 oder 9 Bit)

FPGA

FPGA Grundstruktur

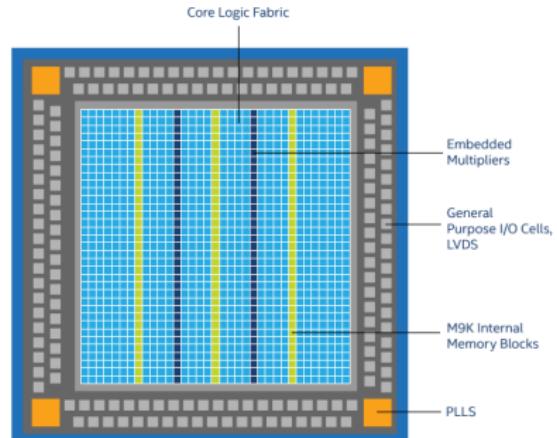
- Basisblock (LE)
- CLB
- M9K Block RAM
- DSP Blocks
- PLL
- Floor Plan



FPGA

FPGA Grundstruktur

- Basisblock (LE)
- CLB
- M9K Block RAM
- DSP Blocks
- PLL
- Floor Plan

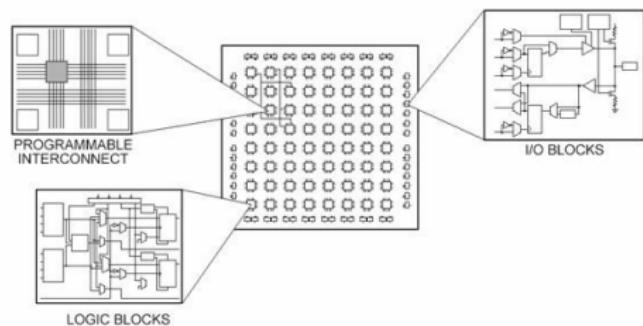


Floor Plan

FPGA

FPGA Grundstruktur

- Logikblöcke
- Verbindungsgeflecht

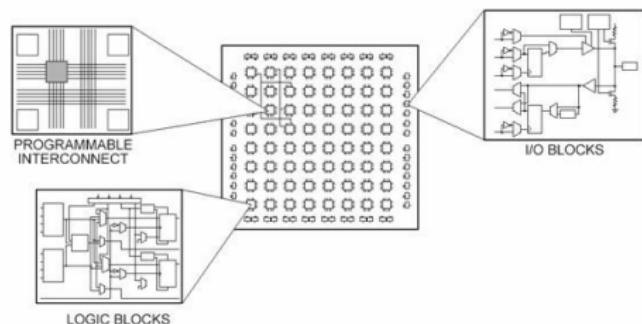


Funktionsblöcke

FPGA

FPGA Grundstruktur

- Logikblöcke
- Verbindungsgeflecht
- I/O-Blöcke

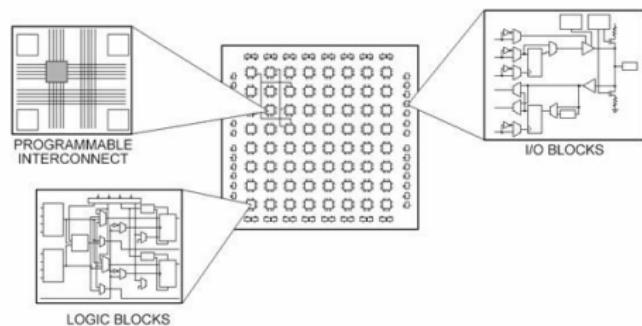


Funktionsblöcke

FPGA

FPGA Grundstruktur

- Logikblöcke
- Verbindungsgeflecht
- I/O-Blöcke



Funktionsblöcke

Cyclone IV E FPGA

Altera Cyclone IV E FPGA

- Cyclone IV E FPGA: EP4CE6E22C8N
- Anzahl der Logikzellen: 6272
- Anzahl der LABs: 392
- Embedded M9K Memory (Block RAM):
270 Kbit (33 Kbyte)
- I/O-Spannung: 3,3 V
- I/O-Pin-Anzahl: 91
- Spannungsversorgung: 1,0 V bis 1,2 V
- Maximale Betriebsfrequenz: 200 MHz
- Gehäuse: QFP-144

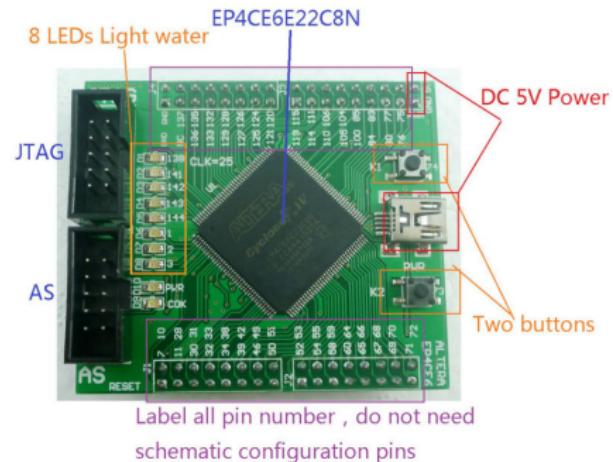


Altera Cyclone IV E

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8



Vorderseite

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8
- Push-Buttons: 2



Vorderseite

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8
- Push-Buttons: 2
- I/O-Pins: 91

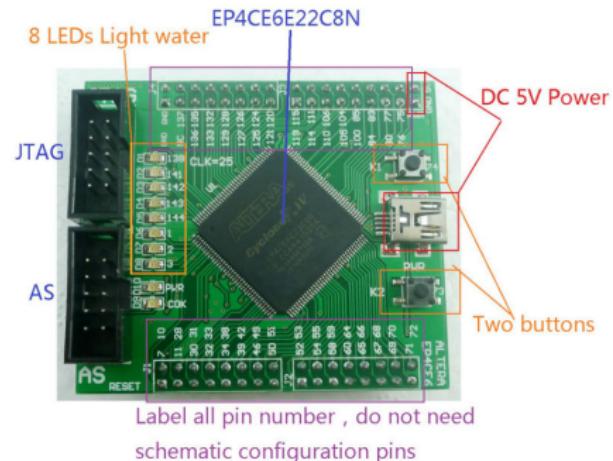


Vorderseite

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8
- Push-Buttons: 2
- I/O-Pins: 91
- JTAG und AS Port

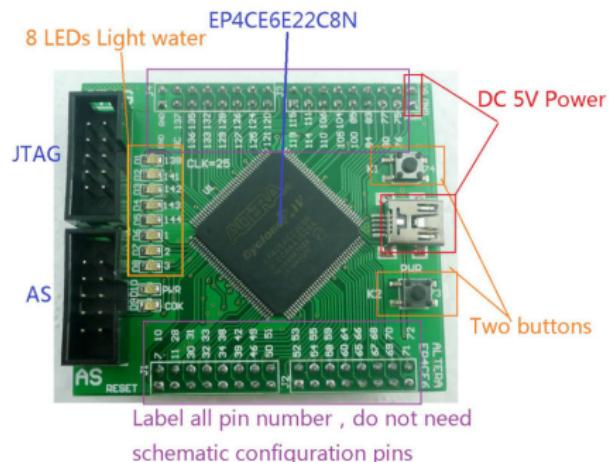


Vorderseite

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8
- Push-Buttons: 2
- I/O-Pins: 91
- JTAG und AS Port
- Betrieb mit 5 V (USB)

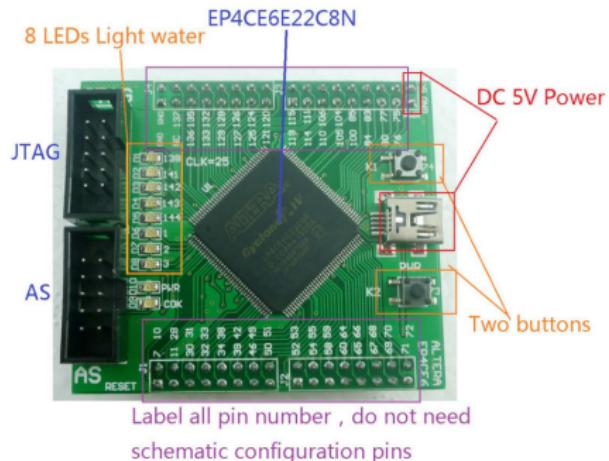


Vorderseite

Entwicklungsboard

Entwicklungsboard

- FPGA: Cyclone IV E EP4CE6E22C8N (Altera)
- LEDs: 8
- Push-Buttons: 2
- I/O-Pins: 91
- JTAG und AS Port
- Betrieb mit 5 V (USB)



Vorderseite

Entwicklungsboard

Entwicklungsboard

- Quarzoszillator: 25 MHz
- Serial-Flash-Memory:
M25P40 4 MB



Rückseite

Entwicklungsboard

Entwicklungsboard

- Quarzoszillator: 25 MHz
- Serial-Flash-Memory:
M25P40 4 MB
- Reset Button



Rückseite

Entwicklungsboard

Entwicklungsboard

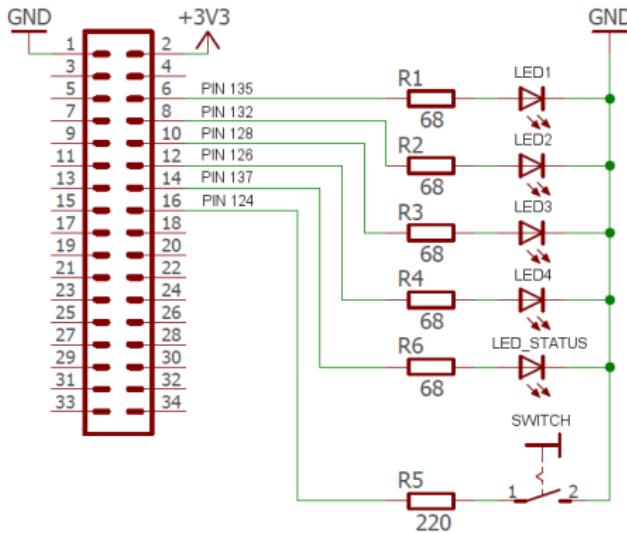
- Quarzoszillator: 25 MHz
- Serial-Flash-Memory:
M25P40 4 MB
- Reset Button



Rückseite

Entwicklungsboard

Schaltungsdiagramm Box - LEDs und Schalter



Entwicklungsboard

Entwicklungsboard

- JTAG-Adapter für Flashen des Bitstreams



JTAG-Adapter

Quartus Prime

Quartus Prime

- Eingabe von VHDL/Verilog Code
- Synthese (Kompilieren)
Vorgang mit Fehlermeldungen und Warnungen

The screenshot shows the Quartus Prime Lite Edition interface. The Project Navigator pane displays a project named 'ov7670_top' containing several source files: debounce.bnn_debounce, ov7670_capture.capture, ov7670_controller.controller, frame_bufferfb, ug419t.vhd, and sys_vuart.vhd. The Hierarchy pane shows the Entity Instance 'ov7670_top'. The Tasks pane lists the following tasks: Compile Design, Analysis & Synthesis, Filter (Place & Route), Assembler (Generate programming), and TimeQuest Timing Analysis. The status bar indicates 'Module % Progress Time'. The bottom pane shows a message window with tabs for All, ID, Message, and System.

```
3 -- Description: Top level for the OV7670 camera project.
4
5 -- State Machine Types
6
7 library IEEE;
8 use IEEE.STD.TEXTIO.ALL;
9
10 package state_machine_states_type is
11 type state_type is (v_axon, v_arct, v_sif, testbars, led, pause);
12 end package state_machine_states_type;
13
14 -- http://stackoverflow.com/questions/20308514/declaring-an-array-within-an-entity
15
16 library IEEE;
17 use IEEE.STD.TEXTIO.ALL;
18 use IEEE.STD.TEXTIO.ALL;
19 use work.state_machine_states_type.all;
20
21 entity ov7670_top is
22 Port (
23 clk50 : in STD_LOGIC;
24 OV7670_S10C : inout STD_LOGIC;
25 OV7670_S10I : inout STD_LOGIC;
26 OV7670_RESET : out STD_LOGIC;
27 OV7670_PWDN : out STD_LOGIC;
28 OV7670_POWER : out STD_LOGIC;
29 OV7670_VREF : out STD_LOGIC);
30
31 end entity ov7670_top;
```

Quartus Prime IDE

Quartus Prime

Quartus Prime

- Eingabe von VHDL/Verilog Code
- Synthese (Kompilieren) Vorgang mit Fehlermeldungen und Warnungen
- Verzweigung zu Unterprogrammen

The screenshot shows the Quartus Prime Lite Edition interface. The top menu includes File, Edit, View, Project, Assignments, Processing, Tools, Window, Help, and a search bar. The main area has tabs for Hierarchy, Source, and Block Diagram.

Hierarchy Tab: Shows a project tree for 'Cyclone IV E: EP4CE15F17C8' with components like 'ov7670_top_tb', 'debounce_btm_debounce', 'ov7670_capture_capture', 'ov7670_controller_controller', 'frame_bufferfb', 'vga_intf_vga', and 'sys_vuart'. A 'Tasks' panel below shows compilation tasks: 'Compile Design', 'Analysis & Synthesis', 'Filter (Place & Route)', 'Assembler (Generate programming)', and 'TimeQuest Timing Analysis', all marked as successful.

Code Editor Tab: Displays VHDL code for 'ov7670_top.vhd'. The code includes library IEEE, component declarations for state machine types, and an entity definition for 'ov7670_top' with various port connections. A scroll bar indicates the code continues.

Status Bar: Shows the status bar with tabs for All, ID, Message, and a progress bar indicating 0% completion.

Quartus Prime IDE

Quartus Prime

Quartus Prime

- Eingabe von VHDL/Verilog Code
- Synthese (Kompilieren)
Vorgang mit Fehlermeldungen und Warnungen
- Verzweigung zu Unterprogrammen

The screenshot shows the Quartus Prime Lite Edition interface. The top menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, Help, and a search bar. The main window has several panes:

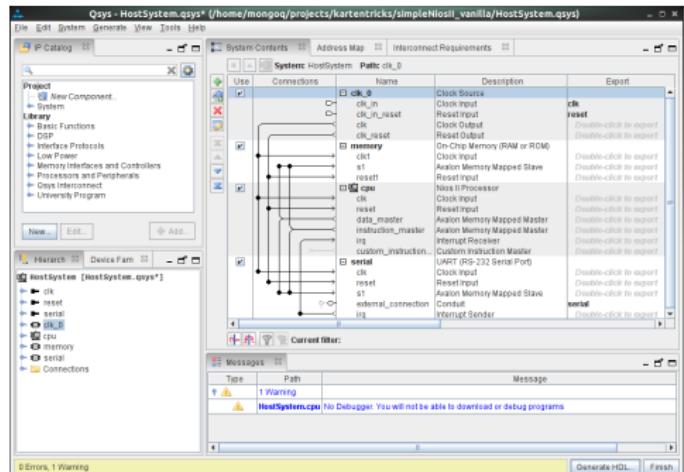
- Project Navigator:** Shows the project structure under "Cyclone IV E: EP4CE15F17C8". The "ov7670_top.vhd" file is selected, showing its hierarchical components: debounce_btt_debounce, ov7670_capture_capture, ov7670_controller_controller, frame_bufferfb, ug419t.vhd, and sys_vuart.
- Hierarchy:** A tree view of the project's hierarchy.
- Tasks:** A list of compilation tasks: Compile Design, Analysis & Synthesis, Filter (Place & Route), Assembler (Generate programming), and TimeQuest Timing Analysis. All tasks are marked as completed with green checkmarks.
- Module:** A status bar at the bottom showing progress, time, and memory usage.
- Code Editor:** The right pane displays VHDL code for the "ov7670_top.vhd" module. The code includes imports for IEEE and STD libraries, a state machine declaration, and a process block for the entity. A link to a Stack Overflow question is also present in the code.
- Messages:** A bottom pane showing a list of messages, with tabs for All, ID, and Message.

Quartus Prime IDE

QSYS

QSYS

- Instanziieren von:
Softcores (NIOS II), RAM,
Schnittstellen (JTAG / UART)
- Grafisches Verbinden der
Funktionsblöcke



QSYS

QSYS

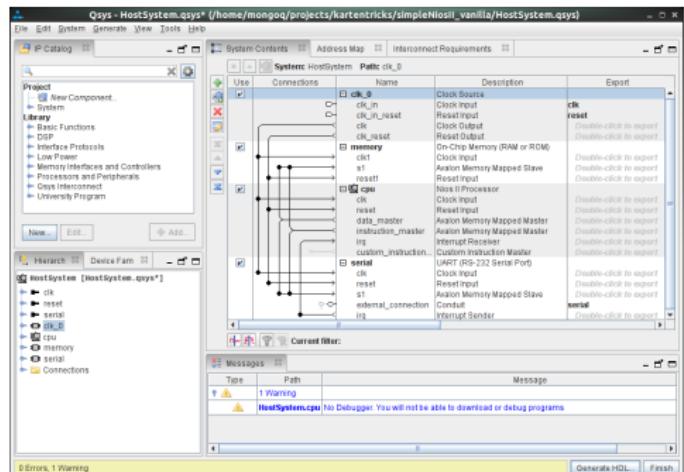
QSYS

- Instanziieren von:

Softcores (NIOS II), RAM,
Schnittstellen (JTAG / UART)

- Grafisches Verbinden der
Funktionsblöcke

- Eclipse basierte
Programmierumgebung



QSYS

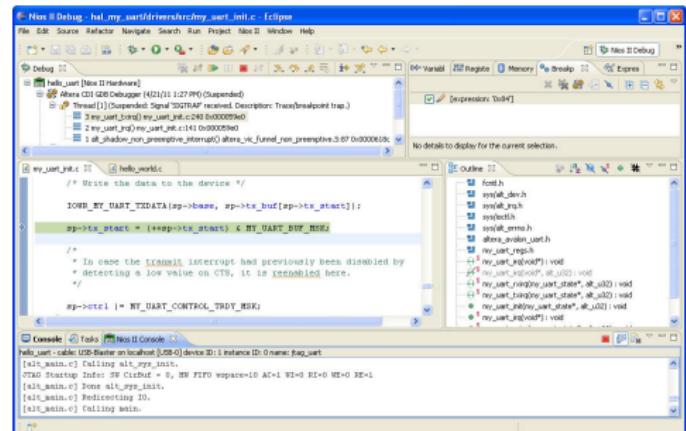
QSYS

QSYS

- Instanziieren von:

Softcores (NIOS II), RAM,
Schnittstellen (JTAG / UART)

- Grafisches Verbinden der
Funktionsblöcke
- Eclipse basierte
Programmierumgebung
- Beispielsystem

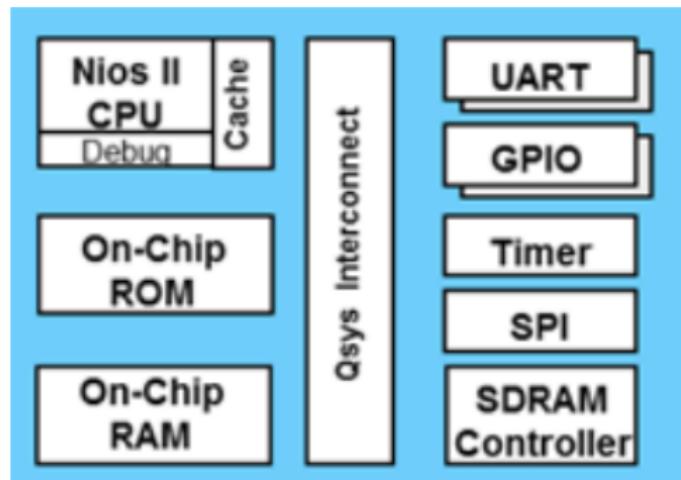


Nios II Embedded Design Suite (Eclipse)

QSYS

QSYS

- Instanziieren von:
Softcores (NIOS II), RAM,
Schnittstellen (JTAG / UART)
- Grafisches Verbinden der
Funktionsblöcke
- Eclipse basierte
Programmierumgebung
- Beispielsystem

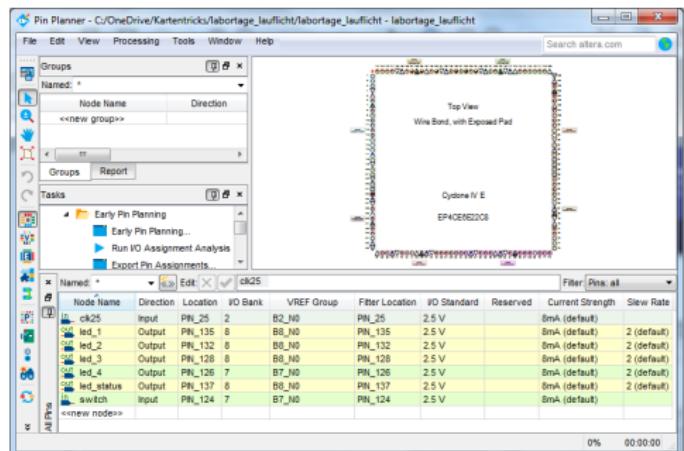


Beispielsystem (SoC - System-on-Chip)

Pin Planner

Pin Planner

- Zuordnung von logischen Ein- und Ausgängen zu physikalisch vorhandenen Pins

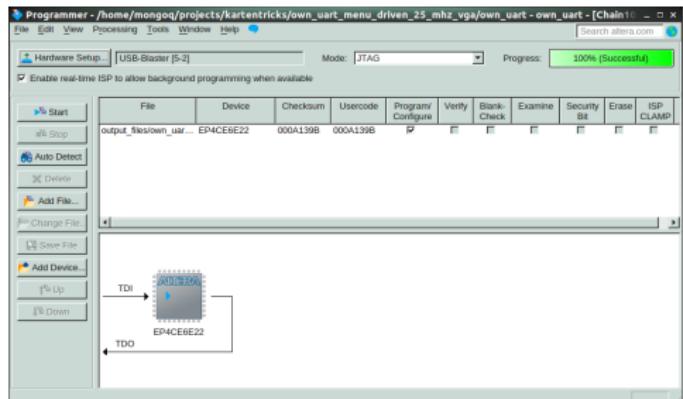


Pin Planner

JTAG Programmer

JTAG Programmer

- Flashen der FPGA Konfigurationsdatei - des Bitstreams
- Wahlweise in FPGA (flüchtig) oder Serial-Flash (permanent)

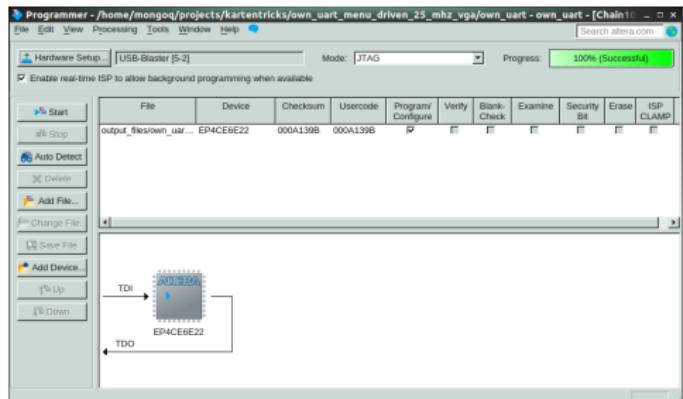


JTAG Programmer

JTAG Programmer

JTAG Programmer

- Flashen der FPGA Konfigurationsdatei - des Bitstreams
- Wahlweise in FPGA (flüchtig) oder Serial-Flash (permanent)

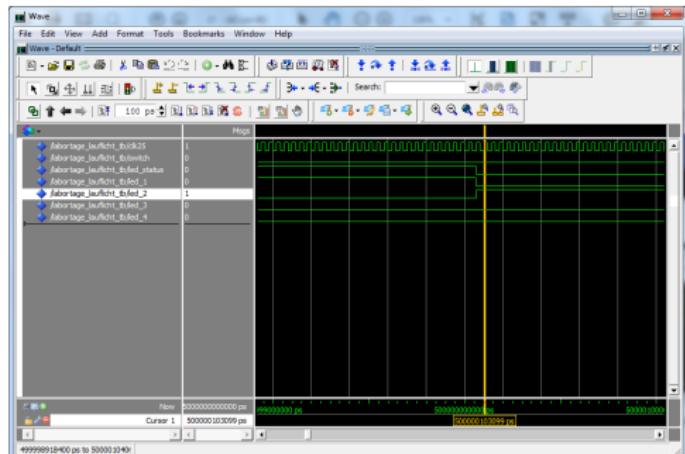


JTAG Programmer

ModelSim

ModelSim

- Simulation eines FPGA Programms
- Nutzung von Testbenches mit Stimuli

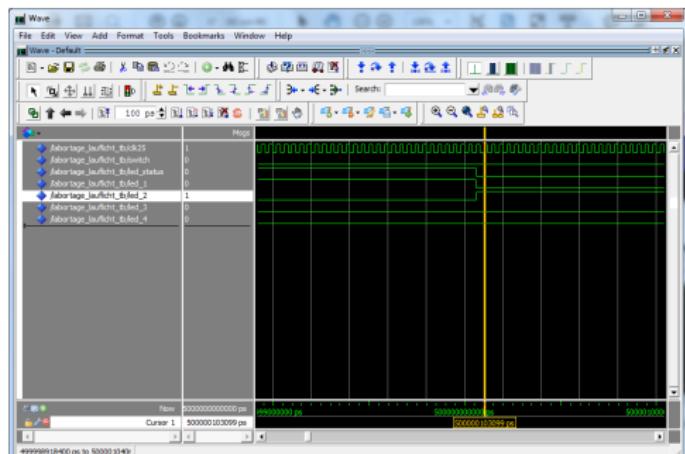


ModelSim

ModelSim

ModelSim

- Simulation eines FPGA Programms
- Nutzung von Testbenches mit Stimuli
- Ausgabe als Pegelwechsel-Diagramm oder Text

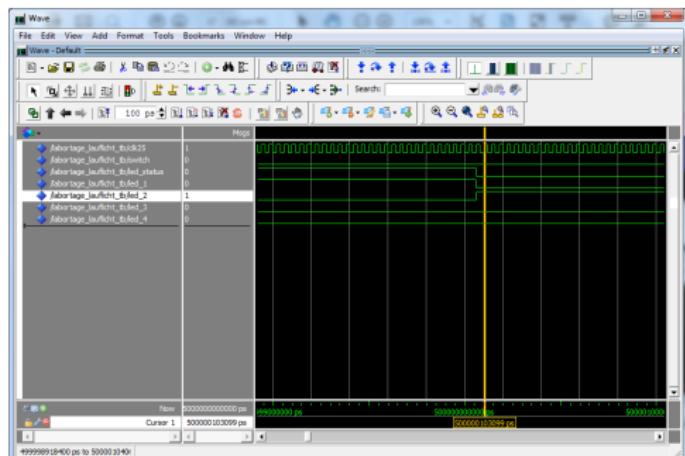


ModelSim

ModelSim

ModelSim

- Simulation eines FPGA Programms
- Nutzung von Testbenches mit Stimuli
- Ausgabe als Pegelwechsel-Diagramm oder Text
- Simulation erfolgt nicht in Echtzeit

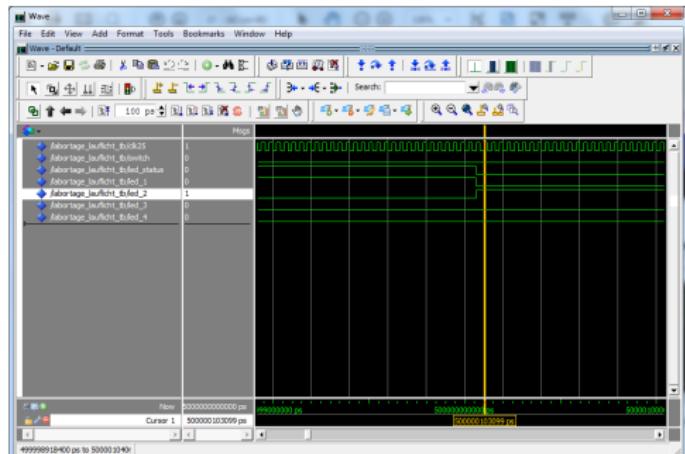


ModelSim

ModelSim

ModelSim

- Simulation eines FPGA Programms
- Nutzung von Testbenches mit Stimuli
- Ausgabe als Pegelwechsel-Diagramm oder Text
- Simulation erfolgt nicht in Echtzeit



ModelSim

Programmierung

VHDL (Europa)

Hardwarebeschreibungssprache
Angelehnt an Ada

```
2 process  ({S0,S1},A,B,C,D)
3 begin
4     case {S0,S1}, is
5         when "00" => Y <= A;
6         when "01" => Y <= B;
7         when "10" => Y <= C;
8         when "11" => Y <= D;
9         when others => Y <= A;
10    end case;
11 end process;
```

Beispiel Code

Verilog (USA)

Hardwarebeschreibungssprache
Angelehnt an C

```
2
3 always @({S0,S1}, A, B, C, D)
4     case ({S0,S1})
5         2'b00: Y = A;
6         2'b01: Y = B;
7         2'b10: Y = C;
8         2'b11: Y = D;
9     endcase
10
```

Beispiel Code

Demonstration - Live!

Demonstration von Beispielcode

- Libraries
- Entity und Architecture
- Ein- und Ausgänge
- LED Auswahl
- Counter (zeitabhängig)
- Case Statement (Finite State Machine)
- Pin Planner
- JTAG Download
- ModelSim Simulation



Box mit Lauflicht

Fragen?

Vielen Dank für Eure Aufmerksamkeit!

Hat vielleicht noch jemand Fragen?

