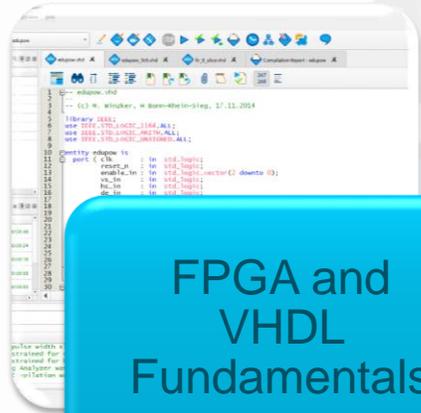




# WORKING WITH THE FPGA VISION REMOTE LAB

**Andrea Schwandt**  
**Institute of Visual Computing, Bonn-Rhein-Sieg University**





FPGA and  
VHDL  
Fundamentals  
–  
a Short  
Overview

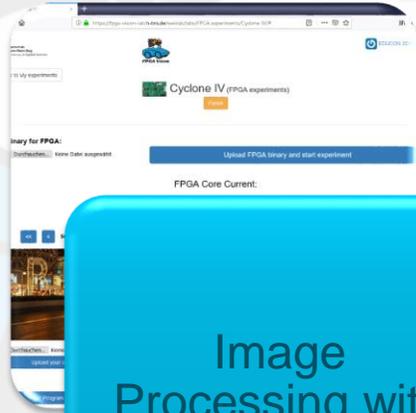
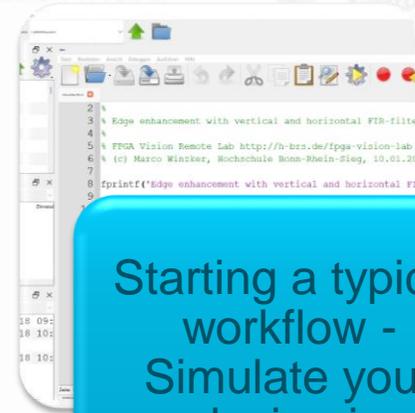


Image  
Processing with  
FPGAs



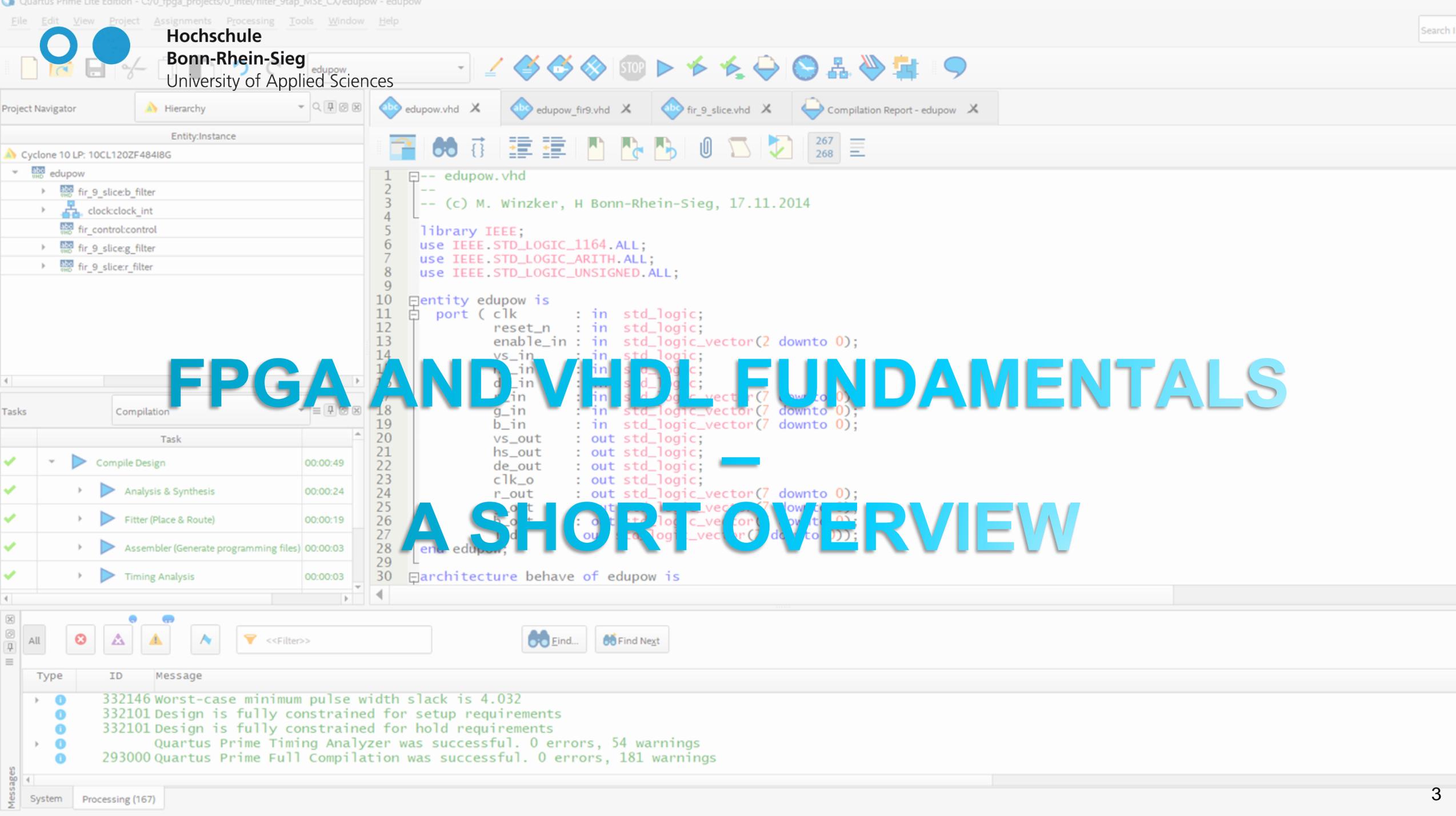
Starting a typical  
workflow -  
Simulate your  
design in  
Octave  
and Questa



Setting up a  
project in  
Quartus and test  
the design with  
the FPGA Vision  
Remote Lab

# FPGA Vision





# FPGA AND VHDL FUNDAMENTALS

## A SHORT OVERVIEW

```
1  -- edupow.vhd
2  --
3  -- (c) M. Winzker, H Bonn-Rhein-Sieg, 17.11.2014
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 entity edupow is
11     port ( clk           : in  std_logic;
12           reset_n       : in  std_logic;
13           enable_in     : in  std_logic_vector(2 downto 0);
14           vs_in         : in  std_logic;
15           r_in          : in  std_logic;
16           d_in          : in  std_logic;
17           t_in          : in  std_logic_vector(7 downto 0);
18           g_in         : in  std_logic_vector(7 downto 0);
19           b_in         : in  std_logic_vector(7 downto 0);
20           vs_out        : out std_logic;
21           hs_out        : out std_logic;
22           de_out        : out std_logic;
23           clk_o         : out std_logic;
24           r_out         : out std_logic_vector(7 downto 0);
25           t_out         : out std_logic_vector(7 downto 0);
26           h_out        : out std_logic_vector(7 downto 0);
27           d_out        : out std_logic_vector(7 downto 0);
28           b_out        : out std_logic_vector(7 downto 0);
29 end edupow;
30
31 architecture behave of edupow is
```

Tasks

Task	Task	Time
✓	Compile Design	00:00:49
✓	Analysis & Synthesis	00:00:24
✓	Fitter (Place & Route)	00:00:19
✓	Assembler (Generate programming files)	00:00:03
✓	Timing Analysis	00:00:03

Messages

Type	ID	Message
!	332146	Worst-case minimum pulse width slack is 4.032
!	332101	Design is fully constrained for setup requirements
!	332101	Design is fully constrained for hold requirements
!		Quartus Prime Timing Analyzer was successful. 0 errors, 54 warnings
!	293000	Quartus Prime Full Compilation was successful. 0 errors, 181 warnings

# Where do we use FPGAs?

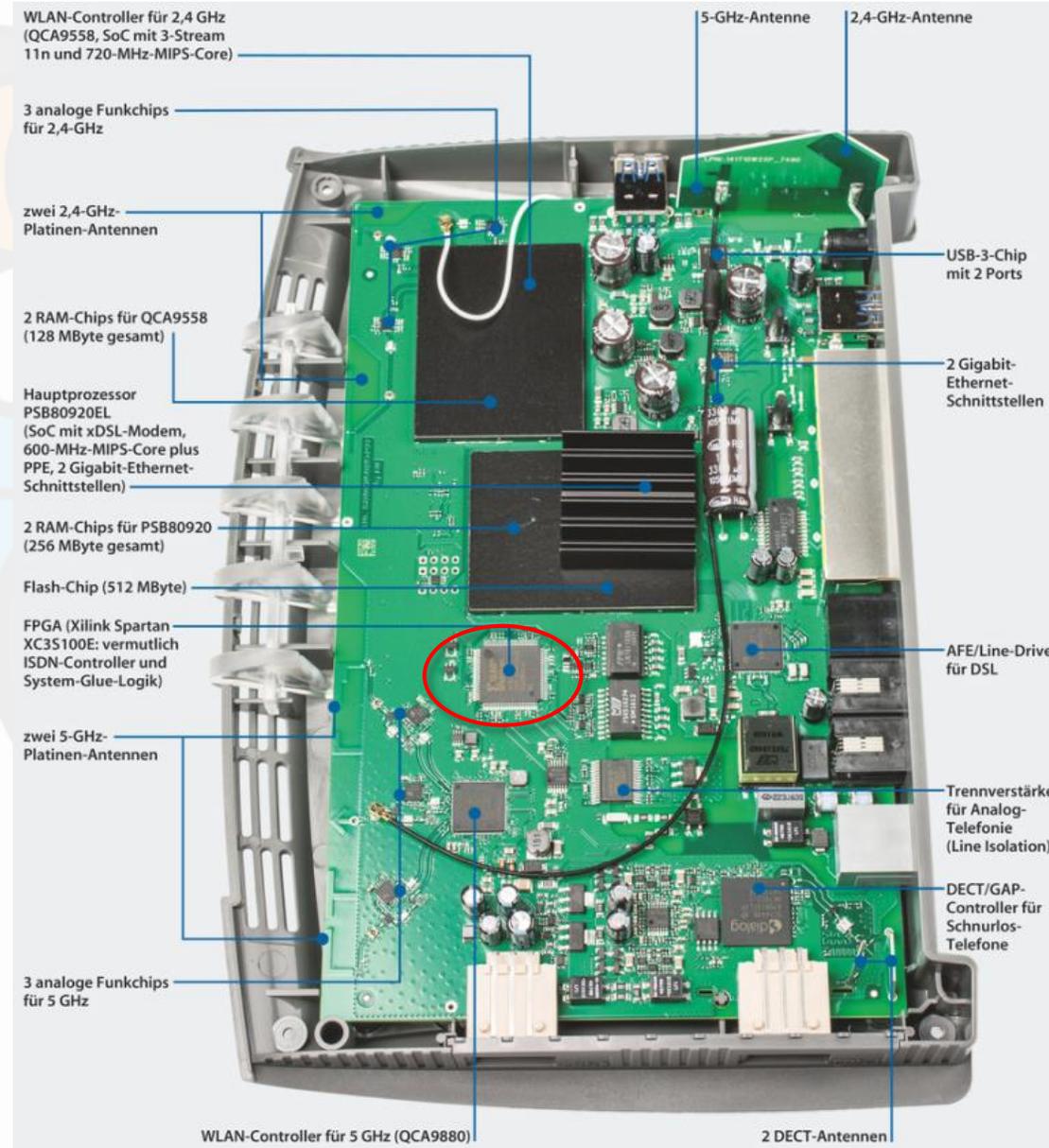
- as glue-logic
- for video processing
- for automotive applications
- ...



*FPGA Vision*



## AVM FritzBox 7490 from c't 17/2016, p. 162



# FPGA

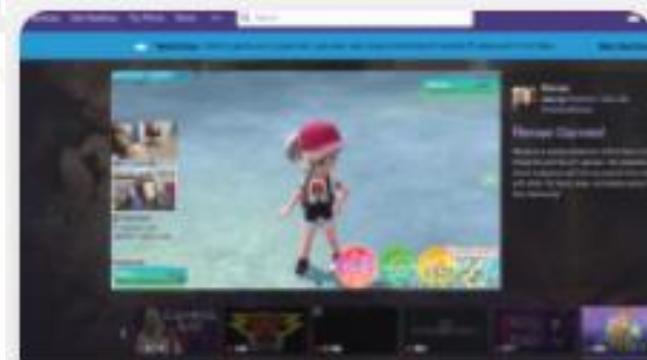




# Video processing



Nextera  
Video Solutions

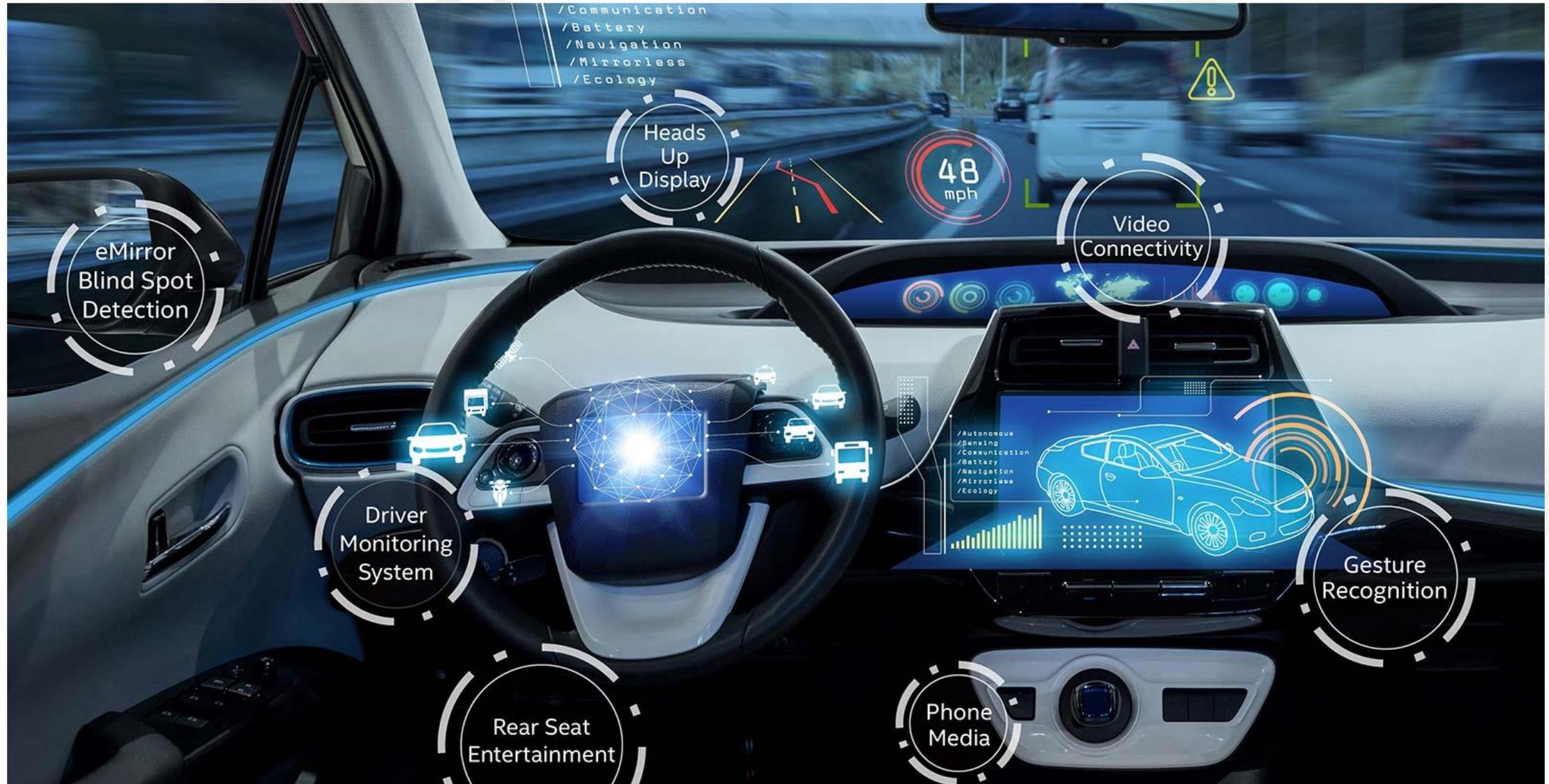


Twitch

Live Video Streaming Powered  
by Xilinx FPGA Acceleration  
and NGCodec VP9 Transcoder



<https://www.xilinx.com/about/xilinx-go/powered-by-xilinx.html>



source: <https://www.intel.de/content/www/de/de/automotive/products/programmable/applications.html>



# HACKADAY

## DE10-LITE-FUL FPGA DEV BOARD HACK PLAYS THE 1981 CLASSIC DEFENDER

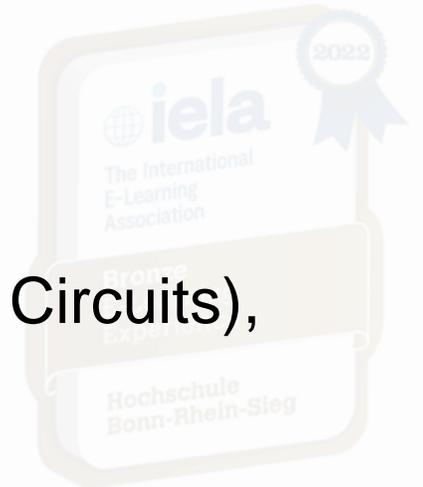
by: Ryan Flowers 6 Comments  
January 8, 2022



VHDL is a **hardware description language** for **design** and **verification** of digital circuits



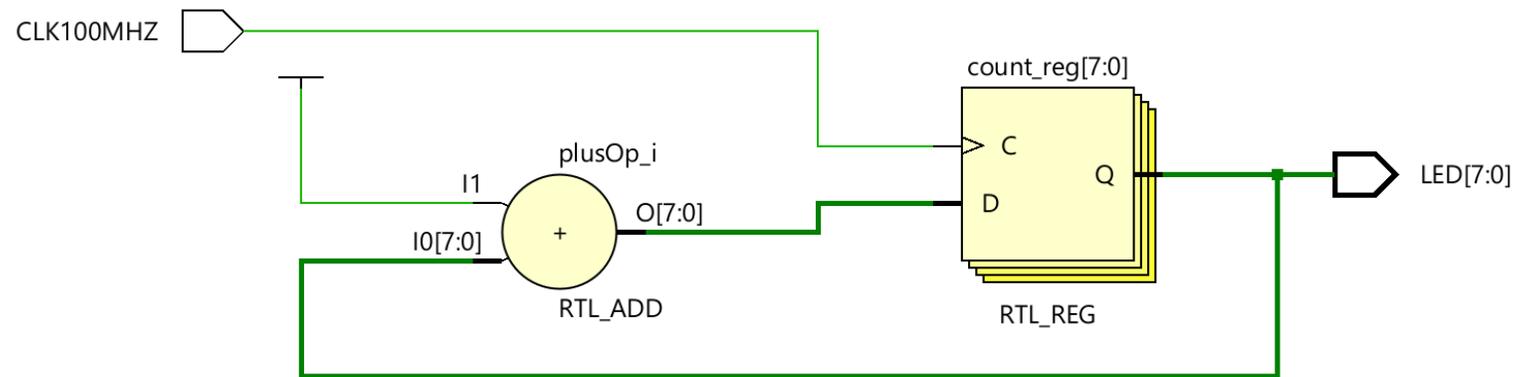
- HDL stands for **H**ardware **D**escription **L**anguage
- V stands for the initiative „VHSIC“ (Very High Speed Integrated Circuits), where VHDL originated



*FPGA Vision*

```
process  
begin  
    wait until rising_edge (CLK100MHZ);  
  
    count <= count + 1;  
  
end process;  
  
LED <= std_logic_vector(count(7 downto 0));
```

- VHDL describes the function of a circuit
- With the VHDL description the design/development suite creates the circuit



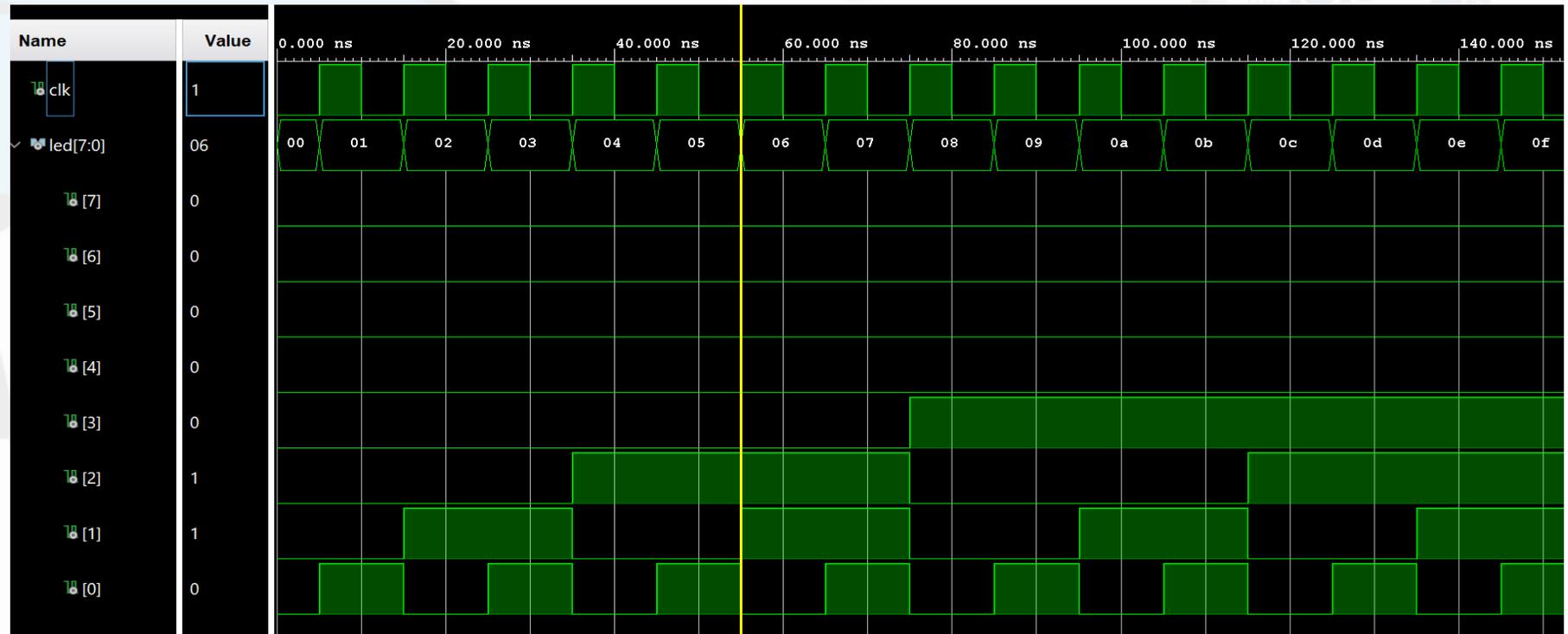


```
-- clock generation
clk <= not clk after 5 ns;

count_1: entity work.zaehler
  port map (CLK100MHZ => clk,
           LED       => led);
```

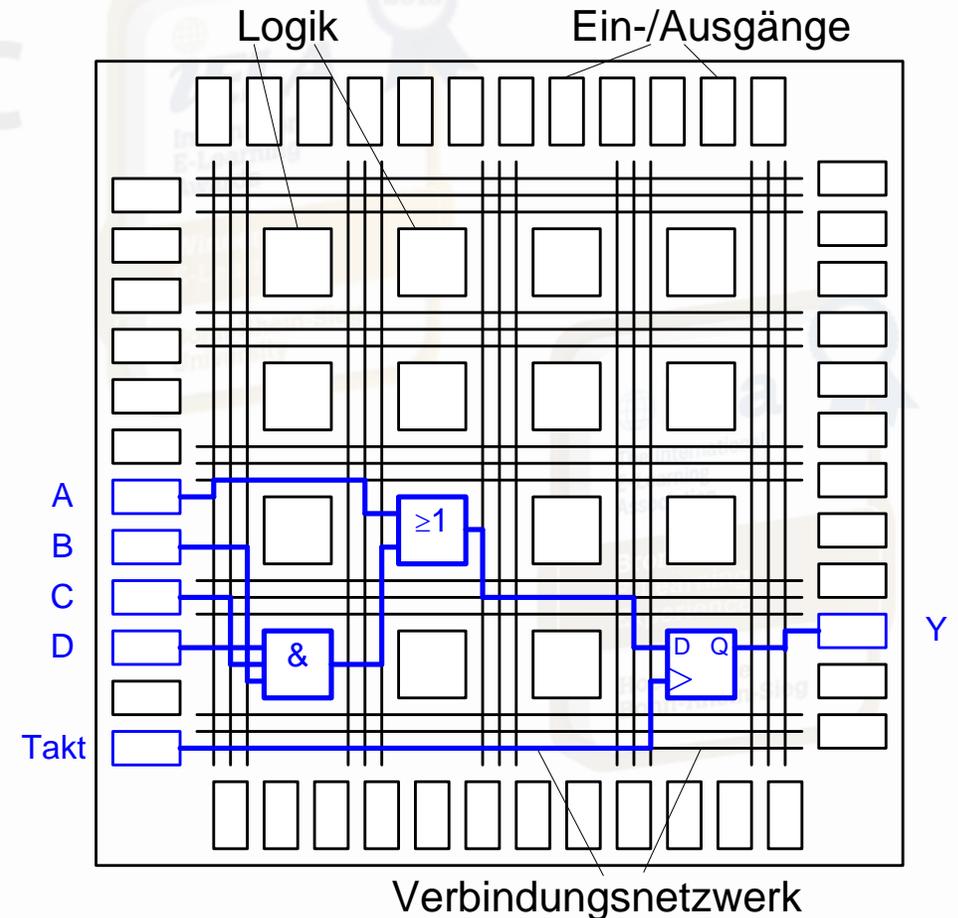
- Verifies the correct behavior of the circuit
- Testbench provides the I/O signals

- Waveform viewer to verify the results



The „Field-Programmable-Gate-Array“ is a microchip containing a programmable circuit

- **field-programmable:** programmable while in use
- **gate-array:** array of gates with
  - some thousands logic functions (AND, OR, EXOR, ...)
  - some thousands memory elements (Registers, Flip-Flops, FFs)
  - connection network
- function and connections are programmable (in **blue**)

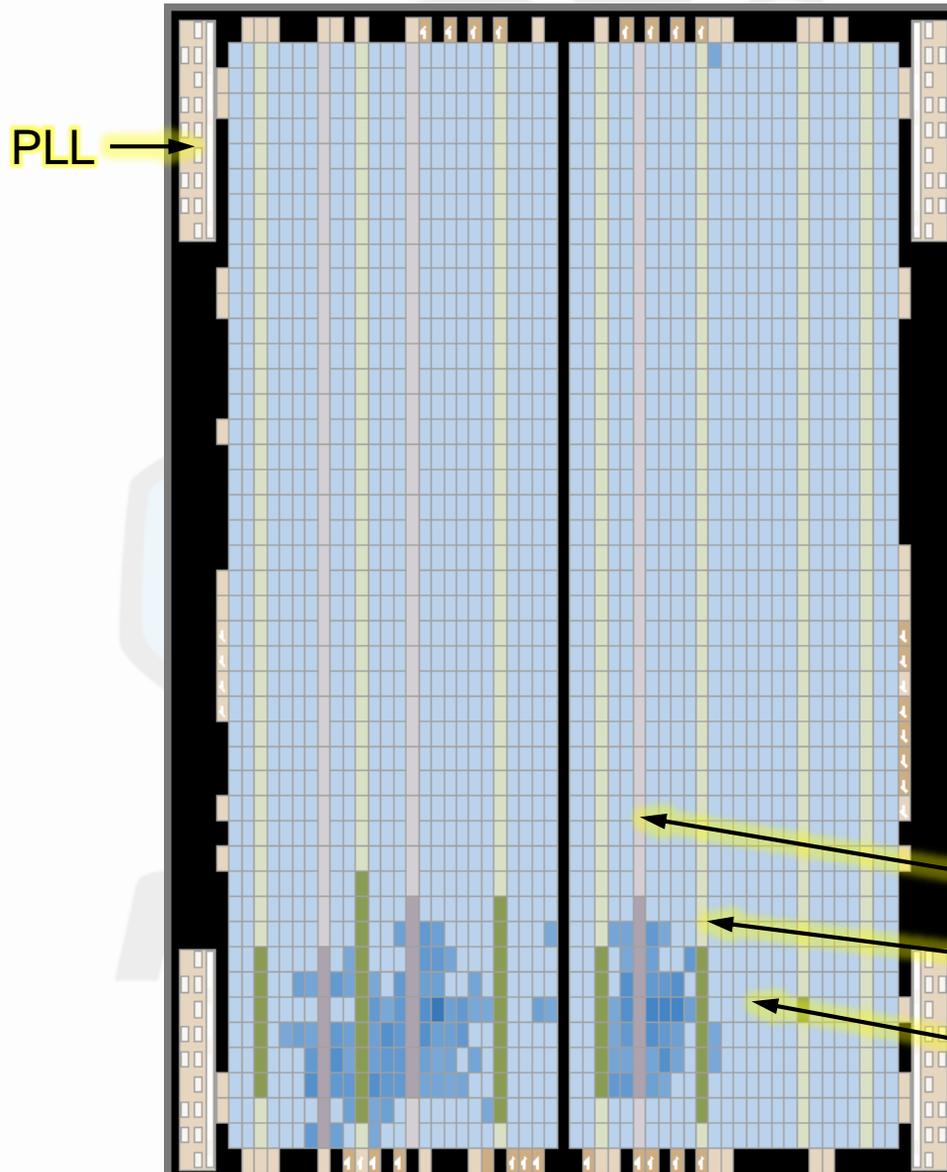


Simplified presentation: Real FPGAs consist of much more logicblocks and I/Os





# Elements of an FPGA (Cyclone V)

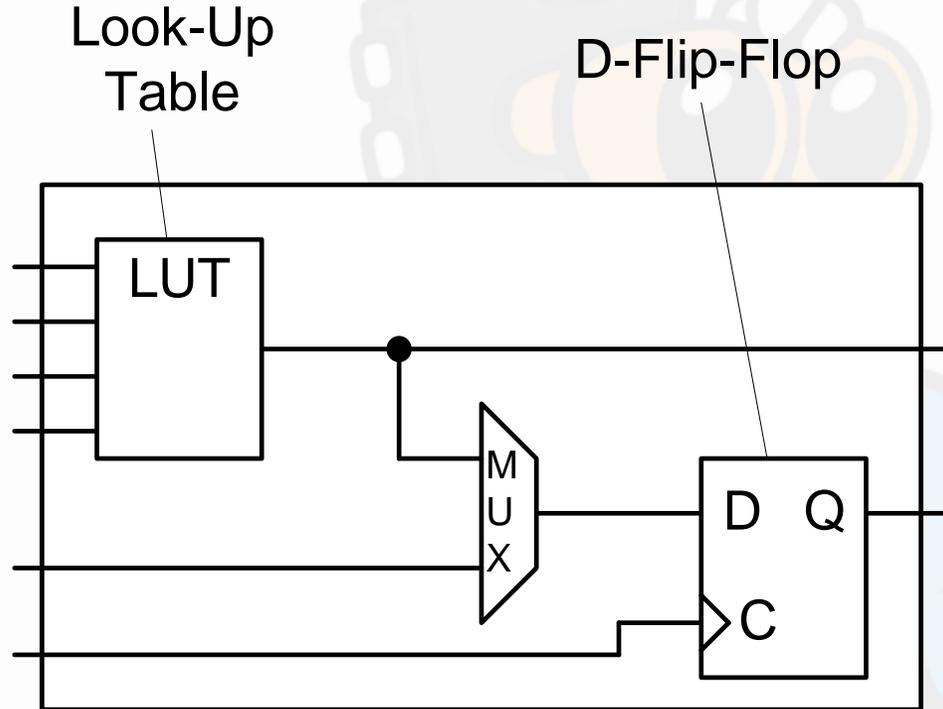


Device	5CEBA2F17C6*
Family	Cyclone V
Device Extents	55 x 46
ALMs	9430
Total I/Os	128
GPIOs	128
GXB Channel PMA	0
GXB Channel PCS	0
PCIe Hard IP Blocks	0
Memory Controllers	0
Memory Bits	1802240
DSP Blocks	25
Fractional PLLs	4
DLLs	4
Global Clocks	16
HPS CPU Cores	0

Flow Status	Successful - Thu Oct 08 18:41:07 2020
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	edupow
Top-level Entity Name	edupow
Family	Cyclone V
Device	5CEBA2F17C6
Timing Models	Final
Logic utilization (in ALMs)	536 / 9,430 ( 6 % )
Total registers	1030
Total pins	63 / 128 ( 49 % )
Total virtual pins	0
Total block memory bits	393,216 / 1,802,240 ( 22 % )
Total DSP Blocks	12 / 25 ( 48 % )
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 4 ( 0 % )
Total DLLs	0 / 4 ( 0 % )

DSP  
Memory  
LE (ALM)

# Logikblocks/Logic elements (LE)



- Logic elements of an FPGAs consist of combinatorial logic and a register or Flip-Flop (FF)
- Name: LE („Logic Element“)
- The combinatorial logic is realized as a Look-Up-Table (LUT)

- Registers can save the results of an LUT or be used independent of the LUT



# Example: Simple module in VHDL

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  -- Uncomment the following library declaration if using
5  -- arithmetic functions with signed or unsigned values
6  -- use IEEE.NUMERIC_STD.ALL;
7
8
9
10 entity taster_leds is
11     Port ( SW : in STD_LOGIC_VECTOR (9 downto 0);
12           LEDR : out STD_LOGIC_VECTOR (9 downto 0));
13 end taster_leds;
14
15 architecture Behavioral of taster_leds is
16
17 begin
18     LEDR <= SW;
19
20 end Behavioral;
21
22
```

call packages with definitions of datatypes

This is the call, not the definition of the package!

entity description of module „taster\_leds“ with 10 inputs SW0 – SW9 and 10 outputs LEDR0 –LEDR9

architecture description of module „taster\_leds“  
Condition of switches SW0 – SW9 are written on LEDs LEDR0 – LEDR9

concurrent instruction

Keywords are printed in blue

FPGA VISION

# VHDL describes parallel processing

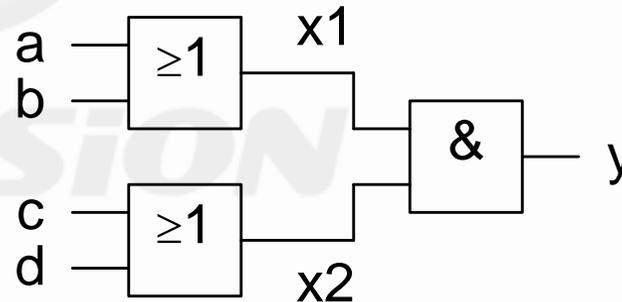
- Multiple concurrent commands are executed in parallel
- The **order** of the commands is **not relevant!**
- The following commands result in the **same behavior**:

```
x1 <= a or b;  
x2 <= c or d;  
y <= x1 and x2;
```

```
y <= x1 and x2;  
x1 <= a or b;  
x2 <= c or d;
```

```
x1 <= a or b;  
y <= x1 and x2;  
x2 <= c or d;
```

- All three descriptions result into the same circuit:





# Overview of VHDL commands

```
process
begin
  wait until rising_edge(clk);

  -- input FFs for control
  reset <= not reset_n;
  enable <= enable_in;
  -- input FFs for video signal
  vs_0 <= vs_in;
  hs_0 <= hs_in;
  de_0 <= de_in;
  r_0 <= to_integer(unsigned(r_in));
  g_0 <= to_integer(unsigned(g_in));
  b_0 <= to_integer(unsigned(b_in));
end process;
```

```
if (reset = '1') then
  wr_address := 0;
  rd_address := 1;
elsif (write_en = '1') then
  wr_address := rd_address;
  if (rd_address = 1279) then
    rd_address := 0;
  else
    rd_address := rd_address + 1;
  end if;
end if;
```

```
process
begin
  wait until rising_edge(clk);
  h_tap(0) <= v_out;
  for i in 0 to 7 loop
    h_tap(i+1) <= h_tap(i);
  end loop;
end process;
```

```
process(bin)
begin
  case bin is
    when "0000" => seg7_invert <= "0111111"; -- 0
    when "0001" => seg7_invert <= "0000110"; -- 1
    when "0010" => seg7_invert <= "1011011"; -- 2
    when "0011" => seg7_invert <= "1001111"; -- 3
    when "0100" => seg7_invert <= "1100110"; -- 4
    when "0101" => seg7_invert <= "1101101"; -- 5
    when "0110" => seg7_invert <= "1111101"; -- 6
    when "0111" => seg7_invert <= "0000111"; -- 7
    when "1000" => seg7_invert <= "1111111"; -- 8
    when "1001" => seg7_invert <= "1101111"; -- 9
    when "1010" => seg7_invert <= "1110111"; -- A
    when "1011" => seg7_invert <= "1111100"; -- B
    when "1100" => seg7_invert <= "0111001"; -- C
    when "1101" => seg7_invert <= "1011110"; -- D
    when "1110" => seg7_invert <= "1111001"; -- E
    when "1111" => seg7_invert <= "1110001"; -- F
    when others => seg7_invert <= "1000000"; -- '-' for rest
  end case;
end process;
```

## VHDL is a hardware-design language

- When you are working with VHDL, you are not programming, you are “designing hardware”. Your VHDL code should reflect this fact.
- If your VHDL code appears too similar to code of a higher-level computer language, it is probably bad VHDL code.

## Have a general concept of what your hardware should look like

- Although VHDL is vastly powerful, if you do not understand basic digital constructs, you will probably be unable to generate efficient digital circuits.
- If you are not able to roughly envision the digital circuit you are trying to model in terms of basic digital circuits, you will probably misuse VHDL.
- VHDL is cool, but it is not as magical as it initially appears to be.

Aus: B. Mealy, F. Tappero, “Free Range VHDL,” <http://freerangefactory.org>



Finish

Binary for FPGA:

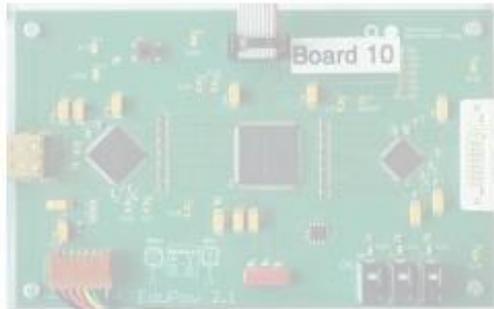
Durchsuchen... Keine Datei ausgewählt.

Upload FPGA binary and start experiment

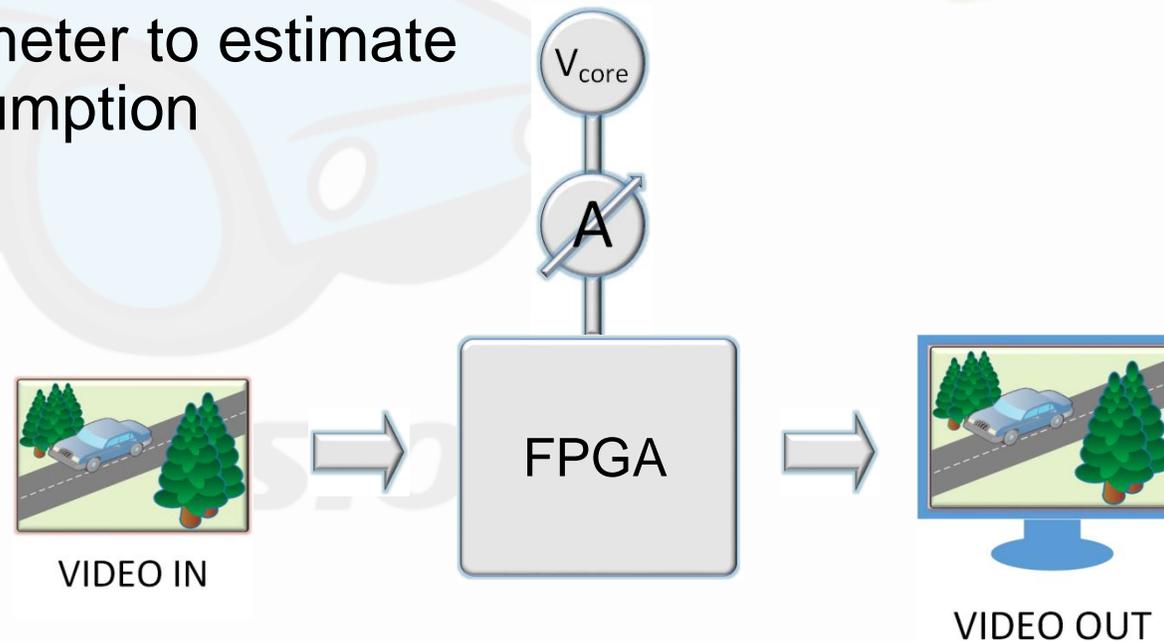
# IMAGE PROCESSING WITH FPGAS

FPGA Core Current:  
**96.6 mA**  
(Core Supply Voltage is 1.2 V)

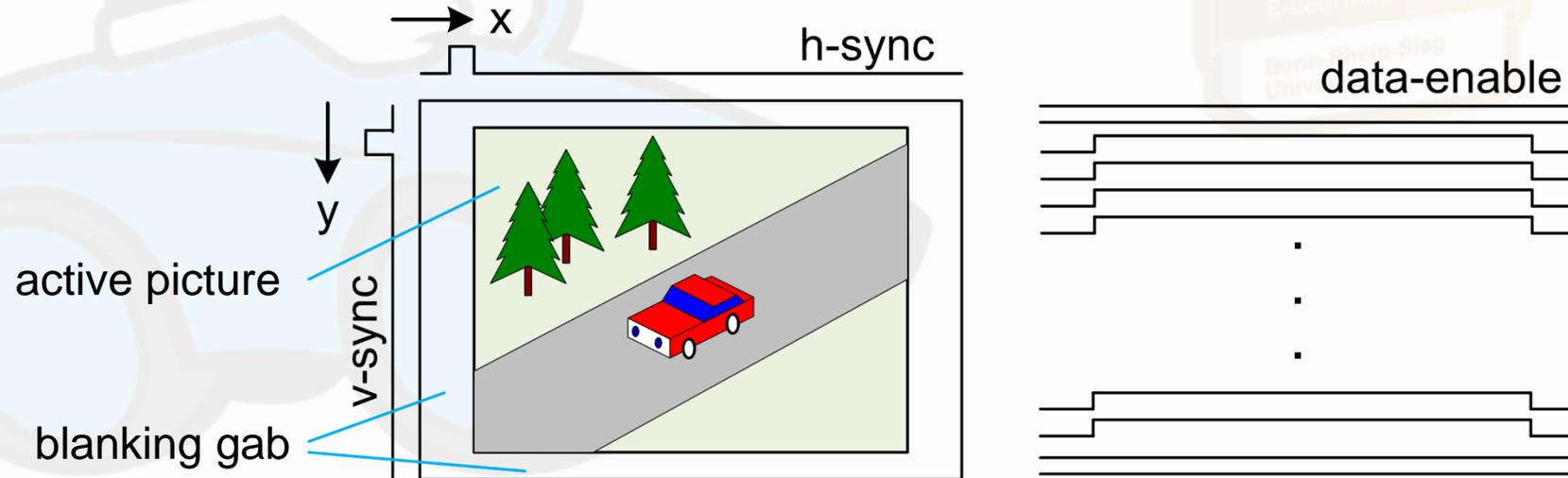
<< < Select Input Image > >>



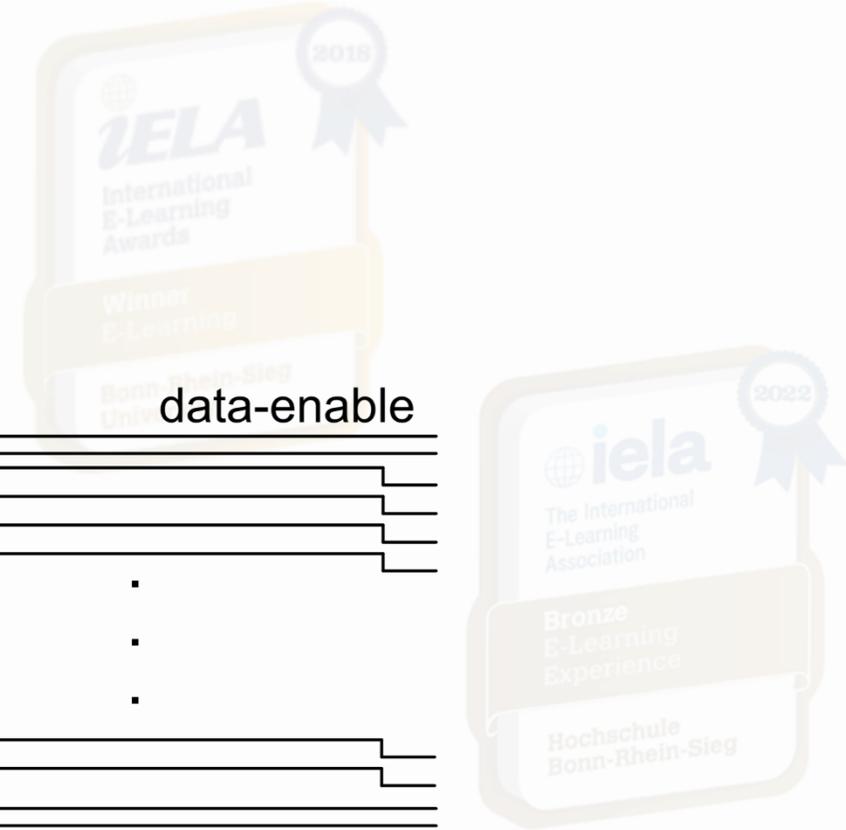
- A typical setup for image processing could consist of
  - a CMOS device  
e.g., an FPGA
  - video in- and output
  - and possible additional components  
⇒ e.g., an ammeter to estimate power consumption



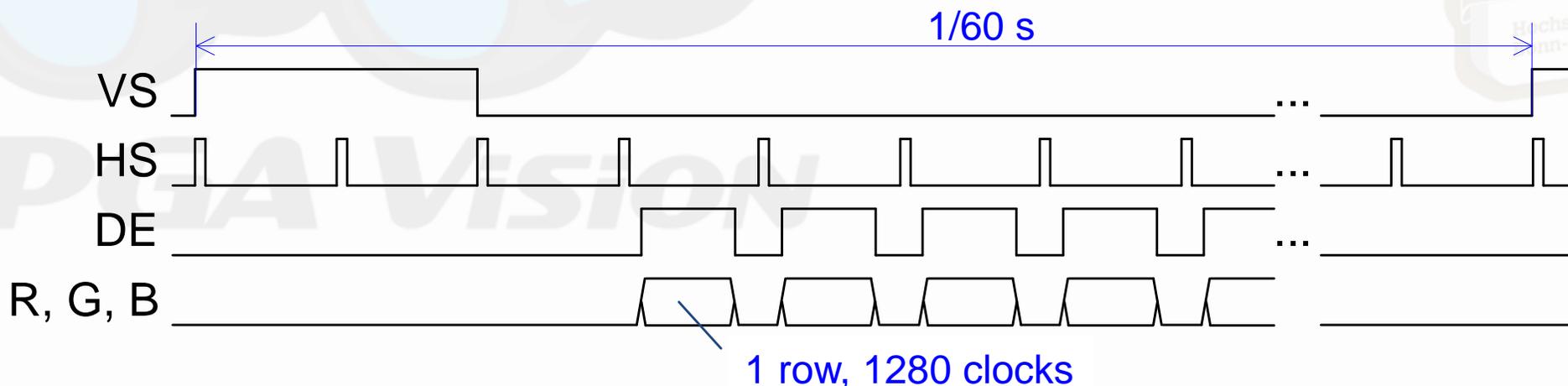
- Output of video signals row by row
- There is an inactive arial horizontally und vertically
  - Control signals v-sync, h-sync, data-enable



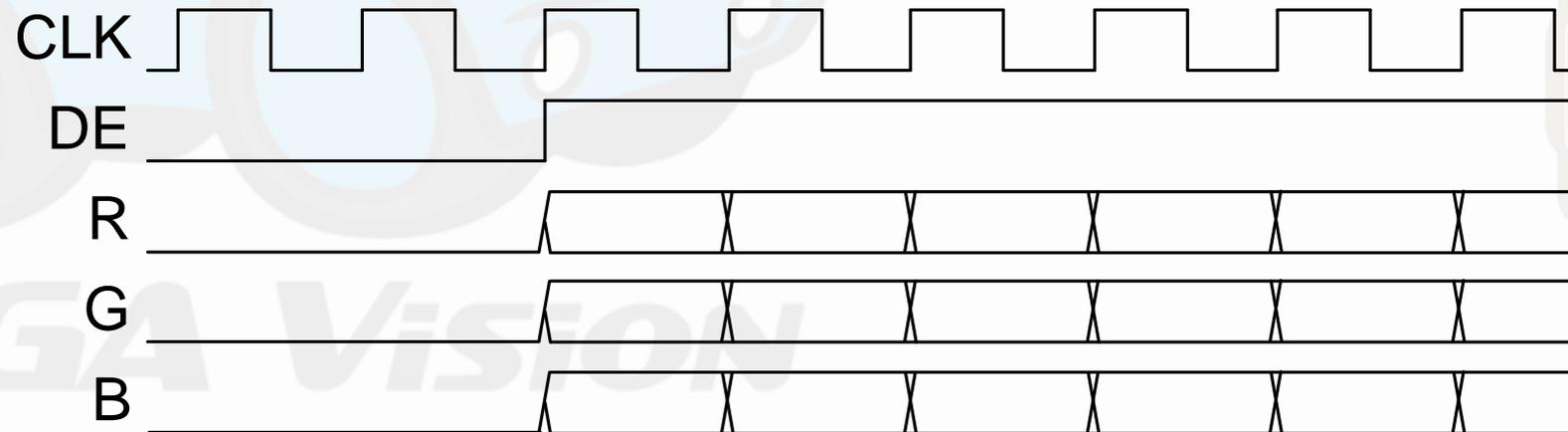
- Example: HDTV graphic format 720p
  - 1280\*720 Pixel, 60 Hz image refresh rate
  - Vertical blanking gab                    30 rows,            total of 750 rows
  - Horizontal blanking gab            370 pixel,            total of 1650 pixel



- 60 pictures/s \* 750 rows \* 1650 pixel result in clock frequency of 74,25 MHz
- Signalization of active pixel by three control signals
  - VS, v-sync: Start of picture
  - HS, h-sync: Start of row
  - DE, data-enable: Active pixel
- Color information R, G, B with 8 bit each



- R, G, B every clock
  - R, G, B values as binary number between 0 and 255, respectively 8 bit per color
  - 0 = no color
  - 255 = maximum color



- There are VHDL-templates available
- **Inputs** are
  - CLK – clock of 74,25 MHz
  - RESET\_N – Low-active reset, inverted in VHDL to the high-active RESET
  - ENABLE\_IN – Three switches
  - VS\_IN, HS\_IN, DE\_IN – control signals of input image
  - R\_IN, G\_IN, B\_IN – Color information input image, 8 bit
- **Outputs** are
  - VS\_OUT, HS\_OUT, DE\_OUT – control signals of output image
  - R\_OUT, G\_OUT, B\_OUT – Color information output image, 8 bit
  - LED – Three LEDs, not implemented in remote lab



Global Online Laboratory Consortium

```
entity edupow is
  port ( clk           : in  std_logic;
        reset_n       : in  std_logic;
        enable_in     : in  std_logic_vector(2 downto 0);
        vs_in         : in  std_logic;
        hs_in         : in  std_logic;
        de_in         : in  std_logic;
        r_in          : in  std_logic_vector(7 downto 0);
        g_in          : in  std_logic_vector(7 downto 0);
        b_in          : in  std_logic_vector(7 downto 0);
        vs_out        : out std_logic;
        hs_out        : out std_logic;
        de_out        : out std_logic;
        r_out         : out std_logic_vector(7 downto 0);
        g_out         : out std_logic_vector(7 downto 0);
        b_out         : out std_logic_vector(7 downto 0);
        clk_o         : out std_logic;
        led           : out std_logic_vector(2 downto 0));
end edupow;
```

**control input**

---

**video input**

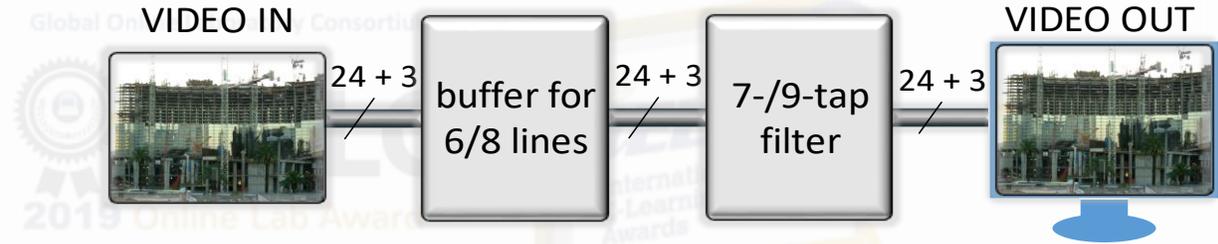
---

**video output**

---

**control output**

- Block diagramm:



- Filter equation (discrete convolution):

$$P(x, y) = \sum_{i=1}^n \sum_{j=1}^n p(x + i - a, y + j - a) \cdot h_{ij}$$

$p(x, y)$  : Pixel at  $(x, y)$

$n$  : No taps

$x$  : horizontal position

$y$  : vertical position

$i$  : horizontal matrix index

$j$  : vertical matrix index

$a$  :  $n/2 + 1$

$h_{ij}$  : Element of convolution matrix at position  $(i, j)$

- Filter matrix (7-tap filter):

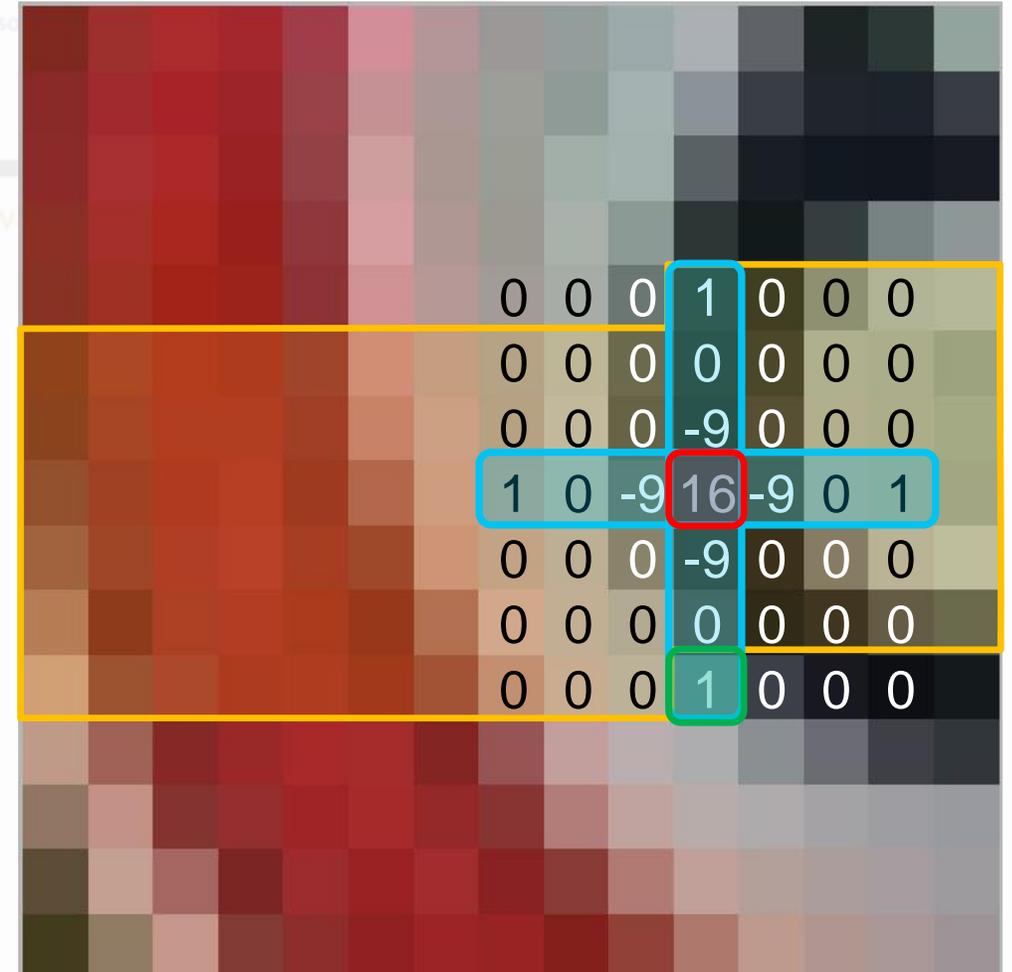
$$h_{7-tap} = \frac{1}{32} \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 \\ 1 & 0 & -9 & 16 & -9 & 0 & 1 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

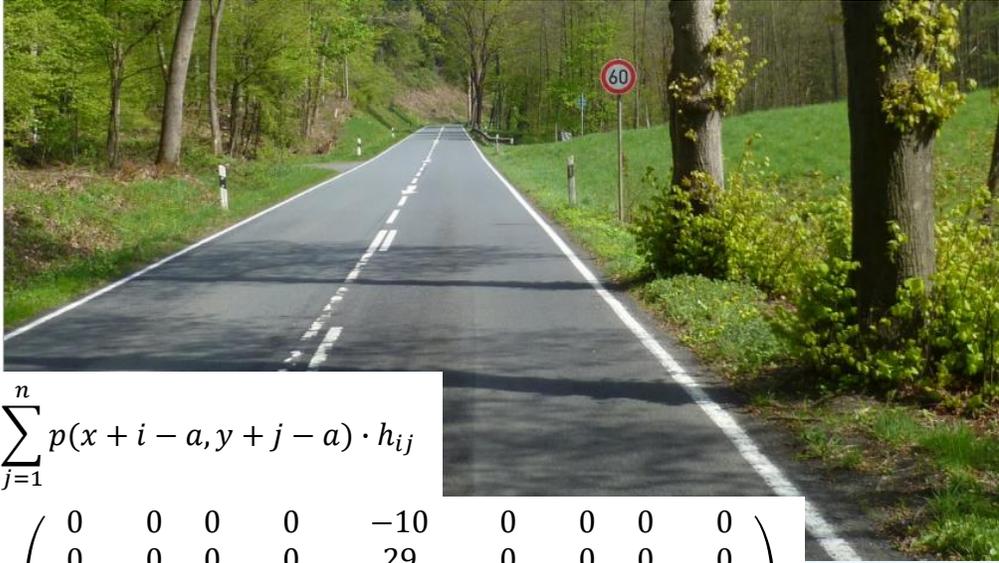


$$P(x, y) = \sum_{i=1}^n \sum_{j=1}^n p(x+i-a, y+j-a) \cdot h_{ij} \quad h_{7\text{-tap}} = \frac{1}{32} \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 \\ 1 & 0 & -9 & 16 & -9 & 0 & 1 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

- current pixel  $p(x, y+3)$  - 3 rows under filtered pixel  $p(x, y)$
- **buffer** 6 rows for 7-tap filter
- Apply filter matrix on  $p(x, y)$
- Apply **effective matrix coefficients**

# 7-Tap FIR Filter

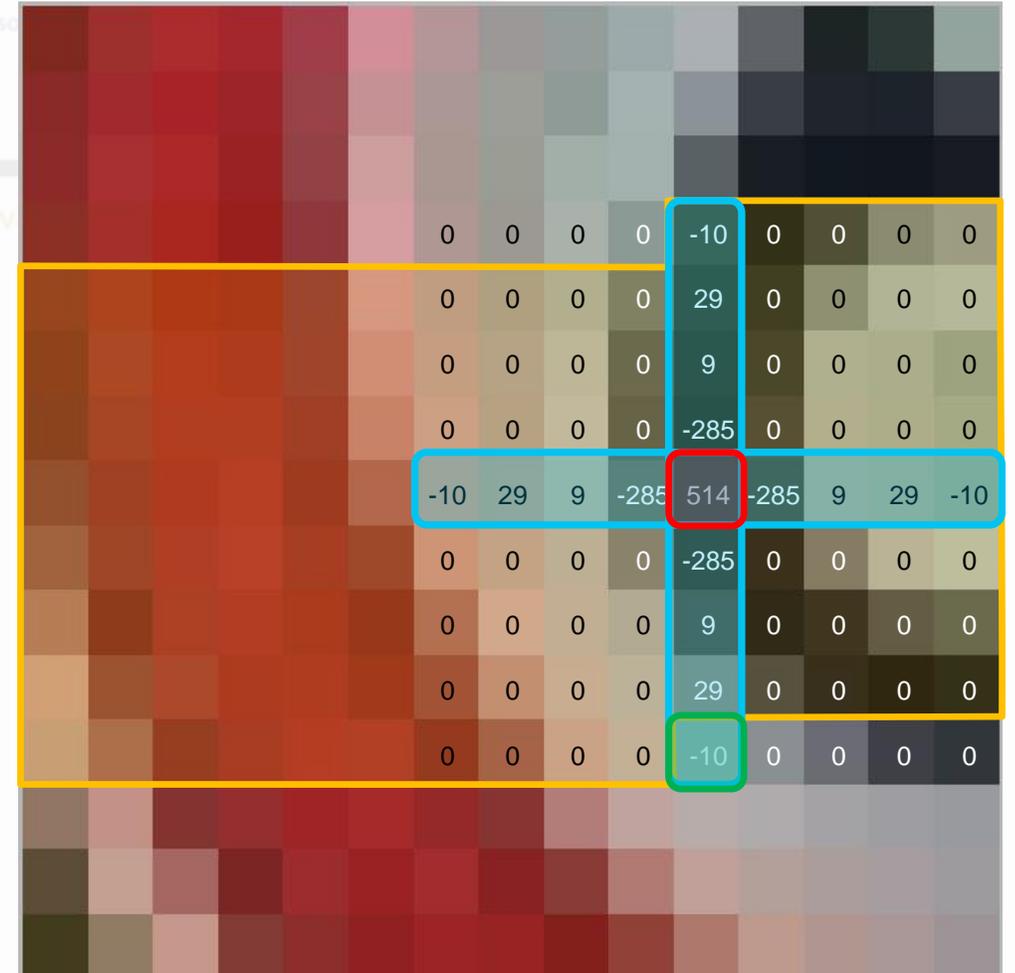




$$P(x, y) = \sum_{i=1}^n \sum_{j=1}^n p(x+i-a, y+j-a) \cdot h_{ij}$$

$$h_{9\text{-tap}} = \frac{1}{1024} \begin{pmatrix} 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -285 & 0 & 0 & 0 & 0 \\ -10 & 29 & 9 & -285 & 514 & -285 & 9 & 29 & -10 \\ 0 & 0 & 0 & 0 & -285 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 29 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# 9-Tap FIR Filter



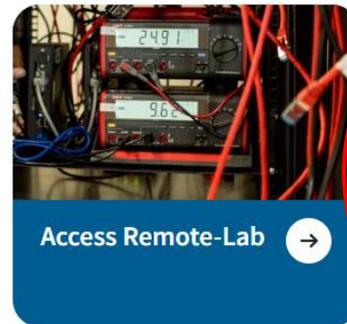
- current pixel  $p(x, y+4)$  - 4 rows under filtered pixel  $p(x, y)$
- buffer 8 rows for 9-tap filter
- Apply filter matrix on  $p(x, y)$
- Apply effective matrix coefficients



# First step: get the templates

- Visit <https://www.h-brs.de/fpga-vision-lab>
- scroll down to

## Learning Material



select

FPGA Vision



# Get the Templates from Github



Marco Winzker  
Marco-Winzker

Follow

Professor for Digital Circuit Design at  
Bonn-Rhein-Sieg University of Applied  
Sciences Head of Centre for Teaching  
Development and Innovation

130 followers · 0 following

Hochschule Bonn-Rhein-Sieg  
St. Augustin, Germany  
[https://www.h-brs.de/en/emt/marco\\_winzker](https://www.h-brs.de/en/emt/marco_winzker)

Achievements

Overview Repositories 6 Projects Packages Stars

## Popular repositories

**NN\_RGB\_FPGA** Public  
FPGA Design of a Neural Network for Color Detection  
VHDL 50 stars 19 forks

**FPGA-Vision** Public  
Learn about image processing with an FPGA. Video lectures explain algorithm and implementation of lane detection for automotive driving. Real hardware is available as a remote lab.  
VHDL 25 stars 15 forks

**NN\_Pattern\_FPGA** Public  
Neural Network for Pattern Recognition on an FPGA. Project for Education. Video lectures explain training of the network and FPGA implementation with VHDL.  
VHDL 12 stars 3 forks

**Spiking\_NN\_RGB\_FPGA**  
FPGA Design of a Spiking Neural Network.  
VHDL 11 stars 2 forks

**FPGA-FIR-Filter** Public  
Lecture about FIR filter on an FPGA  
VHDL 8 stars 4 forks

**Image-Generator-for-FPGA-Evaluation-Board**  
Design of an image generator to represent a street scene. Can be stand-alone design for image generator or as a test pattern generator for lane detection circuit.  
VHDL 7 stars 3 forks

5 contributions in the last year

Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												



master 1 branch 0 tags

Go to file **Code**

Marco-Winzker Delete multipler.vhd	
Docs	Add files via upload
FPGA-Design	Add files via upload
Octave	Add files via upload
Verification	Add files via upload
LICENSE	Add files via upload
README.md	Add files via upload

Local Codespaces

Clone ?

HTTPS GitHub CLI

<https://github.com/Marco-Winzker/FPGA-FIR-F11>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

**Download ZIP**

# FPGA Vision



The screenshot shows the Octave software interface. On the left, a file explorer displays the directory `C:/Users/aschwa3m/Downloads/RL4Eng/FPGA-FIR-Filter-master/Octave` with files: `PB022744_720p.jpg`, `PB022744.JPG`, `sharp_filter_coefficients.m`, `sharp_frequency_response.m`, and `sharp_image_filter.m`. The main window shows a code editor with the following MATLAB code:

```
1 % sharp_image_filter
2 %
3 % Edge enhancement with vertical and horizontal FIR-filter
4 %
5 % FPGA Vision Remote Lab http://h-brs.de/fpga-vision-lab
6 % (c) Marco Winzker, Hochschule Bonn-Rhein-Sieg, 10.01.2020
7
8 fprintf('Edge enhancement with vertical and horizontal FIR-filter\n')
9
10 pkg load image;
11 img_in = imread("Lindau_Harbour_720p.jpg"); % change name for your test image
12
13 f_hor = [1, 0, -9, 48, -9, 0, 1]/32;
14 f_ver = f_hor'; % transpose filter matrix
15
16 img_tmp = imfilter(img_in, f_ver);
17 img_out = imfilter(img_tmp, f_hor);
18
19 imwrite(img_out, "Lindau_Harbour_sharp.jpg"); % change name for your test image
20
21
22
23
```

Annotations in the image include:

- A red circle around the `"Lindau_Harbour_720p.jpg"` string in line 11, with a red '1.' next to it.
- A red circle around the `"Lindau_Harbour_sharp.jpg"` string in line 19, with a red '1.' next to it.
- A red circle around the Run button (a yellow play icon) in the toolbar, with a red '2.' next to it.
- A blue box with the text `change file names` and two blue arrows pointing to the red circles in lines 11 and 19.

The command window at the bottom shows the Octave version and date: `# Octave 7.2.0, Mon Sep 18 09:55:35 2023 G` and `# Octave 7.2.0, Mon Sep 18 10:07:40 2023 G`.

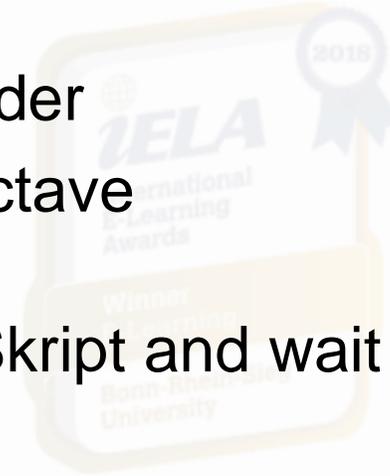


# Prepare files for simulation

- copy converted 720p image into the verification folder
- open „sharp\_generate\_testbench\_images.m“ in Octave (ensure write\_ascii\_ppm.m is in the same folder)
- adjust rows 12, 21 und 22 to own file names, run Skript and wait a bit until files are created



Global Online Laboratory Consortium





# Simulation in Questa/Modelsim

The image shows the ModelSim - INTEL FPGA STARTER EDITION 10.5b interface. The main window displays the 'Welcome to ModelSim' dialog with the following text:

**Welcome to ModelSim**

2. **Create a Project**  
ModelSim Projects contain simulation detail source files, or libraries. The Project Manager references to shared global files. Start here project.

**Open a Project**  
Your most recent project is opened when you if you'd like to open an older project.

The 'Add items to the Project' dialog is open, showing the 'Add Existing File' option circled in red (5). The 'Project Name' field contains 'sim\_sharp' (3). The 'Project Location' field shows 'C:/Users/aschwa3m/ownCloud' (4). The 'Default Library Name' is 'work'. The 'Copy Settings From' dropdown is set to 'modelsim\_ase/modelsim.ini'. The 'Copy Library Mappings' radio button is selected.

The 'Add file to Project' dialog is open, showing the 'File Name' field with the path 'C:/Users/aschwa3m/ownCloud/Projects/sim\_sharp' (6). The 'Add file as type' is 'default' (10). The 'Folder' is 'Top Level'. The 'Copy to project directory' radio button is selected (11). The 'OK' button is circled in red (11).

The 'Select' dialog is open, showing the 'sim\_sharp' folder selected (7). The 'Dateiname' field contains '"sim\_sharp.vhd" "sharp.vhd" "sharp\_arith.vhd"' (8). The 'Dateityp' is 'HDL Files (\*.v,\*.\*.vhd,\*.\*.vho,\*.\*.hdl,\*.\*.vo\*)' (8). The 'Öffnen' button is circled in red (8).

The 'Library' pane on the left shows a list of libraries with their names, types, and paths. The 'Transcript' pane at the bottom shows the error message: 'couldn't open "transcript": permission denied' and the command '# Reading C:/intelFPGA\_lite/19.1/modelsim\_ase/tcl/vsim/pref.tcl'.

1. The 'Jumpstart' button in the 'Welcome to ModelSim' dialog is circled in red.



# Simulation in Questa/Modelsim II

File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks V

ColumnLayout AllColumns

12.

Project - C:/Users/aschwa3m/ownCloud/Projects/sim\_sharp/sim\_sharp

Name	Status	Type	Or
sim_sharp.vhd		VHDL	0
sharp.vhd		VHDL	1
sharp_arith.vhd		VHDL	2
sharp_control.vhd		VHDL	3
sharp_linemem.vh...		VHDL	4
sharp_slice.vhd		VHDL	5

14.

Start Simulation

Design VHDL Verilog Libraries SDF Others

Name	Type	Path
work	Library	C:/Users/aschwa3m/ownCloud/Project...
sharp	Entity	C:/Users/aschwa3m/ownCloud/Project...
sharp_arith	Entity	C:/Users/aschwa3m/ownCloud/Project...
sharp_control	Entity	C:/Users/aschwa3m/ownCloud/Project...
sharp_linemem	Entity	C:/Users/aschwa3m/ownCloud/Project...
sharp_slice	Entity	C:/Users/aschwa3m/ownCloud/Project...
sim_sharp	Entity	C:/Users/aschwa3m/ownCloud/Project...
220model	Library	\$MODEL_TECH/./altera/vhdl/220model
220model_ver	Library	\$MODEL_TECH/./altera/verilog/220m...
altera	Library	\$MODEL_TECH/./altera/vhdl/altera

15.

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help

sim - Default

Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage
sim_sharp	sim_sharp/...	Architecture	DU Instance	+acc=...	

16. right mouse button

ColumnLayout AllColumns

Project - C:/Users/aschwa3m/ownCloud/Projects/sim\_sharp/sim\_sharp

Name	Status	Type	Or	Modified
sim_sharp.vhd		VHDL	0	11/09/2020 10:31:38 ...
sharp.vhd		VHDL	1	01/20/2020 02:15:48 ...
sharp_arith.vhd		VHDL	2	01/20/2020 02:15:48 ...
sharp_control.vhd		VHDL	3	01/20/2020 02:15:48 ...
sharp_linemem.vh		VHDL	4	01/20/2020 02:15:48 ...

13.

sim\_sharp

- View Declaration
- View Instantiation
- UVM
- UPF
- Add Wave Ctrl+W 17.
- Add Wave To
- Add Dataflow Ctrl+D
- Add to
- Copy Ctrl+C
- Find... Ctrl+F
- Save Selected...



# Simulation in Questa/Modelsim III

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help

100 ps

Layout Simulate ColumnLayout AllColumns

sim - Default

Instance	Design unit	Design unit type	Top Category
sim_sharp	sim_sharp(...)	Architecture	DU Instance
duv	sharp(behav...)	Architecture	DU Instance
stimuli_process	sim_sharp(...)	Process	-
response_proc...	sim_sharp(...)	Process	-
line_53	sim_sharp(...)	Process	-
standard	standard	Package	Package
textio	textio	Package	Package
std_logic_1164	std_logic_1...	Package	Package
numeric_std	numeric_std	Package	Package
std_logic_textio	std_logic_t...	Package	Package

Wave - Default

Signal	Value
/sim_sharp/clk	0
/sim_sharp/reset_n	U
/sim_sharp/enable_in	UUU
/sim_sharp/vs_in	U
/sim_sharp/hs_in	U
/sim_sharp/de_in	U
/sim_sharp/fr_in	UUUUUUUU
/sim_sharp/g_in	UUUUUUUU
/sim_sharp/b_in	UUUUUUUU
/sim_sharp/vs_out	U
/sim_sharp/hs_out	U
/sim_sharp/de_out	U
/sim_sharp/fr_out	UUUUUUUU
/sim_sharp/g_out	UUUUUUUU
/sim_sharp/b_out	UUUUUUUU
/sim_sharp/clk_o	U
/sim_sharp/led	UUU
/sim_sharp/x_size	-2147483648
/sim_sharp/y_size	-2147483648
/sim_sharp/end_tb	0
/sim_sharp/mismatch	0

Now 0 ps

Cursor 1 0 ps

Project Memory List sim

Transcript

```
# Loading work.sharp_arith(behavior)
# Loading work.sharp_control(behavior)
add wave -position insertpoint sim:/sim_sharp/*

VSIEM 5>
```

Project : sim\_sharp Now: 0 ps Delta: 0 /sim\_sharp/clk



# Simulation in Questa/Modelsim IV

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

100 ps

Layout Simulate ColumnLayout AllColumns

sim - Default

Instance	Design unit	Design unit type	Top Category
sim_sharp	sim_sharp(...)	Architecture	DU Instance
duv	sharp(beha...)	Architecture	DU Instance
stimuli_process	sim_sharp(...)	Process	-
response_proc...	sim_sharp(...)	Process	-
line_53	sim_sharp(...)	Process	-
standard	standard	Package	Package
textio	textio	Package	Package
std_logic_1164	std_logic_1...	Package	Package
numeric_std	numeric_std	Package	Package
std_logic_textio	std_logic_t...	Package	Package

Wave - Default

Signal	Value
/sim_sharp/clk	1
/sim_sharp/reset_n	1
/sim_sharp/enable_in	111
/sim_sharp/vs_in	0
/sim_sharp/hs_in	0
/sim_sharp/de_in	1
/sim_sharp/fr_in	00000101
/sim_sharp/g_in	00000000
/sim_sharp/b_in	00010111
/sim_sharp/vs_out	0
/sim_sharp/hs_out	0
/sim_sharp/de_out	1
/sim_sharp/fr_out	00000000
/sim_sharp/g_out	00000000
/sim_sharp/b_out	00000001
/sim_sharp/clk_o	1
/sim_sharp/led	000
/sim_sharp/x_size	1280
/sim_sharp/y_size	720
/sim_sharp/end_tb	0
/sim_sharp/mismatch	0

Transcript

```
# ** Failure: Simulation completed
# Time: 9946120 ns Iteration: 0 Process: /sim_sharp/stimuli_process File: C:/Users/aschwa3m/ownCloud/Projects/sim_sharp/sim_sharp.vhd
# Break in Process stimuli_process at C:/Users/aschwa3m/ownCloud/Projects/sim_sharp/sim_sharp.vhd line 164
```

VSIM 5>

0 ps to 55168896 ps Project: sim\_sharp Now: 9,946,120 ns Delta: 0 /sim\_sharp/b\_out

# Simulate with self checking testbench

- create project in Modelsim, compile and simulate it (see slides 35ff)
- watch messages:

```
VSIM 3> run -all
# ** Failure: simulation completed, EVERYTHING OK
#   Time: 9946120 ns  Iteration: 0  Process: /sim_sharp/stimuli_process File: C:/Users/aschwa3m/ownCloud/Projects/sim_sct_sharp/sim_sharp_self-checking.vhd
# Break in Process stimuli_process at C:/Users/aschwa3m/ownCloud/Projects/sim_sct_sharp/sim_sharp_self-checking.vhd line 167

VSIM 4>
```



(C:/O\_fpga\_...tap\_MSE\_CX/edupow.qpf)  
(C:/O\_fpga\_.../edupow\_cx/edupow.qpf)  
ds\_7segment...hes\_leds\_7segment.qpf)  
(C:/O\_fpga\_pr...ntel/zaehler/zaehler.qpf)



# SETTING UP A PROJECT IN QUARTUS AND TEST THE DESIGN WITH THE FPGA VISION REMOTE LAB

[18G Transceivers](#)

[Hyperflex Architecture](#)

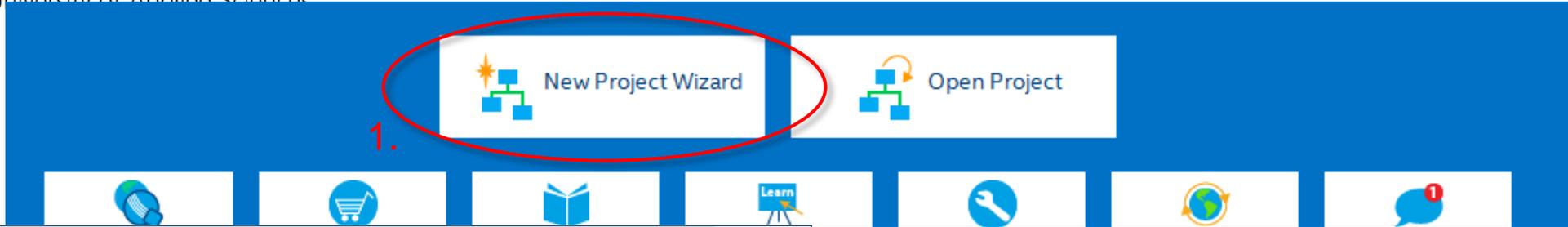
[PCIeGen2 DMA to DDR4](#)

after project load

this screen again



# Starting with the Project Wizard I



New Project Wizard

### Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/Users/aschwa3m/ownCloud/Projects/sharp\_impl

What is the name of this project?

sharp

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.

sharp

Use Existing Project Settings...

2.

3.

< Back Next > Finish Cancel

Fill in the correct name of the top-level entity!

New Project Wizard

### Project Type

4. Select the type of project to create.

Empty project

Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.

Project template

Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).

5.

< Back Next > Finish Cancel Help



# Starting with the Project Wizzard II

**Add Files**

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.  
Note: you can always add design files to the project later.

File name:

File Name | Type | Library | Design Entry/Synthesis Tool | HDL Version

Select File

Organisieren | Neuer Ordner

Name	Änderungsdatum	Typ
db	09.11.2020 13:24	Dateiordner
incremental_db	09.11.2020 12:13	Dateiordner
output_files	09.11.2020 12:20	Dateiordner
<input checked="" type="checkbox"/> sharp.vhd	20.01.2020 02:15	Festplatten-Im
<input checked="" type="checkbox"/> sharp_arith.vhd	20.01.2020 02:15	Festplatten-Im
<input checked="" type="checkbox"/> sharp_control.vhd	20.01.2020 02:15	Festplatten-Im
<input checked="" type="checkbox"/> sharp_linemem.vhd	20.01.2020 02:15	Festplatten-Im
<input checked="" type="checkbox"/> sharp_slice.vhd	20.01.2020 02:15	Festplatten-Im

Dateiname: "sharp\_slice.vhd" "sharp.vhd" "sh" Design Files (\*.tdf \*.vhd \*.vho)

**Family, Device & Board Settings**

Device | Board

Select the family and device you want to target for compilation.  
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family  
Family: **Cyclone V (E/GX/GT/SX/SE/ST)** 10.  
Device: All

Target device  
 Auto device selected by the Fitter  
 Specific device selected in 'Available devices' list  
 Other: n/a

Show in 'Available devices' list

Package: Any  
Pin count: Any  
Core speed grade: Any  
Name filter: **5CEBA2F17C6** 11.  
 Show advanced devices

Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS	PCIe Hard IP BL
5CEBA2F17C6	1.1V	9430	128	128	0	0	0

12.



# Import Assignments and Adding Constraints

Quartus Prime Lite Edition - C:/Users/aschwa3m/ownCloud/Projects/sharp\_impl/sharp - sha

File Edit View Project **Assignments** Processing Tools Window Help

Device... 13.

Settings... Ctrl+Shift+E

Assignment Editor Ctrl+Shift+A

Pin Planner Ctrl+Shift+N

Remove Assignments...

Back-Annotate Assignments...

**Import Assignments...** 14.

Export Assignments...

Assignment Groups...

Logic Lock Regions Window Alt+L

Design Partitions Window Alt+D

Entity: Instar

Cyclone V: 5CEBA2F17C6

right-click  
-> Settings

Without the assignments the software doesn't know the location of the pins and you won't have video signals!

Settings - sharp

Category:

- General
- Files
- Libraries
- IP Settings
  - IP Catalog Search Locations
  - Design Templates
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Incremental Compilation
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Board-Level
- Compiler Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- Timing Analyzer** 18.
- Assembler
- Design Assistant
- Signal Tap Logic Analyzer
- Logic Analyzer Interface
- Power Analyzer Settings
- SSN Analyzer

Device/Board...

**Timing Analyzer**

Specify Timing Analyzer options.

SDC files to include in the project

File name:  19.

Select File

Organisieren Neuer Ordner

Name	Änderungsdatum	Typ
db	09.11.2020 13:24	Dateiordner
incremental_db	09.11.2020 12:13	Dateiordner
output files	09.11.2020 12:20	Dateiordner
<input checked="" type="checkbox"/> sharp.sdc	20.01.2020 02:15	SDC-Datei

20.

Dateiname: sharp.sdc

Synopsys Design Constraints Fi

21.

Description:  
Associates a Synopsys Design Constraint File (.sdc) with this project.

22.

17.

15.

16. sharp.sdc sharp\_default\_Cyclone\_V.qsf

17. sharp\_default\_Cyclone\_V\



23.

Entity:Instance

```
1  -- sharp.vhd
2  -- top level
3  -- FPGA Vision Remote Lab http://h-brs.de/fpga
4  -- (C) Marco Winzker, Hochschule Bonn-Rhein-Sieg
5
6
7
8  library IEEE;
9  use IEEE.STD_LOGIC_1164.ALL;
10 use IEEE.NUMERIC_STD.ALL;
11
12 entity sharp is
13 port (clk      : in  std_logic;
14       reset_n  : in  std_logic;
15       enable_in : in  std_logic_vector(2 downto 0);
16       video_in  : in  std_logic;
17       ...
```

Entity:Instance

- Cyclone V: 5CEBA2F17C6
  - sharp
    - sharp\_slice:b\_slice
    - sharp\_control:control
    - sharp\_slice:g\_slice
    - sharp\_slicer\_slice

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Flow Messages
- Flow Suppressed Messages
- Assembler
- Timing Analyzer

Flow Summary

Flow Status	Successful - Mon Nov 09 12:20:47 2020
Quartus Prime Version	19.1.0 Build 670 09/22/2019 SJ Lite Edition
Revision Name	sharp
Top-level Entity Name	sharp
Family	Cyclone V
Device	5CEBA2F17C6
Timing Models	Final
Logic utilization (in ALMs)	345 / 9,430 ( 4 %)
Total registers	326
Total pins	63 / 128 ( 49 %)
Total virtual pins	0
Total block memory bits	294,912 / 1,802,240 ( 16 %)
Total DSP Blocks	0 / 25 ( 0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 4 ( 0 %)
Total DLLs	0 / 4 ( 0 %)

Tasks

- Compile Design ✓
- Analysis & Synthesis ✓
- Fitter (Place & Route) ✓
- Assembler (Generate programming files) ✓
- Timing Analysis ✓
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)





fpga-vision-lab.h-brs.de/weblab/login



Hochschule  
Bonn-Rhein-Sieg  
University of Applied Sciences



Einloggen

Benutzername

drea\_test

Passwort

.....

Einloggen »

Einige Experimente erlauben einen Gast-Zugang

Log in as guest

Welcome to the

## FPGA Vision Remote Lab

The FPGA Vision Remote Lab, based on WebLab-Deusto, is a remote laboratory. Students access experiments physically located in the Hochschule Bonn-Rhein-Sieg, having the same experience as if in traditional hands-on lab sessions.



### Support

For any technical issue you may find, please contact us at [fpga-vision-lab@h-brs.de](mailto:fpga-vision-lab@h-brs.de)



### Demo

If you do not have a user account, you can try our demo experiments with the username **demo** and the password **welcome**.



### Get an Account

If you want to have full access to the provided experiments you can register [here!](#)

For information on [privacy](#), please see the [FAQ!](#)



### Open-Source

WebLab-Deusto is Open Source Software, and it is available in <https://github.com/weblabdeusto/weblabdeusto/>



# Choose and Reserve your Experiment

Hochschule Bonn-Rhein-Sieg University of Applied Sciences

**3.**

CV 1 CV 2 CV all

**4.**

Benötigte Zeit: 5:00 **Reservieren**

**C V all** (FPGA experiments)

Willkommen zum Experiment!!!

Before reserving the experiment please make sure you have the bit-file(s) ready.

The Quartus software can be downloaded from [Intel FPGA](#).

For additional information please visit our [FAQ](#) or our [project website!](#)

Stats  
Number of uses: 515  
Status: Online

Details

FPGA



5.

Binary for FPGA:

Datei auswählen Keine ausgewählt

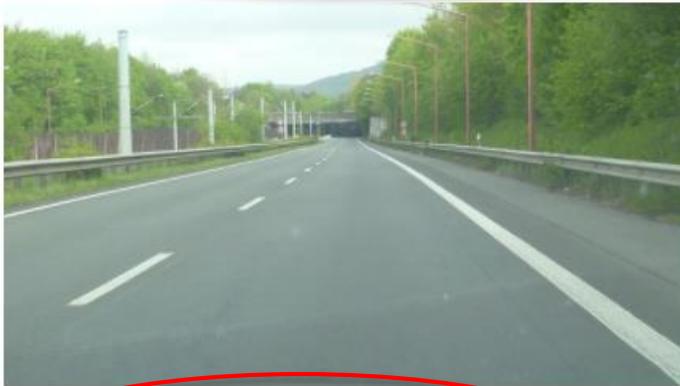
6.

Upload FPGA binary and start experiment

FPGA Core Current:

\_\_\_\_.\_\_\_\_ mA  
(Core Supply Voltage is 1.1 V)

<< < Select Input Image > >>



You are working on FPGA:CV1

Please perform an experiment to get an output image



Datei auswählen Keine ausgewählt

Upload your own input image now

enable\_in(0)  OFF

enable\_in(1)  OFF

enable\_in(2)  OFF

00:03:05

Program Default Design

Update Output Image and Core Current

Update Core Current Only



Global Online Laboratory Consortium



**AND NOW?**

**ADJUST THE FILTER, TRY OUT OTHER  
TEMPLATES, ...**



**FPGA Vision**