

# XILINX ISE WEBPACK und DIGILENT BASYS2

Eine kurze Einführung. [Download ISE Projektdateien](#) auch links im Menü (ZIP)

Abteilung Elektronik an der HTL Mödling 2011/2015

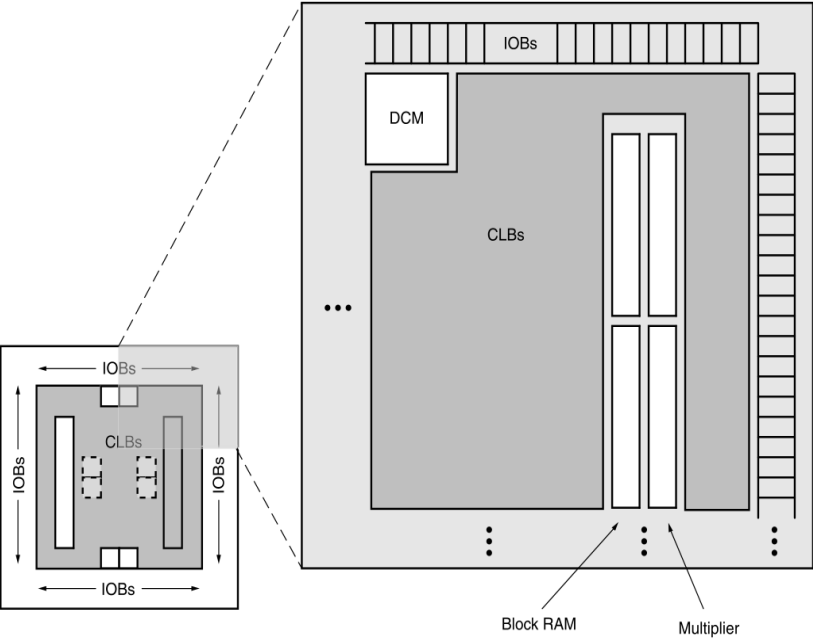
## Grundbegriffe

- [Xilinx](#) ist eine Firma die programmierbare Digitalbausteine herstellt. Die Entwicklungsplattform heißt „ISE Webpack“ und ist frei verfügbar.
- [FPGA](#) (field programmable gate array): RAM-Technologie d.h. Bei Stromausfall geht die Schaltung verloren; muß in einem Bootprozess von externem Flash (USB Stick Technologie) geladen werden.
- [CPLD](#) (complex programmable device): Flash-Technologie; behält die Schaltung bei Stromausfall; Flash hat große Speicherzellen, CPLD's enthalten daher nur einen kleineren Speicher als FPGAs.
- **1 GE** (gate equivalent, Gatteräquivalent): Maßeinheit für die Komplexität einer Digitalschaltung z.B: 2-fach-Nand = 1GE, 2-fach-NOR = 1GE, Inverter = 0.5GE, RS-FF = 2GE, D-FF = 8 GE, Garagentorsteuerung: 100GE, Mikroprozessor: 1E4 bis 1E7 GE, CPLD: 1E4 GE, FPGA 1E6 GE
- **Software/Firmware:** Firmware legt die grundlegende Verschaltung eines Systems fest; liegt meist in einem Flash; es geht um die Belange der Hardware. Software liegt meist in einem Schreib-/Lesespeicher und kümmert sich um die Bedürfnisse der Anwender.
- **VHDL** Programmiersprache zur Entwicklung von Digitalschaltungen (Firmware)

## ***Baustein Spartan XC3E100S CP132 -4***

1. FPGA
2. Package CP132 (132 Pin Ball Grid Array)
3. Standard Speed Grade -4 (Laufzeit von Eingang auf Ausgang ca. 10ns)

# Innenleben



# Features

Table 1: Summary of Spartan-3E FPGA Attributes

| Device    | System Gates | Equivalent Logic Cells | CLB Array<br>(One CLB = Four Slices) |         |            |              | Distributed RAM bits <sup>(1)</sup> | Block RAM bits <sup>(1)</sup> | Dedicated Multipliers | DCMs | Maximum User I/O | Maximum Differential I/O Pairs |
|-----------|--------------|------------------------|--------------------------------------|---------|------------|--------------|-------------------------------------|-------------------------------|-----------------------|------|------------------|--------------------------------|
|           |              |                        | Rows                                 | Columns | Total CLBs | Total Slices |                                     |                               |                       |      |                  |                                |
| XC3S100E  | 100K         | 2,160                  | 22                                   | 16      | 240        | 960          | 15K                                 | 72K                           | 4                     | 2    | 108              | 40                             |
| XC3S250E  | 250K         | 5,508                  | 34                                   | 26      | 612        | 2,448        | 38K                                 | 216K                          | 12                    | 4    | 172              | 68                             |
| XC3S500E  | 500K         | 10,476                 | 46                                   | 34      | 1,164      | 4,656        | 73K                                 | 360K                          | 20                    | 4    | 232              | 92                             |
| XC3S1200E | 1200K        | 19,512                 | 60                                   | 46      | 2,168      | 8,672        | 136K                                | 504K                          | 28                    | 8    | 304              | 124                            |
| XC3S1600E | 1600K        | 33,192                 | 76                                   | 58      | 3,688      | 14,752       | 231K                                | 648K                          | 36                    | 8    | 376              | 156                            |

CLB = Konfigurierbarer Logikblock (LUT Wahrheitstafel + Flipflops)

Block Ram oder verteiltes Ram

18x18bit Multiplizierer

DCM Digitale Clock Manager (erzeugt aus langsamen Takten schnelle usw.)

Ausgänge auch differenziell (für höhere Störsicherheit)

## IO

1. Slew Rate einstellbar; Pullups zuschaltbar u.v.a.m
2. Kompatibel zu
  - 3.3V low-voltage TTL (LVTTL)
  - Low-voltage CMOS (LVCMOS) at 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
  - aber auch zu differenziellen Standards
    - LVDS low voltage differential signalling

# Projekt anlegen

New Project Wizard

### Project Settings

Specify device and project properties.  
Select the device and design flow for the project

| Property Name                          | Value                         |
|--|-------------------------------|
| Evaluation Development Board           | None Specified                |
| Product Category                       | All                           |
| Family                                 | Spartan3E                     |
| Device                                 | XC3S100E                      |
| Package                                | CP132                         |
| Speed                                  | -5                            |
| Top-Level Source Type                  | Schematic                     |
| Synthesis Tool                         | XST (VHDL/Verilog)            |
| Simulator                              | ISim (VHDL/Verilog)           |
| Preferred Language                     | Verilog                       |
| Property Specification in Project File | Store non-default values only |
| Manual Compile Order                   | <input type="checkbox"/>      |
| VHDL Source Analysis Standard          | VHDL-93                       |
| Enable Message Filtering               | <input type="checkbox"/>      |

More Info      Next      Cancel

- Projektordner anlegen z.B: M:\hwe\xilinx
- Beispiel kopieren und entpacken I:\kneringer\xilinx\uebung1.zip → m:\xilinx\uebung1
- Software starten: „Xilinx 64bit“

### ***Projekt öffnen und Einstellungen kontrollieren***

- //Project/Design Properties
  - Family: Spartan3E
  - Device: XC3S100E
  - Package: CP132
  - Simulator: ISIM
  - Preferred Language: VHDL



## **Schaltplan zeichnen**

**keine Sonderzeichen nirgendwo!** (auch nicht in Kommentaren; english spoken)  
Verschieben von Schaltungsteilen mit Attributen (z.B: Signalnamen) ist nicht möglich, daher ist vor dem Platzieren ein Floorplaning nötig oder Attribute erst zuletzt am Schaltplan sichtbar machen.

### **Location-Attribut vergeben**

legt fest, bei welchen Pins die Ein- und Ausgänge angeschlossen sind

1. Location-Attribut über User Constraints File (.UCF)
  - menu://Project/New File/Implementation Constraints File

Vorlage kopieren von : <http://kner.at/home/60.Elektronik/xilinx/tb.template1.zip>

### Beispiel:

```
#loc=p38 ... Achtung, die Pin-Namen ändern sich je nach Package  
des ASICS  
#bufg=clk ... führt das Signal clk über eine Lonngline (=schnelle  
Leiterbahn zur Signalverteilung)  
#data_gate Eigenschaft führt das Signal über ein Latch, dadurch  
wird Strom gespart, weil nicht jede Signaländerung am Eingang in-  
terne Logik umschaltet  
  
net clk loc = p38 | bufg=clk | data_gate;
```

### **Häufige Fehler:**

Takt für Flipflops über normale Leiterbahn zur Verfügung stellen ist ein Fehler; Clock muss über schnelle Leiterbahnen verteilt werden, dafür gibt es eigene Clock-Eingänge am ASIC

Abhilfe: NET "mclk" CLOCK\_DEDICATED\_ROUTE = FALSE; in das UCF File schreiben erlaubt die Verwendung von normalen Pins

## Schaltplan prüfen

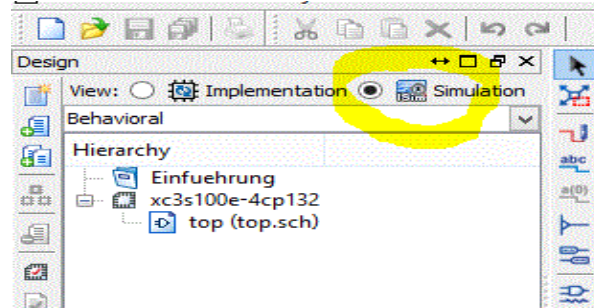
1. menu://Tools/Check Schematic

## Simulation

siehe [ISIM](#)

### Simulation starten

1. Projektansicht auf „Simulation
2. auf das zu simulierende Modul (top.sch)
3. [Simulate Behavioral Model] Simulator.



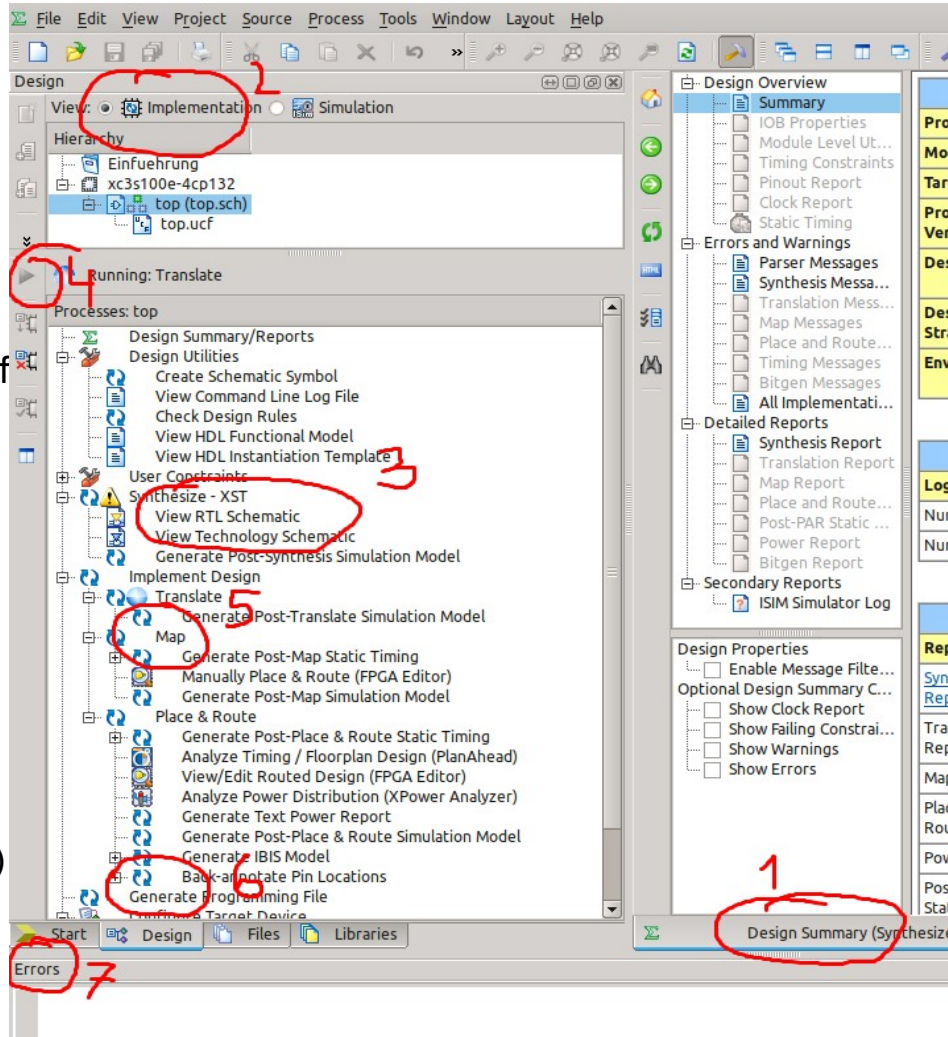
umschalten  
klicken  
startet den

### Bedienung Simulator

- `put sw(0) 1`
- `put sw 15 -radix dec`
- `put sw 0x3f -radix hex`
- `isim force add sw(0) 0 -value 1 -time 50 ns -repeat 80 ns` oder einfacher über Rechtsklick auf das Signal //Force Clock
- Zoom: F6 (Zoom all) und „Zoom to Cursors“ (Lupe mit rotem Rechteck)

## Implementierung

- Umschalten auf (2)
- Starten mit (4)
- Programmierdatei erzeugen (6); bei den Eigenschaften evtl. auf Takt JTAG umstellen
- wenn Schaltungsteile nicht verwendet werden, in den Optionen von Map (5) auf map -u stellen
- Kontrolle (7) und (1)
- Ergebnis der Implementierung in (3)



## Erzeugen der Firmware

4. Generate Programming File

5. submenu://Process Properties/Startup Options/JTAG Clock

Category Properties are not editable while a process is running.

| Switch Name    | Property Name                  | Value                    |
|----------------|--------------------------------|--------------------------|
| -g StartUpClk: | FPGA Start-Up Clock            | JTAG Clock               |
| -g DonePipe:   | Enable Internal Done Pipe      | <input type="checkbox"/> |
| -g DONE_cycle: | Done (Output Events)           | Default (4)              |
| -g GTS_cycle:  | Enable Outputs (Output Events) | Default (5)              |

## ***Programmierung über ADEPT2 Software und JTAG Interface***

6. Anschließen an USB Kabel

7. Checken, ob Demoboard auf JTAG eingestellt ist (Jumper JP6 in Stellung „PC“)

## ***Ein VHDL Modul in den Schaltplan einbauen***

1. VHDL Modul schreiben und debuggen
  1. z.B: menu://Language Templates/VHDL/Synthesis Constructs/Coding Examples/Misc/7-Segment Decoder
  2. processes://synthesize-XST/Check Syntax
2. processes://DesignUtilities/Create Schematic Symbol erzeugt ein Symbol in der Library „WORK“
3. das Symbol einbauen und anschlieÙe



## ***Links***

- Kurze Dokumentation über das Demoboard  
<http://kner.at/home/60.Elektronik/xilinx/index.html>
- Materialien  
<http://kner.at/home/60.Elektronik/xilinx/materialien/index.html>