



# Realisierung eines fernsteuerbaren Testcontrollers für FPGA-basierte Systeme

## Kolloquium zum Mastermodul INF-PM-FPG

Patrick Lehmann

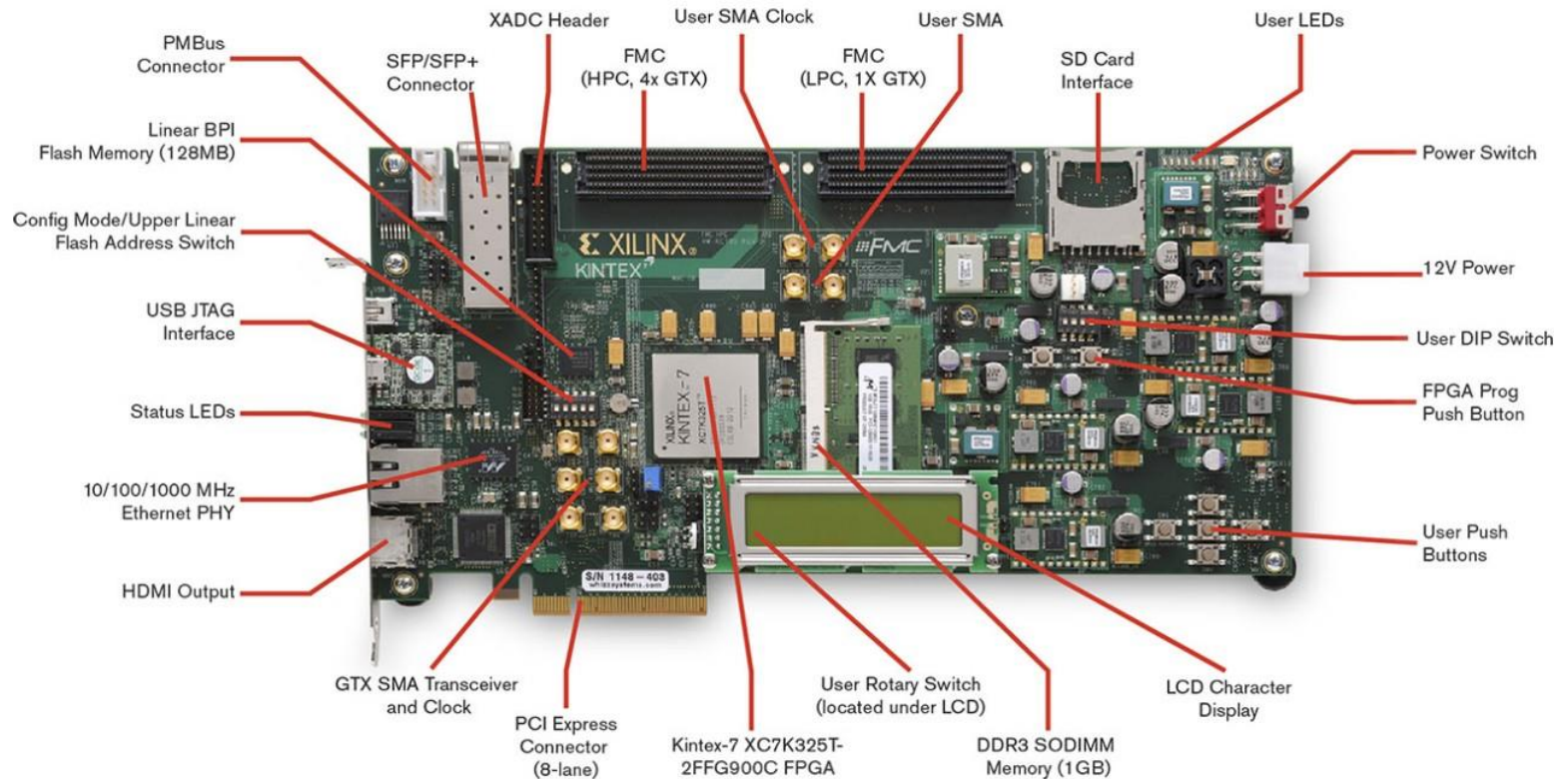
Patrick.Lehmann@tu-dresden.de

Dresden, 07.05.2015



# Motivation

## Prototyping-Board mit Schnittstellenübersicht



[1] Xilinx KC705 Produktbild, 06.05.2015 - Xilinx Kintex-7 FPGA KC705 Evaluation Board

# Agenda

- 1 Zielstellung
- 2 Analyse
- 3 PicoBlaze Softcore und SoFPGA
- 4 Ergebnisse

# 1 Zielstellung

## 1.1 Zielstellung

### Hardware:

- Implementierung eines Testcontrollers
- Integration einer Kommunikationsschnittstelle
- Entwurfsunterstützung für den Entwickler

### Software:

- Nachweis der Fernsteuerbarkeit des Testcontrollers
  - Fernsteuerung vom Messarbeitsplatz
  - Fernsteuerung von einem Remote-Arbeitsplatz
- Bereitstellung von wiederverwendbaren Bibliotheken

## 2 Analyse

### **2.1 Interaktionsmöglichkeiten mit dem FPGA**

#### Dateneingabe:

- Einzelbit Eingabe
- Einzelbyte Eingabe
- ROM unterstützte Eingaben
- Eingaben in Echtzeit
- Erzeugung von Testdaten in Echtzeit

## 2 Analyse

### **2.1 Interaktionsmöglichkeiten mit dem FPGA**

#### Datenausgabe:

- Ausgabe von Einzelbits und -bytes
- Endergebnis
- Messwertlisten
- Weiterverarbeitet der Messwerte

## 2 Analyse

### **2.1 Interaktionsmöglichkeiten mit dem FPGA**

#### Fehlerausgabe:

- Fehlerindikation
- Fehlermeldung
- Fehlerprotokoll
- Abbild des Systemzustandes
- Tracelog

## 2 Analyse

### **2.2 Interaktionsgrad und Beeinflussbarkeit**

Interaktionsgrad:

- Manuell
- Interaktiv
- Vollautomatisiert

Beeinflussbarkeit:

- Parametrierbar
- (Fern-)steuerbar
- Rekonfigurierbar



## 2 Analyse

### **2.3 Testfälle**

Merkmale der Testfälle:

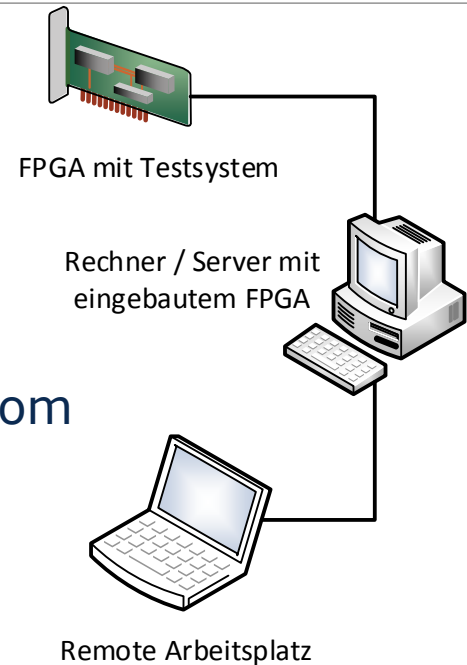
- Begrenzte Anzahl, fest kodiert
- Parametrierbar
- Austauschbar
- Benchmark

## 2 Analyse

### 2.4 Fernsteuerbarkeit

#### Terminal Sitzung:

- Programmieren per iMPACT oder Skript
- SoFPGA Steuerung per Kitty / minicom
- Direkte ChipScope-Nutzung

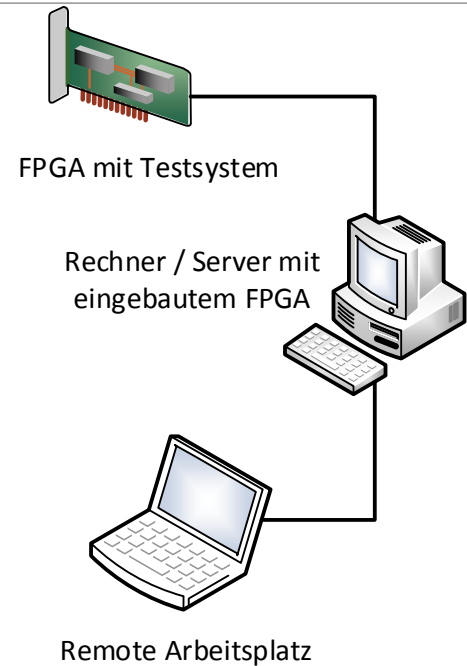


## 2 Analyse

### 2.4 Fernsteuerbarkeit

#### Remote Konsole:

- SSH Sitzung
- Programmieren per Script
- SoFPGA Steuerung per minicom

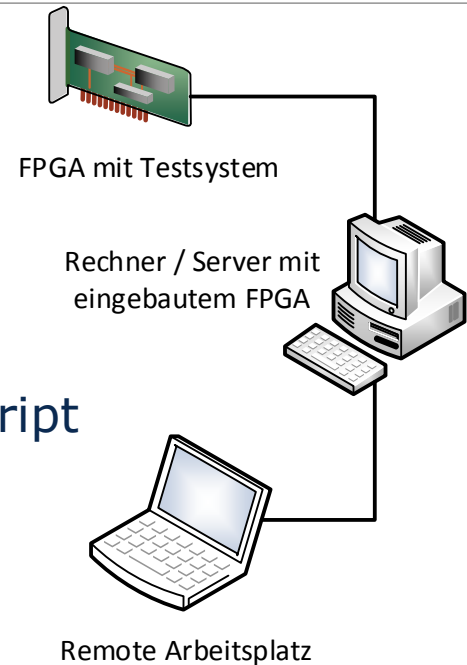


## 2 Analyse

### 2.4 Fernsteuerbarkeit

#### Remote Sitzung:

- Remote Desktop Verbindung, ...
- Programmieren per iMPACT oder Skript
- Terminalsteuerung per Kitty
- Remote ChipScope-Nutzung

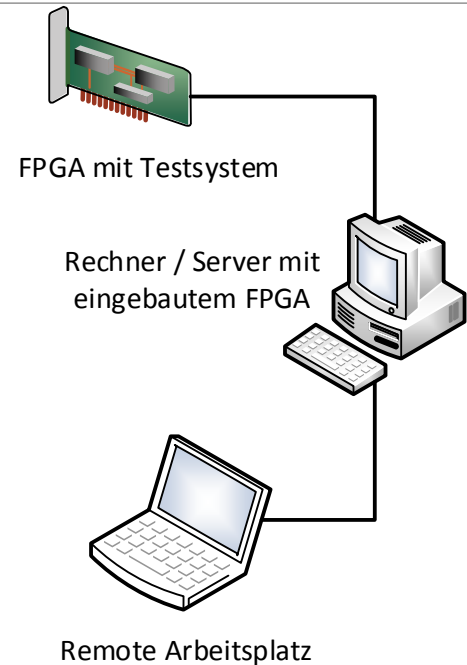


## 2 Analyse

### 2.4 Fernsteuerbarkeit

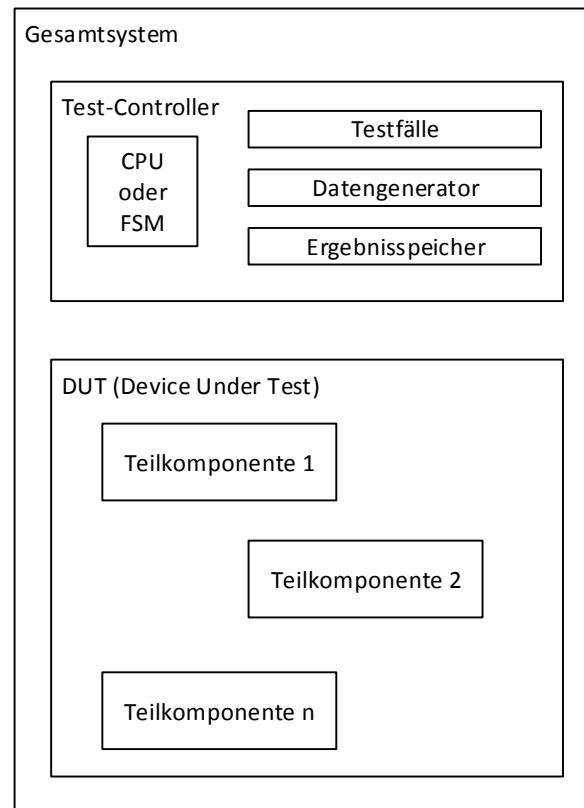
#### Weiterleitung der Schnittstellen:

- Programmieren per klink und rc3e
- UART Forwarding per socat  
`tcp://fpga-cloud:40 001`
- ChipScope Zugriff auf  
`tcp://fpga-cloud:50 001`



## 2 Analyse

### 2.5 Architekturübersicht



## 3 Der PicoBlaze Softcore (KCPSM6)

### 3.1 Architekturmerkmale

Ken Chapman's Programmable State Machine [2]:

- 8-Bit Prozessor / Programmierbarer Zustandsautomat
- 4096 Worte Instruktionsspeicher
- 2 Registerbänke á 16 Register
- 256 Byte Scratchpad Memory
- 256 I/O Adressen
- Bis zu 280 MHz (140 MIPS) auf einem 7-Series FPGA
- Unterstützt Interrupts

[2] PicoBlaze6 - Rev9, 30.09.2014

## 3 Der PicoBlaze Softcore

### 3.2 Erweiterungen

#### Paged Instruction ROM:

- Erweiterung auf bis zu 8 Pages á 4096 Instruktionen  
=> 32 Ki Instruktionen  
=> Erlaubt die Speicherung vieler String-Konstanten
- Zur Laufzeit Reprogrammierbar per JTAG

#### Interrupt Controller:

- Unterstützt bis zu 32 maskierbare Interrupts
- Round-Robin Arbitrierung
- Fehlermeldungsregister pro Interrupt-Port



## 3 Der PicoBlaze Softcore

### **3.3 System-on-FPGA (SoFPGA)**

#### Grundsystem:

- PicoBlaze
- PagedROM
- InterruptController

#### Arithmetische Beschleuniger:

- Multiplizierer (16 Bit)
- Dividierer (32 Bit)
- Scaler (40x8/8 Bit)
- BCD-Counter

## 3 Der PicoBlaze Softcore

### **3.3 System-on-FPGA (SoFPGA)**

#### Kommunikation:

- General Purpose I/O (GPIO)
- Bit-Banging I/O (BBIO)
- LC-Display
- UART (~ 1 MBaud)
- DRP Adapter

#### DUT Adapter:

- SATAController Adapter
- SATASController Adapter
- DMATest Adapter

## 4 Ergebnisse

### 4.1 FPGA Ressourcenbedarf

VHDL Modul	Register	LUTs	BlockRAM
Gesamt (Kintex-7 325T)	407.600	203.800	445
PicoBlaze mit 2 ROM-Pages	79	194	4
SoFPGA mit 15 Devices	1336	1661	6
DUT Adapter	456	226	0
SATAController Stack (DUT)	1220	2137	10

# 4 Ergebnisse

## 4.2 Fernsteuerung per Kitty

```
- KITTY
SATAController
-----
License: Apache License 2.0
Author(s): Patrick Lehmann
          Steffen Koehler
          Thomas Frank
          Martin Zabel
Design: PicoBlazeDMAtest_KC705 - Reference Design
-----
Configuring PCA9548 I2C switch for Si570 access... [DONE]
Configuring Si570:
  Recalling configuration from NVRAM... [DONE]
  Loading configuration into RAM... [DONE]
Settings:
  HS-DIV: 04
  NI: 07
  RFreq: 02BC005588
  Calculating new RFreq for 150 MHz... [DONE]
  Saving configuration to device... [DONE]
FreqM: 000.000,000 MHz
Powering up SATAController...
  wait for ClockNetwork_Reset... [DONE]
  wait for initial Reset... [DONE]
  wait for SATAc.PhysicalLayer... [DONE]
  wait for SATAc.LinkLayer... [DONE]
List of available commands: h, H, r, R, p, P, m, c, 1, 2, 3
                              (press H for explanation)
SATA> |
```

```
- KITTY
SATAController
-----
License: Apache License 2.0
Author(s): Patrick Lehmann
          Steffen Koehler
          Thomas Frank
          Martin Zabel
Design: PicoBlazeDMAtest_KC705 - Reference Design
-----
Configuring PCA9548 I2C switch for Si570 access... [DONE]
Configuring Si570:
  Recalling configuration from NVRAM... [DONE]
  Loading configuration into RAM... [DONE]
Settings:
  HS-DIV: 04
  NI: 07
  RFreq: 02BC005588
  Calculating new RFreq for 150 MHz... [DONE]
  Saving configuration to device... [DONE]
FreqM: 000.000,000 MHz
Powering up SATAController...
  wait for ClockNetwork_Reset... [DONE]
  wait for initial Reset... [DONE]
  wait for SATAc.PhysicalLayer... [DONE]
  wait for SATAc.LinkLayer... [DONE]
List of available commands: h, H, r, R, p, P, m, c, 1, 2, 3
                              (press H for explanation)
SATA> 1
Block 00: Checksum: 0x45C4 Time: 000.04483 s
Block 01: Checksum: 0x094A Time: 000.01119 s
Block 02: Checksum: 0x636C Time: 000.00644 s
Block 03: Checksum: 0x87B5 Time: 000.00524 s
Block 04: Checksum: 0x4562 Time: 000.00404 s
Block 05: Checksum: 0xB9ED Time: 000.00524 s
Block 06: Checksum: 0xC3F5 Time: 000.00405 s
Block 07: Checksum: 0x22E2 Time: 000.00522 s
Block 08: Checksum: 0x3692 Time: 000.00405 s
Block 09: Checksum: 0xB3D9 Time: 000.00525 s
SATA> 2
Block 00: Checksum: 0x1C83 Time: 005.46207 s
Block 01: Checksum: 0xF405 Time: 005.52436 s
Block 02: Checksum: 0xC705 Time: 005.42737 s
Block 03: Checksum: 0x341E Time: 005.51406 s
Block 04: Checksum: 0x6C0C Time: 005.49047 s
Block 05: Checksum: 0x3979 Time: 005.49681 s
Block 06: Checksum: 0x14D6 Time: 005.43462 s
Block 07: Checksum: 0x2F11 Time: 005.52406 s
Block 08: Checksum: 0x13C5 Time: 005.46807 s
Block 09: Checksum: 0x36D2 Time: 005.47883 s
SATA> 3
Block 00: Checksum: 0xDE14 Time: 352.77404 s
SATA> |
```

## 4 Ergebnisse

### **4.3 Wünschenswerte Verbesserungen**

Weitere PicoBlaze Devices:

- Timer
- Echtzeituhr mit Batteriebackup
- UDP/IP Integration

Software:

- Interrupt-Handler-Routinen
- Co-Routinen / Asynchrone Programmierung
- Makrounterstützung (m4 oder cpp) [3]

# Quellen

- [1] <http://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html#baseboard>
  - Xilinx KC705 Evaluation Kit – Produktbild
- [2] [http://www.xilinx.com/ipcenter/processor\\_central/picoblaze/member/](http://www.xilinx.com/ipcenter/processor_central/picoblaze/member/)
  - Xilinx PicoBlaze Sourcefiles (KCPSM6 Rev9 30.09.2014)
  - PicoBlaze for Spartan-6, Virtex-6, 7-Series, Zynq and UltraScale Devices (KCPSM6)
  - Ultra-Compact UART Macros for Spartan-6, Virtex-6 and 7-Series
- [3] <https://code.google.com/p/opbasm/>
  - Open PicoBlaze Assembler von Kevin Thibedeau (v1.2.7)
  - m4 support for opbasm