

# Smart Metering: Entwicklung eines intelligenten Stromzählers

**Boppard, Echtzeit 2010**

**Steffen Mauch  
Dirk Benyoucef**

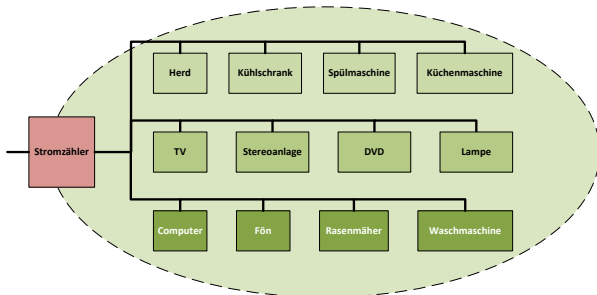
19. Nov. 2010

- 1 Kontext
- 2 Arbeit
- 3 Ergebnisse / Bewertung
- 4 Zusammenfassung

- 1 Kontext
- 2 Arbeit
- 3 Ergebnisse / Bewertung
- 4 Zusammenfassung

## Forschungsvorhabens

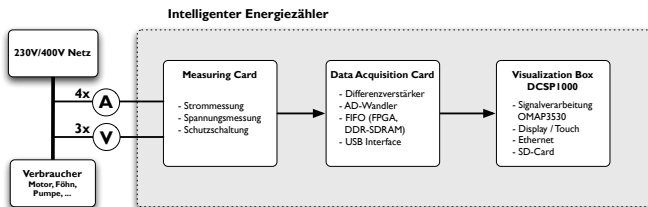
Ziel des Forschungsvorhabens 'Smart Metering: Disaggregation von Endverbraucher' ist die Identifikation der Verbraucher durch Beobachtung des Energieverbrauchs am Energiezähler.



## Messsystem

Das System besteht aus den Komponenten

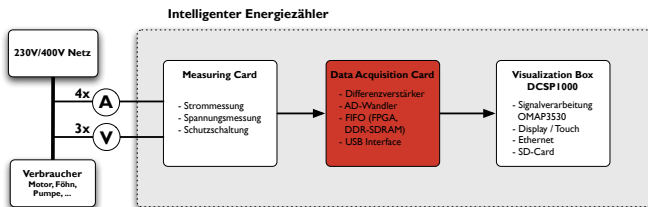
- ▶ Measuring Card,
- ▶ Data Acquisition Card und
- ▶ Visualization Box.

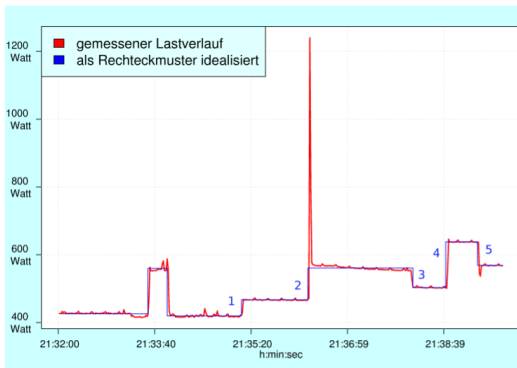
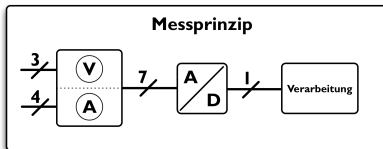


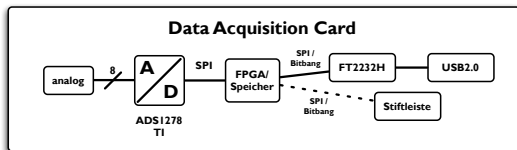
## Messsystem

Das System besteht aus den Komponenten

- ▶ Measuring Card,
- ▶ Data Acquisition Card und
- ▶ Visualization Box.







## Technische Daten

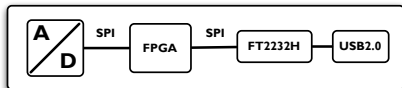
- ▶ 8 parallele differentielle Eingangskanäle
- ▶ 24 Bit A/D-Wandler (ADS1278 von TI)
- ▶ max. Abtastfrequenz 125 kHz
- ▶ Spartan 3E FPGA XC3S500E-4FG320C
- ▶ 64 MByte DDR-SDRAM
- ▶ USB-Schnittstelle mittels FT2232H



## Ziele

Sicherer Datentransfer von DAC →  
PC möglichst in Echtzeit über USB  
oder Ethernet

- ▶ Datenvolumen bis 24 MBit/s
- ▶ kein Verlust von Daten
- ▶ Konfiguration des AD-Wandlers per USB
- ▶ optional Ethernet

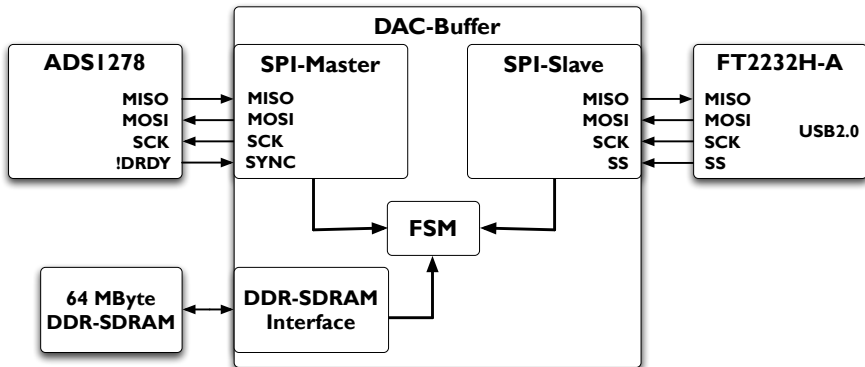


## Problem

USB ist nur eingeschränkt echtzeitfähig!

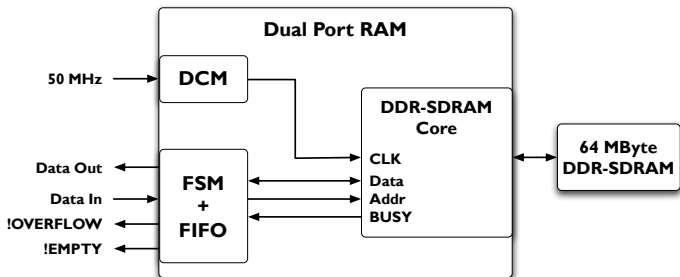
- 1 Kontext
- 2 Arbeit
  - DDR-Core Konzept
  - MicroBlaze Konzept
- 3 Ergebnisse / Bewertung
- 4 Zusammenfassung

1. Konzept DDR-Core - SPI
2. Konzept DDR-Core - FIFO sync
3. Konzept MicroBlaze



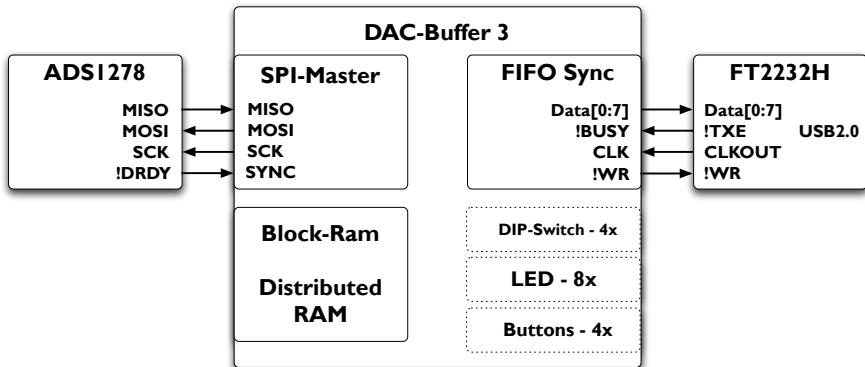
## Problem

Der Speichercontroller in VHDL ist nicht Dual Port fähig.



## Lösung

Takt wird durch die DCM des FPGA verdoppelt.  
SPI-Master und -Slave können nun 'gleichzeitig' zugreifen.



## Vorteile

- ▶ Einfache Realisierung
- ▶ Geringe Anforderungen an den FPGA

## Nachteile

- ▶ aufwendig bei Erweiterungen
- ▶ keine einfache Integration von Ethernet möglich

## Merkmale

- ▶ 32-bit RISC Microcontroller für FPGAs von Xilinx
- ▶ keine physischer  $\mu$ C sondern ein Softcore
- ▶ 3- bis 5-stufige Pipeline, interner Cache, PLB und OPB Bus, ...
- ▶ umfangreiche IP Module
- ▶ Betriebssysteme:  $\mu$ Clinux, Linux oder FreeRTOS

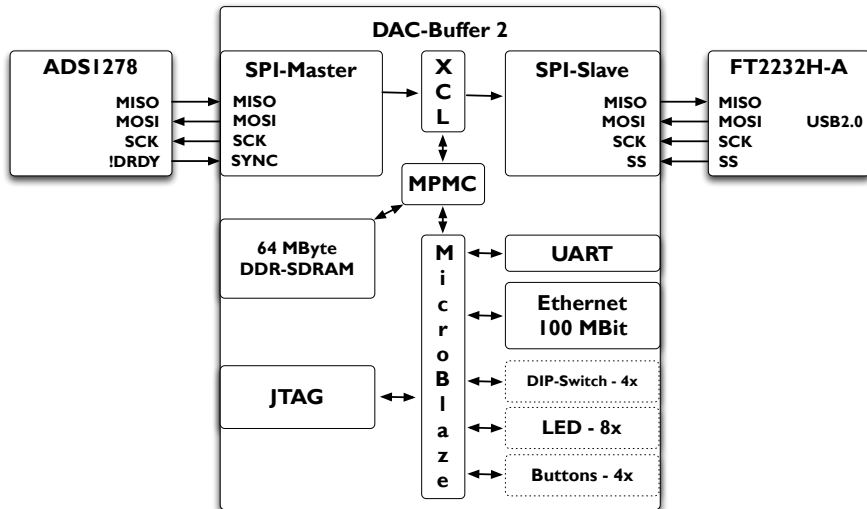


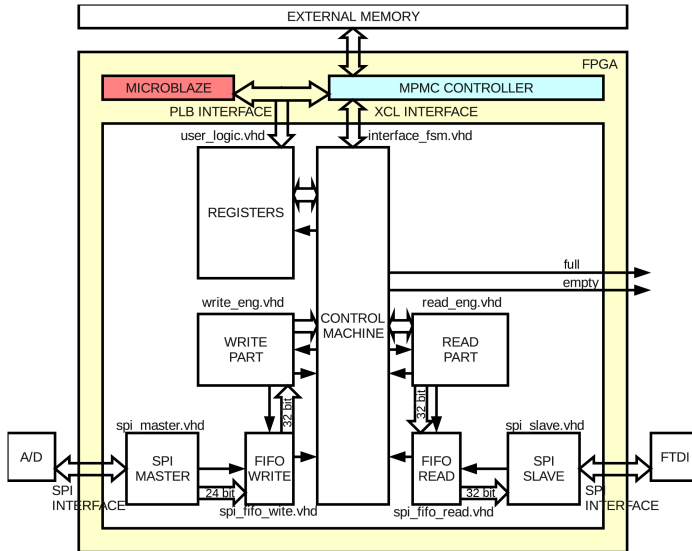
## Vorteile

- ▶ MPMC Anbindung durch PLB Bus
- ▶ OPB Bus für Peripherie
- ▶ z.B. Integration von Ethernet
- ▶ sehr leistungsstark und relativ leicht modifizierbar

## Nachteil

- ▶ größerer Overhead für Projektstart
- ▶ viele vordefinierte Schnittstellen (Spezifikationen)
- ▶ komplexe Struktur
- ▶ aufwendige Fehlersuche



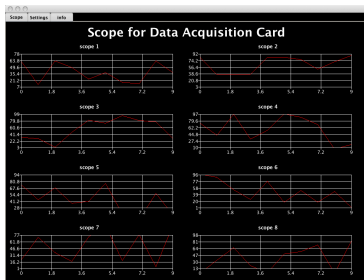




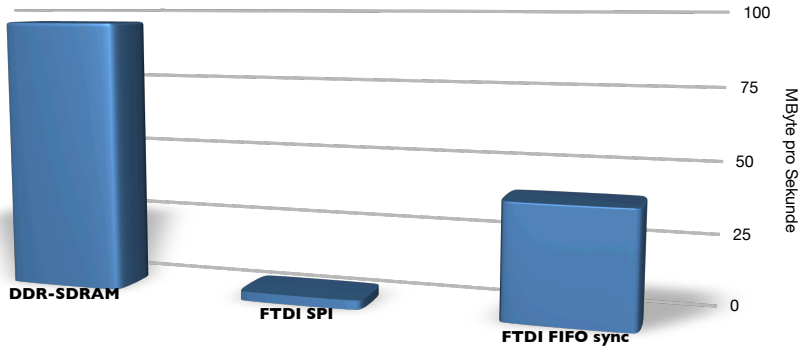
- 1 Kontext
- 2 Arbeit
- 3 Ergebnisse / Bewertung
  - Benchmark
- 4 Zusammenfassung

## Software

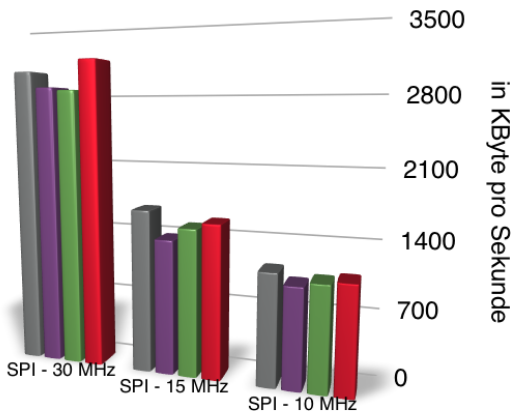
- ▶ libFTDI mit asynchroner Programmierung (SPI, FIFO-Modus)
- ▶ Qt4-Core für die Threads und Dateiverwaltung
- ▶ MySQL Datenbankbindung
- ▶ Scope für graphische Visualisierung



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <INFO>
3   <card>
4     <version>rev_1</version>
5     <software></software>
6     <build-date>Wednesday, 17. November 2010, 22:32:50 Uhr</build-date>
7   </card>
8   <dates>
9     <days>2010-11-17</days>
10    <time>22:32:51</time>
11  </dates>
12 </INFO>
13 <DATA>
14   <block>
15     <ch1>0.00000000</ch1>
16     <ch2>-3.30000000</ch2>
17     <ch3>3.30000000</ch3>
18     <ch4>0.00000000</ch4>
19     <ch5>-2.50000000</ch5>
20     <ch6>2.50000000</ch6>
21     <ch7>0.00000000</ch7>
22     <ch8>0.00000000</ch8>
23   </block>
24   <block>
25     <ch1>0.84147098</ch1>
26     <ch2>-3.30000000</ch2>
27     <ch3>3.30000000</ch3>
28     <ch4>0.00000000</ch4>
29     <ch5>-2.50000000</ch5>
30     <ch6>2.50000000</ch6>
31     <ch7>0.00000000</ch7>
32     <ch8>0.00000000</ch8>
```



- Mac OS X 10.5
- Linux Ubuntu 9.04
- Beagleboard
- Linux Debian 4.0







- 1 Kontext
- 2 Arbeit
- 3 Ergebnisse / Bewertung
- 4 Zusammenfassung**




## Ziel

1.  kontinuierliche hochgenaue Datenerfassung
2. echt parallele Abtastung
3. Echtzeit Datenerfassung
4. Testen auf dem Xilinx Evalboard
5. Entwicklung eines kompakten DAC-Boards





## Ziel

1.  kontinuierliche hochgenaue Datenerfassung
2.  echt parallele Abtastung
3. Echtzeit Datenerfassung
4. Testen auf dem Xilinx Evalboard
5. Entwicklung eines kompakten DAC-Boards






## Ziel

1.  kontinuierliche hochgenaue Datenerfassung
2.  echt parallele Abtastung
3.  Echtzeit Datenerfassung
4. Testen auf dem Xilinx Evalboard
5. Entwicklung eines kompakten DAC-Boards

## Ziel

1.  kontinuierliche hochgenaue Datenerfassung
2.  echt parallele Abtastung
3.  Echtzeit Datenerfassung
4.  Testen auf dem Xilinx Evalboard
5. Entwicklung eines kompakten DAC-Boards

## Ziel

1.  kontinuierliche hochgenaue Datenerfassung
2.  echt parallele Abtastung
3.  Echtzeit Datenerfassung
4.  Testen auf dem Xilinx Evalboard
5.  Entwicklung eines kompakten DAC-Boards

## Ausblick

- ▶ Einbindung des FIFO-Sync Modus an XCL-FSM
- ▶ Downsampling bzw. Datenverarbeitung im FPGA
- ▶ Ethernetstream per Linux (UDP, TCP/IP)
- ▶ A/D-Wandler wechseln für z.B. Logic Analyser
- ▶ Fehlerkorrigierender Code für Datenübertragung an Computer

Vielen Dank für Ihre Aufmerksamkeit.

