

# Simulations

Peter Marwedel  
TU Dortmund, Informatik 12  
Germany

# Gliederung


## Zeitplan

- Einführung
- SystemC
  - Vorlesungen und Programmierung

3,5 Wochen

- FPGAs
  - Vorlesungen
  - VHDL-basierte Konfiguration von FPGAs mit dem XUP VII Pro Entwicklungssystem

3,5 Wochen

- 
- Algorithmen
    - Mikroarchitektur-Synthese
    - Automaten-synthese
    - Logiksynthese
    - Layoutsynthese

6 Wochen

# Simulation example

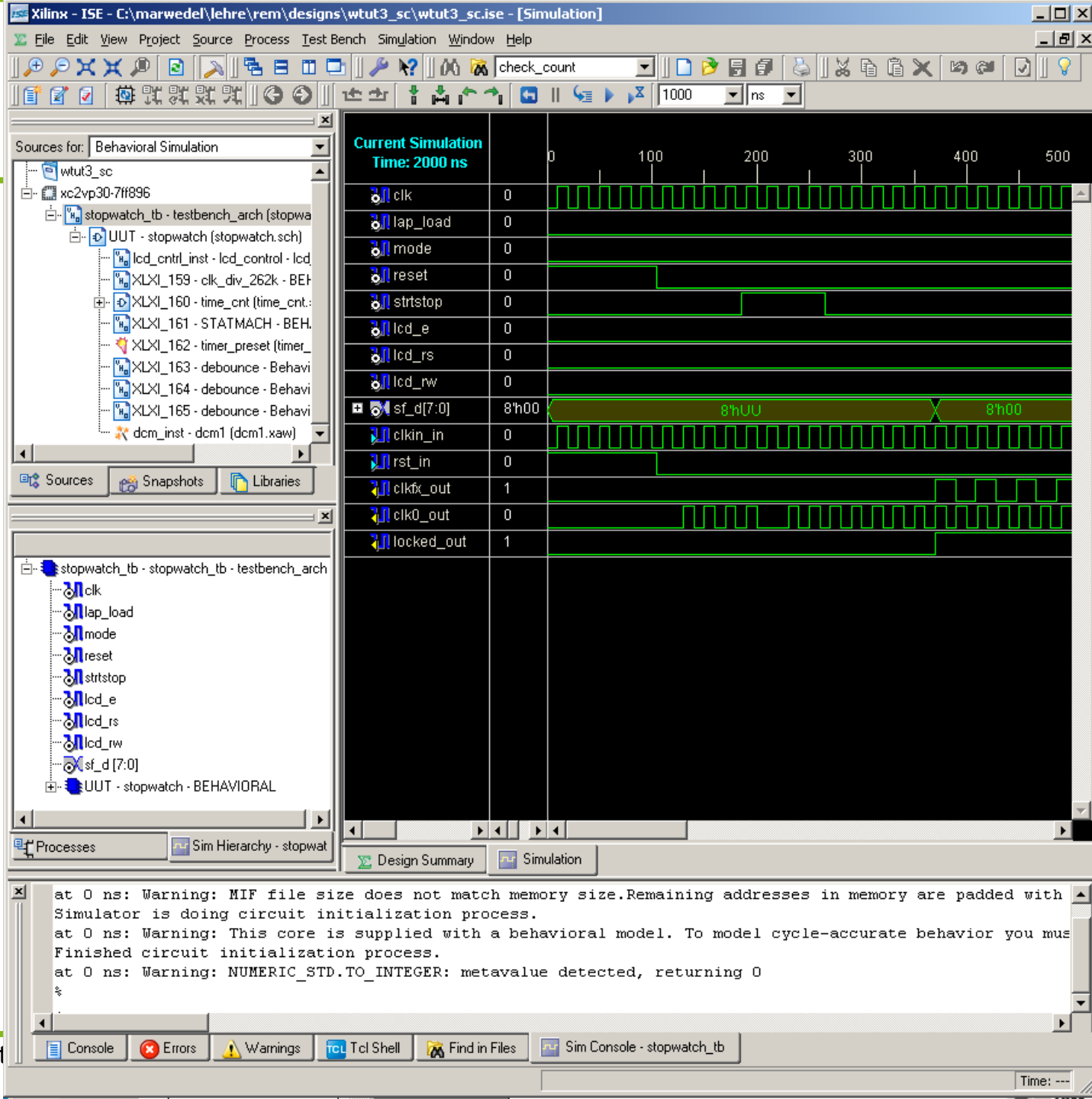
---

## Procedure:

- Start with partial design of runners watch
- Copy all provided files of the completed design into project directory
- Add source stopwat\_tb (testbench) to project
- Select new top level design unit stopwat\_tb
- Select Behavioral simulation view
- In processes tab, double click on behavioral simulation
- Circuit is simulated at 50 MHz range; generating results for a second of real time would take long time
- ☞ Just observe the clock circuit by dragging signals into the waveform area.

Source: [//toolbox.xilinx.com/docsan/xilinx9/books/docs/xst/xst.pdf](http://toolbox.xilinx.com/docsan/xilinx9/books/docs/xst/xst.pdf); Chapter 4

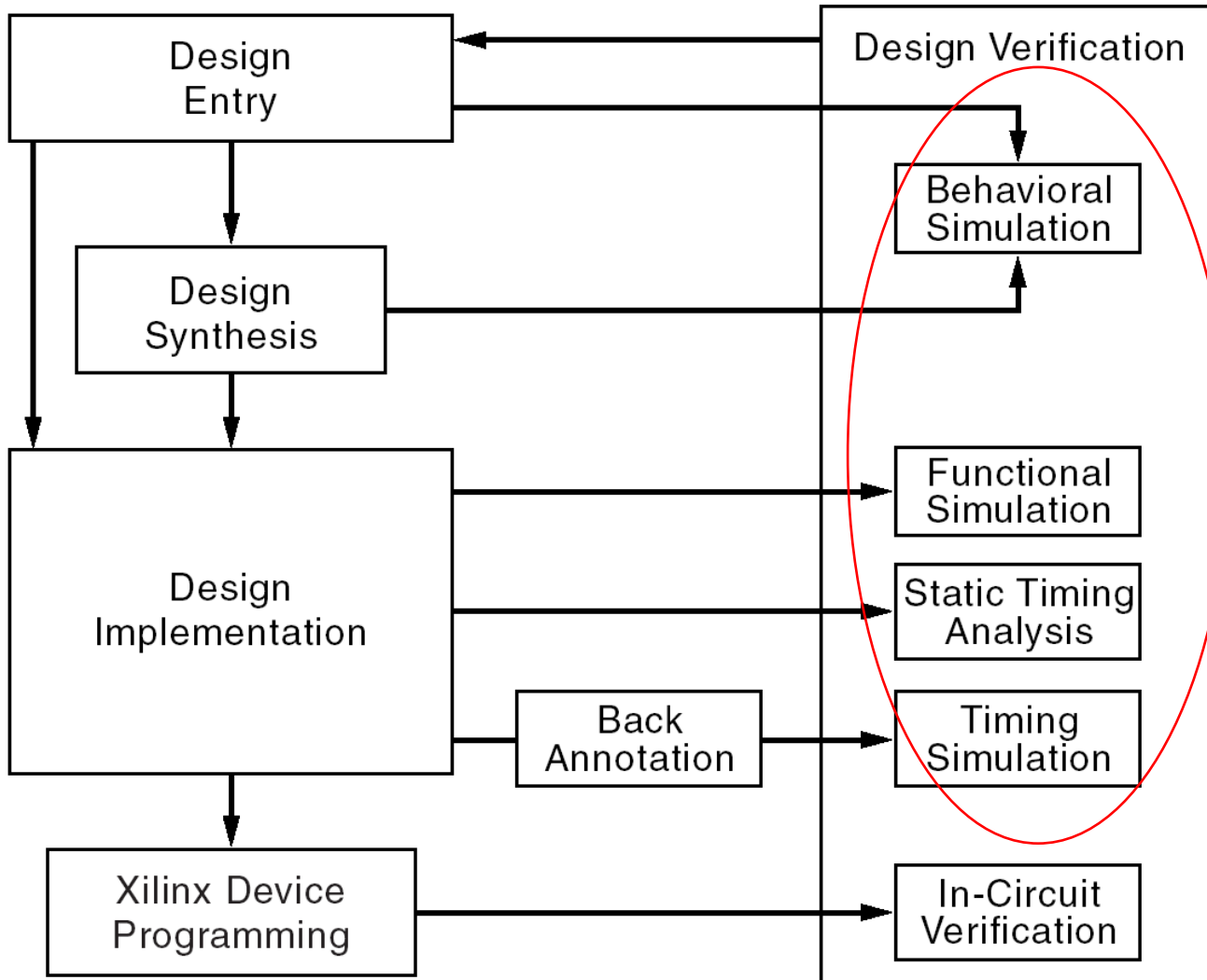
# Simulation result



strtstop manually  
modified

© Xilinx

# Other simulations

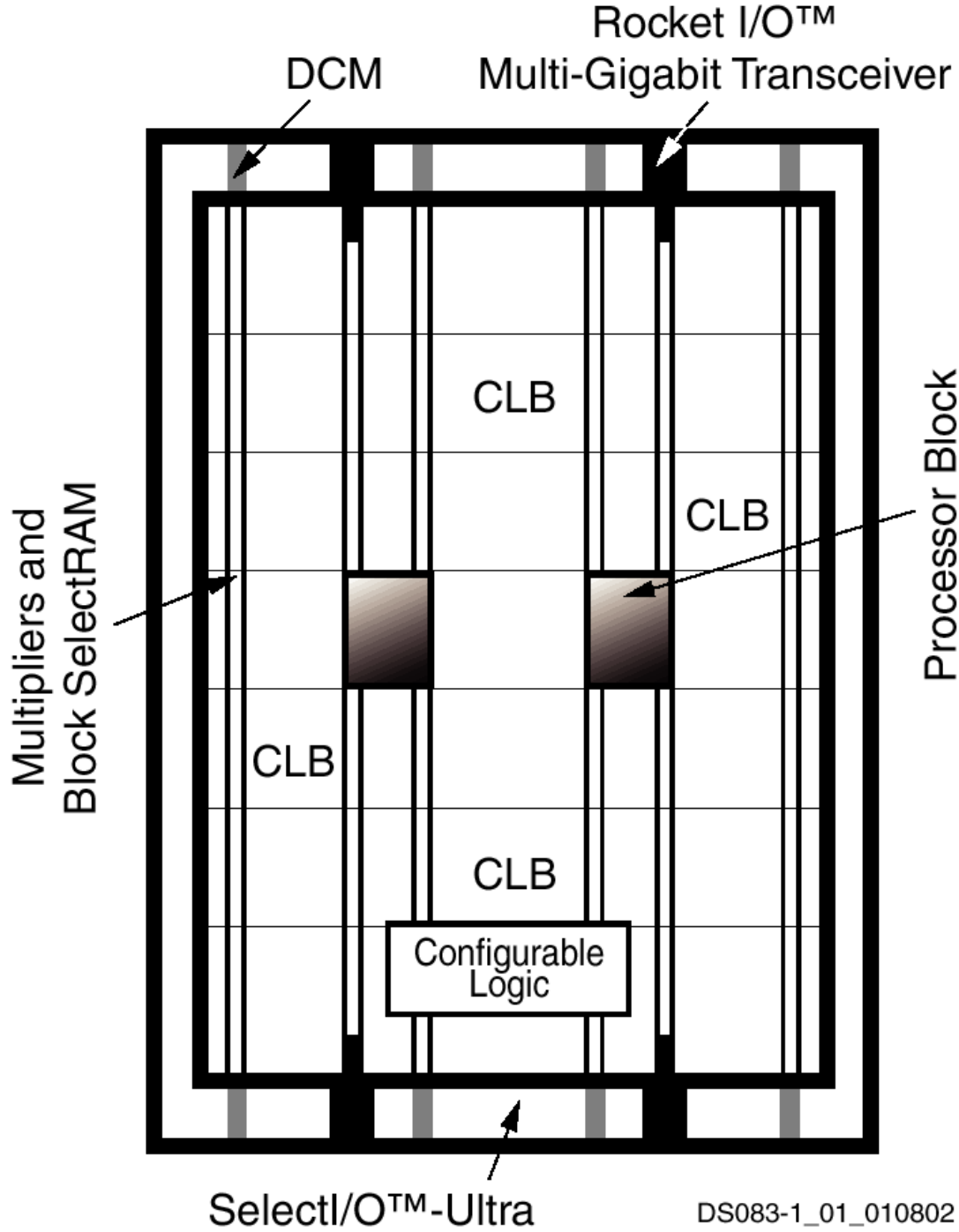


[http://www.xilinx.com/support/sw\\_manuals/xilinx8/download/qst.zip](http://www.xilinx.com/support/sw_manuals/xilinx8/download/qst.zip) (not included in version 9.1i)

# Using PowerPC Cores

Peter Marwedel  
TU Dortmund, Informatik 12  
Germany

# Virtex II Pro Devices include up to 4 PowerPC processor cores



[© and source: Xilinx Inc.: Virtex-II Pro™ Platform FPGAs: Functional Description, Sept. 2002, //www.xilinx.com]

# Die IBM Core Connect Architektur

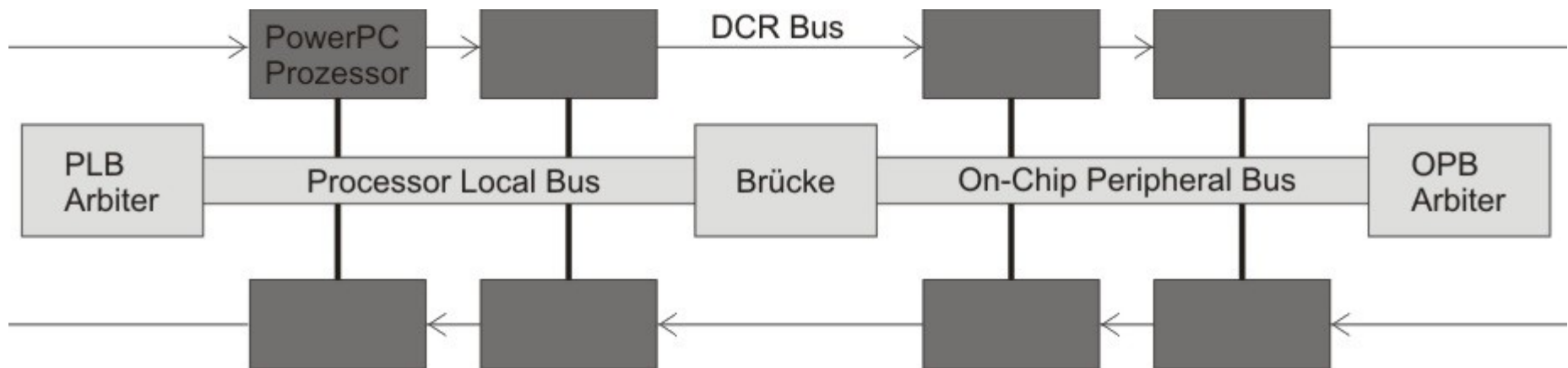
---

Von IBM standardisierte Busarchitektur. Es werden 3 Busse unterschieden:

- PLB Bus: Der *processor local bus* ist ein Hochgeschwindigkeitsbus. 64 Bit breiter Daten- und 32 Bit breiter Adressbus.
- OPB (on-chip peripheral bus) Bus: Für Komponenten, die nicht auf hohe Geschwindigkeit angewiesen sind.  
32 Bit breiter Datenbus  
Es existiert eine Brücke zwischen OPB und PLB, um Datentransfers von *masters* am PLB zu *slaves* am OPB zu ermöglichen.
- DCR Bus: Der *device control register* Bus dient dem Abfragen von Kontroll- und Statusregistern von Komponenten über einen eigenen Bus.



# Verschiedene Anschlussmöglichkeiten



[DA Christopher Heckmann]

# Tools for using the PowerPCs: EDK/XPS

The screenshot shows the Xilinx Platform Studio interface. The main window displays a flowchart titled "Starting your project" with three main sections: Documentation, Examples, and Tech Tips. The flowchart starts with "Starting your project" and branches into "Software Development" and "Hardware Development".

**Starting your project**

- Documentation**
- Examples**
- Tech Tips**

**Software Development**

- Configuring and generating the software platform
- Developing your software applications
  - Using XPS
    - Debugging using XMD and GDB
    - Profiling\*
    - Using SDK\*

**Hardware Development**

- Begin by using the Base System Builder\*
- Creating custom peripherals\*\*
- Populating and connecting your hardware design
- Implementing your hardware platform
- Simulating your embedded sub-system\*
- Debugging hardware using ChipScope Pro™\*

**FPGA Device Configuration**

- Generating the hardware bitstream
- Downloading the complete system bitstream
- Writing embedded software to a flash PROM\*
- Initializing FPGA on-chip memory with embedded software

\*optional/\*\*advanced topics

The interface also shows a menu bar (File, Edit, View, Project, Hardware, Software, Device Configuration, Debug, Simulation, Window, Help), a toolbar, and a status bar at the bottom with "Output", "Warning", and "Error" tabs, and the word "Fertig" (Finished) in the bottom left corner.

Source: [http://www.xilinx.com/support/document/ation/Tutorials/EDK91\\_PPC\\_Tutorial.pdf](http://www.xilinx.com/support/document/ation/Tutorials/EDK91_PPC_Tutorial.pdf)

# Creating the Project File in XPS

---

- XPS allows you to control the HW and SW development of the PowerPC system, and includes:
  - Editor and project management interface for creating and editing source code
  - SW tool flow configuration options
- 1st step: use XPS to create a project file.  
You can use XPS to create the following files:
  - Project Navigator project file for controlling the HW flow
  - Microprocessor Hardware Specification (MHS) file
  - Microprocessor Software Specification (MSS) file

# Starting XPS

- Select Base System Builder Wizard (BSB) to open the Create New Project Using BSB Wizard dialog box shown in [Figure 1](#).
- Click **OK**.
- Use the Project File Browse button to browse to the folder you want as your project directory. Click **Open** to create the system.xmp file then **Save**.
- Click **OK** to start the BSB wizard.

**Note:** XPS does not support directory or project names which include spaces.

[http://www.xilinx.com/support/documentation/Tutorials/EDK91\\_PPC\\_Tutorial.pdf](http://www.xilinx.com/support/documentation/Tutorials/EDK91_PPC_Tutorial.pdf)  
© xilinx

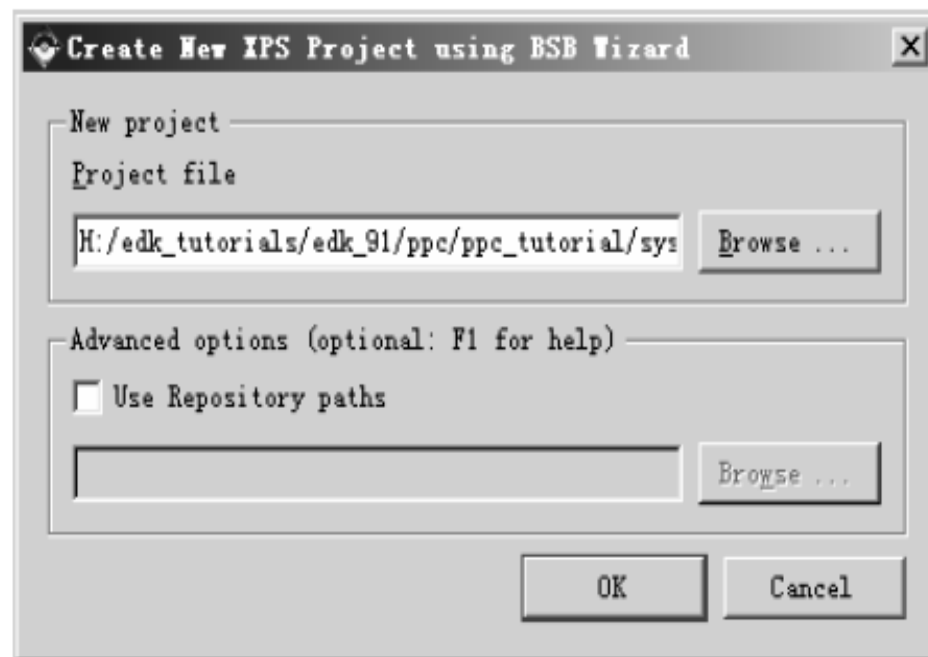


Figure 1: Create New Project Using Base System Builder Wizard

# MHS File

---

The Microprocessor Hardware Specification (MHS) file describes the following:

- Embedded processor: either the soft core MicroBlaze processor or the hard core PowerPC (only available in Virtex-II Pro and Virtex-4 FX devices)
- Peripherals and associated address spaces
- Buses
- Overall connectivity of the system

The MHS file is a readable text file that is an input to the Platform Generator (the hardware system building tool).

Conceptually, the MHS file is a textual schematic of the embedded system. To instantiate a component in the MHS file, you must include information specific to the component.

[http://www.xilinx.com/support/documentation/Tutorials/EDK91\\_PPC\\_Tutorial.pdf](http://www.xilinx.com/support/documentation/Tutorials/EDK91_PPC_Tutorial.pdf) © xilinx

# MPD File

---

- Each system peripheral has a corresponding MPD file. The MPD file is the symbol of the embedded system peripheral to the MHS schematic of the embedded system. The MPD file contains all of the available ports and hardware parameters for a peripheral.
- EDK provides two methods for creating the MHS file. Base System Builder Wizard and the Add/Edit Cores Dialog assist you in building the processor system, which is defined in the MHS file.

[http://www.xilinx.com/support/documentation/Tutorials/EDK91\\_PPC\\_Tutorial.pdf](http://www.xilinx.com/support/documentation/Tutorials/EDK91_PPC_Tutorial.pdf)

# Selecting a board

Base System Builder - Select Board

Select a target development board:

Select board

I would like to create a system for the following development board

Board vendor:

Board name:

Board revision:

Note: Visit the vendor website for additional board support materials.

[Vendor's Website](#) [Contact Info](#)

[Download Third Party Board Definition Files](#)

I would like to create a system for a custom board

Board description

The ML403 board is intended to showcase and demonstrate Virtex-4 technology, especially the new features being added to the FPGA. The ML403 board utilizes Xilinx Virtex 4 XC4VFX12-FF668 device. It is a demonstration platform to showcase the enormous power and flexibility of Virtex-4 FPGAs including new and improved clock technology, DSP blocks, Smart RAM blocks, advanced I/Os, embedded MACs, embedded processors, USB, and more.

# Configure PowerPC

Base System Builder - Configure PowerPC

## PowerPC™

System wide settings

Reference clock frequency: 100.00 MHz

Processor clock frequency: 300.00 MHz

Bus clock frequency: 100.00 MHz

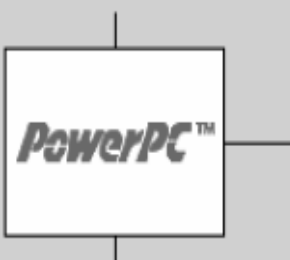
Ensure that your board is configured for the specified frequency.

Reset polarity: Active LOW

Processor configuration

Debug I/F

- FPGA JTAG
- CPU debug user pins only
- CPU debug and trace pins
- No debug



On-chip memory (OCM)

(Use BRAM)

Data: NONE

Instruction: NONE

Cache setup

Enable

For optimal performance, enable burst and/or cacheline on memory



# Configure I/O interfaces

Base System Builder - Configure I/O Interfaces (1 of 4) ? X

The following external memory and IO devices were found on your board:  
Xilinx Virtex 4 ML403 Evaluation Platform Revision 1

Please select the IO devices which you would like to use:

IO devices

RS232\_Uart [Data Sheet](#)

Peripheral: OPB UARLITE

Baudrate (bits per seconds): 57600

Data bits: 8

Parity: NONE

Use interrupt

LEDs\_4Bit [Data Sheet](#)

Peripheral: OPB GPIO

Use interrupt

LEDs\_Positions [Data Sheet](#)

# Configure I/O interfaces (2)

Base System Builder - Configure I/O Interfaces (2 of 4) ? X

The following external memory and IO devices were found on your board:  
Xilinx Virtex 4 ML403 Evaluation Platform Revision 1

Please select the IO devices which you would like to use:

IO devices

- Push\_Buttons\_Position [Data Sheet](#)  
Peripheral: OPB GPIO  
 Use interrupt
- IIC\_EEPROM [Data Sheet](#)
- SysACE\_CompactFlash [Data Sheet](#)
- Cypress\_USB [Data Sheet](#)
- DDR\_SDRAM\_64Mx32 [Data Sheet](#)

# Configure I/O interfaces (3)

Base System Builder - Configure IO Interfaces (3 of 4)

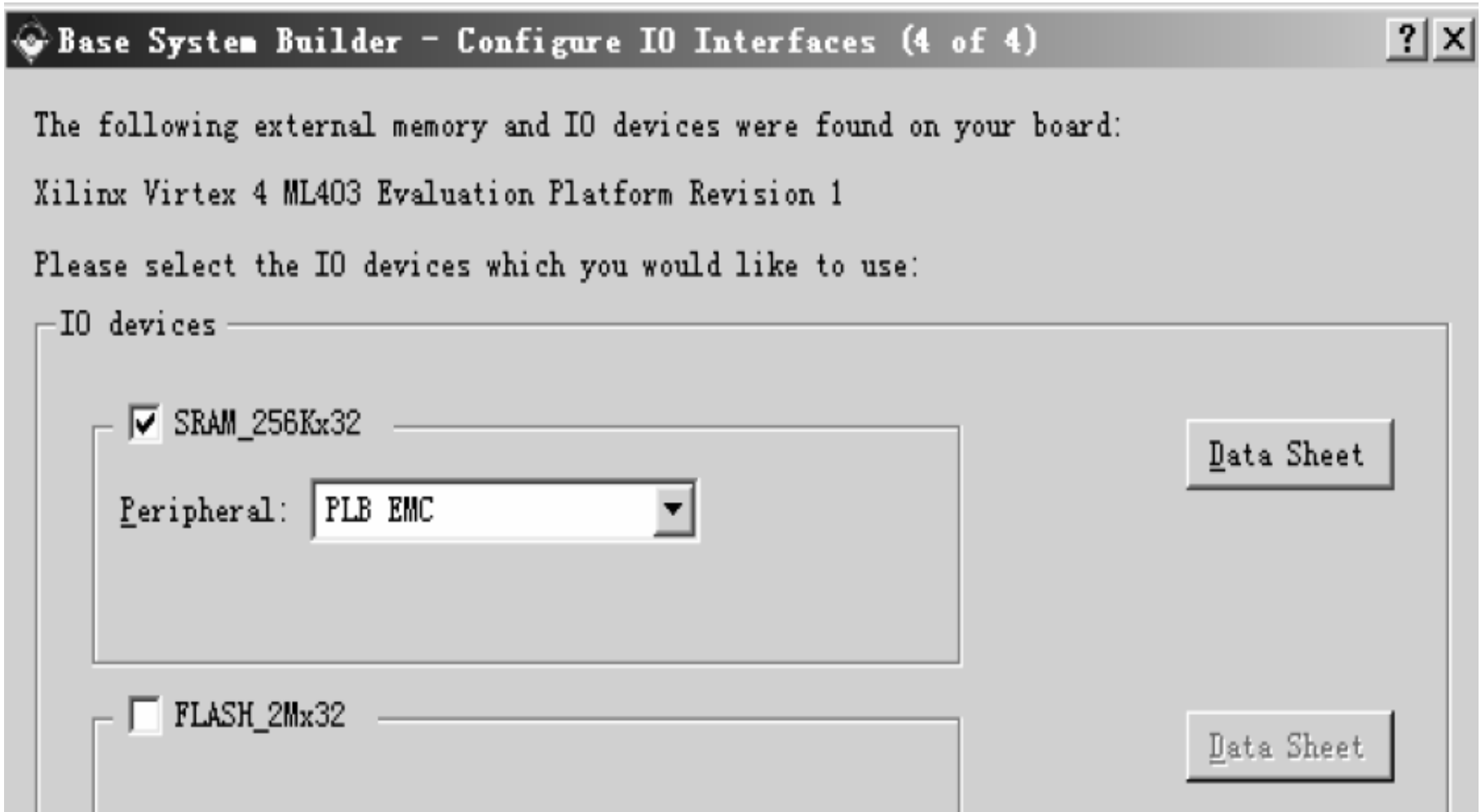
The following external memory and IO devices were found on your board:  
Xilinx Virtex 4 ML403 Evaluation Platform Revision 1

Please select the IO devices which you would like to use:

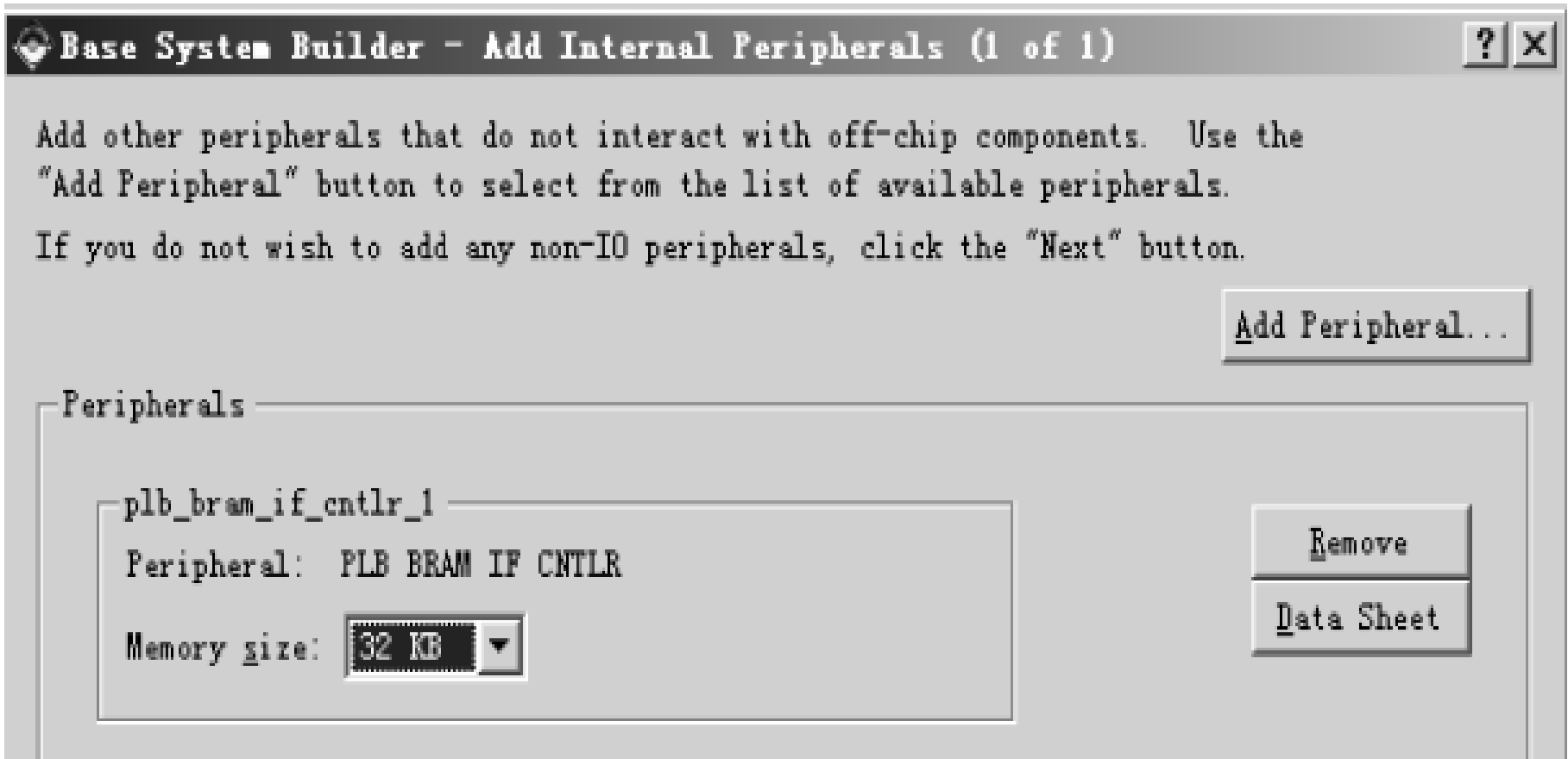
IO devices

- Ethernet\_MAC [Data Sheet](#)  
[Note](#)
- TriMode\_MAC\_GMII [Data Sheet](#)

# Configure I/O interfaces (4)



# Add “internal” peripherals



# Software setup

Base System Builder - Software Setup

Devices to use as standard input, standard output, and boot memory

STDIN: RS232\_Uart

STDOUT: RS232\_Uart

Boot Memory: plb\_bram\_if\_cntlr\_1

Sample application selection

Select the sample C application that you would like to have generated. Each application will include a linker script.

Memory test  
Illustrate system aliveness and perform a basic read/write test to each memory in your system

Peripheral selftest  
Perform a simple self-test for each peripheral in your system.

Below are other software applications found for your board. In order to select an application, please ensure your system satisfy the requirements. See "More Details".

ML403 Cypress USB Application [More Details...](#)

# System Created

Below is a summary of the system you have created. Please review the information below. If it is correct, hit <Generate> to enter the information into the XPS data base and generate the system files. Otherwise return to the previous page to make corrections.

```
Processor: PPC 405
Processor clock frequency: 300.000000 MHz
Bus clock frequency: 100.000000 MHz
Debug interface: FPGA JTAG
On Chip Memory : 32 KB
Total Off Chip Memory : 1 MB
```

The address maps below have been automatically assigned. You can modify them using the editing features of XPS.

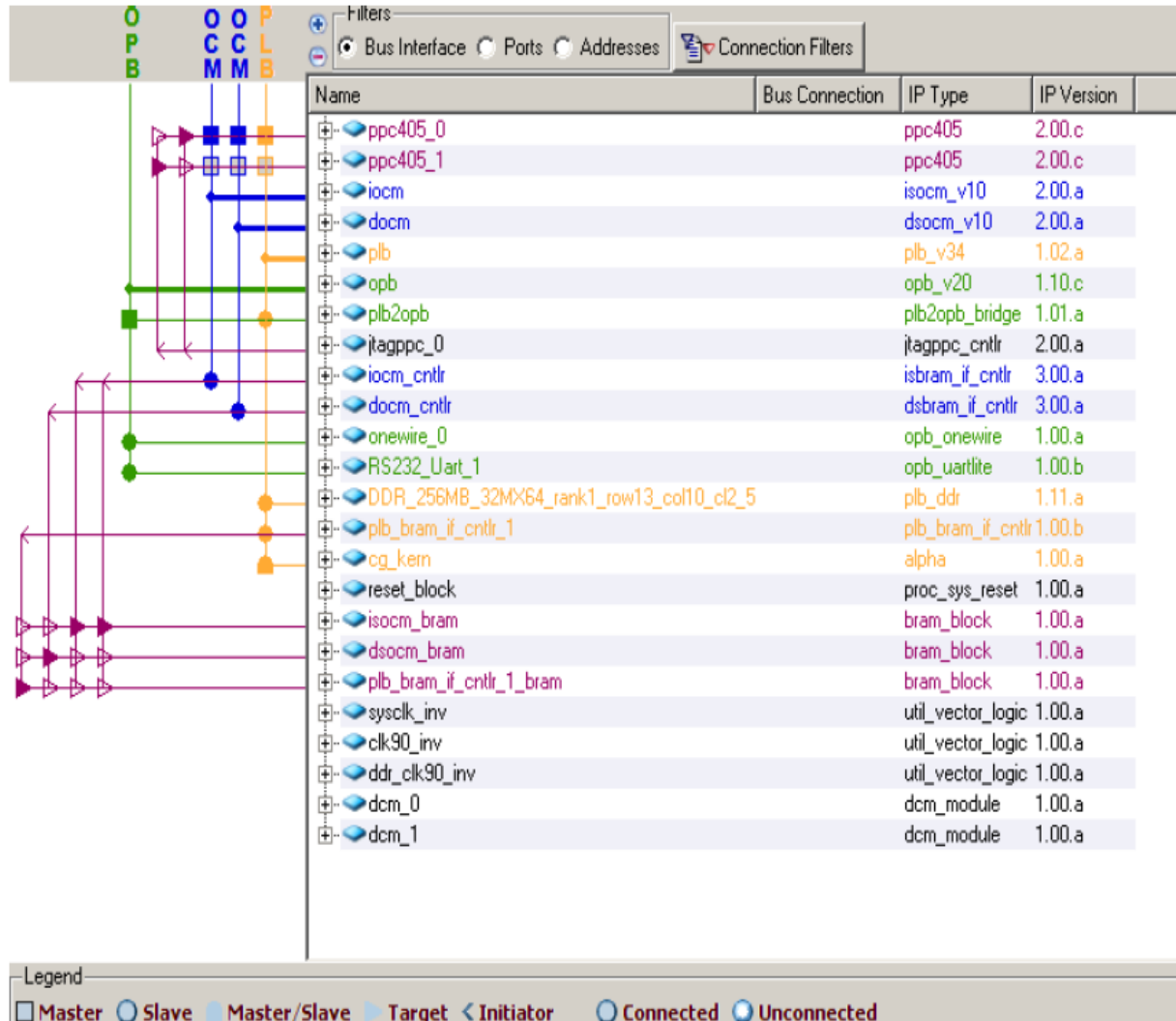
**PLB Bus : PLB\_V34 Inst. name: plb Attached Components:**

Core Name	Instance Name	Base Addr	High Addr
plb2opb_bridge	plb2opb_C_RNGO_BASEAD	0x40000000	0x7FFFFFFF
plb_emc	SRAM_256Kx32	0x00000000	0x000FFFFFF
plb_bram_if_cntlr	plb_bram_if_cntlr_1	0xFFFF8000	0xFFFFFFFF

**OPB Bus : OPB\_V20 Inst. name: opb Attached Components:**

Core Name	Instance Name	Base Addr	High Addr
opb_uartlite	RS232_Uart	0x40600000	0x4060FFFF
opb_gpio	LEDs_4Bit	0x40020000	0x4002FFFF
opb_gpio	Push_Buttons_Position	0x40000000	0x4000FFFF

# Graphische Darstellung der Struktur des Entwurfs



[DA Mirko Sykorra]



# Applications (1)

Peter Marwedel  
Informatik 12, TU Dortmund

# Typical applications

---

- ➔ ■ Diploma theses @ Dortmund: real-time computations
  - Graphics accelerators
  - Encryption/decryption
  - Bio-sequence database scanning
  - Network applications (e.g. network intrusion detection)
  - Parallel pattern recognition in physics
  - Emulation (of new hardware processors)

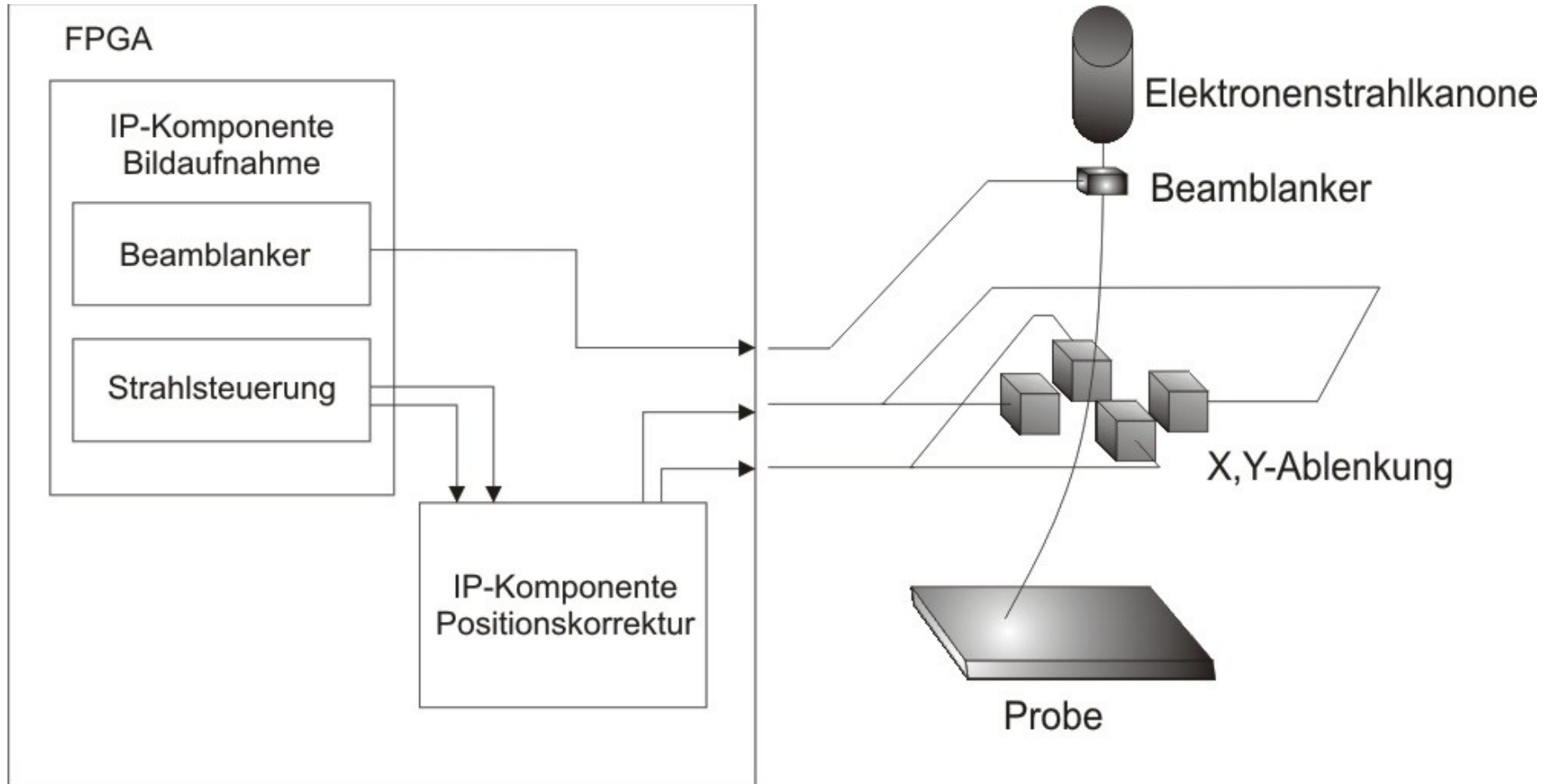
# Bildverarbeitung in der Rasterelektronenstrahlolithographie

Diplomarbeit: Konzeption und Implementierung eines synthetisierbaren VHDL-Kerns zur Bildaufnahme in der Rasterelektronenmikroskopie von Christoph Heckmann, Sept. 2007  
Betreuung: Prof. Götze, Prof. Marwedel; Raith GmbH

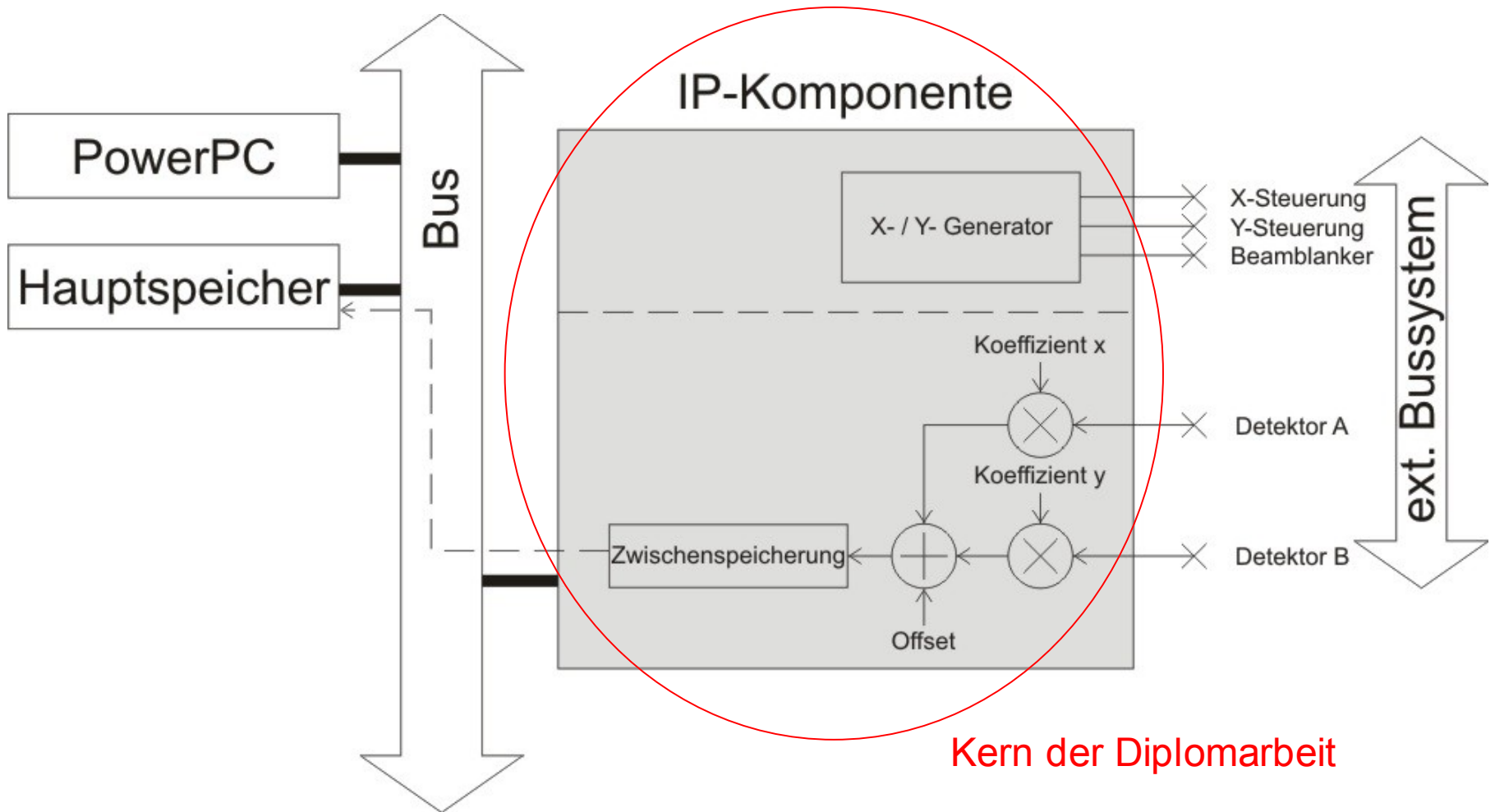


# Anschluss des FPGAs zur Echtzeitsignalverarbeitung

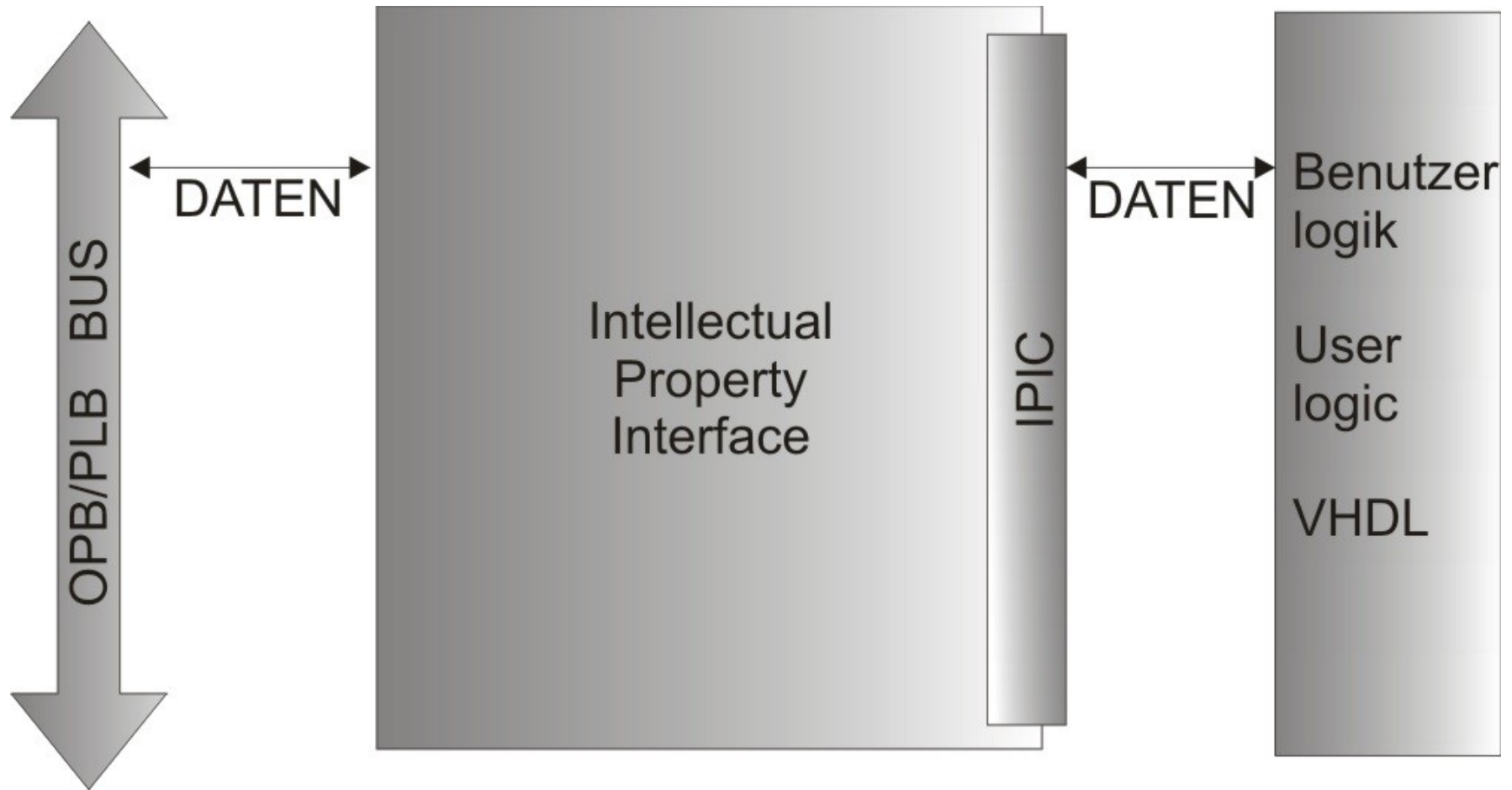
## Ausgehende Signale



# Übersicht über die Hardwareblöcke

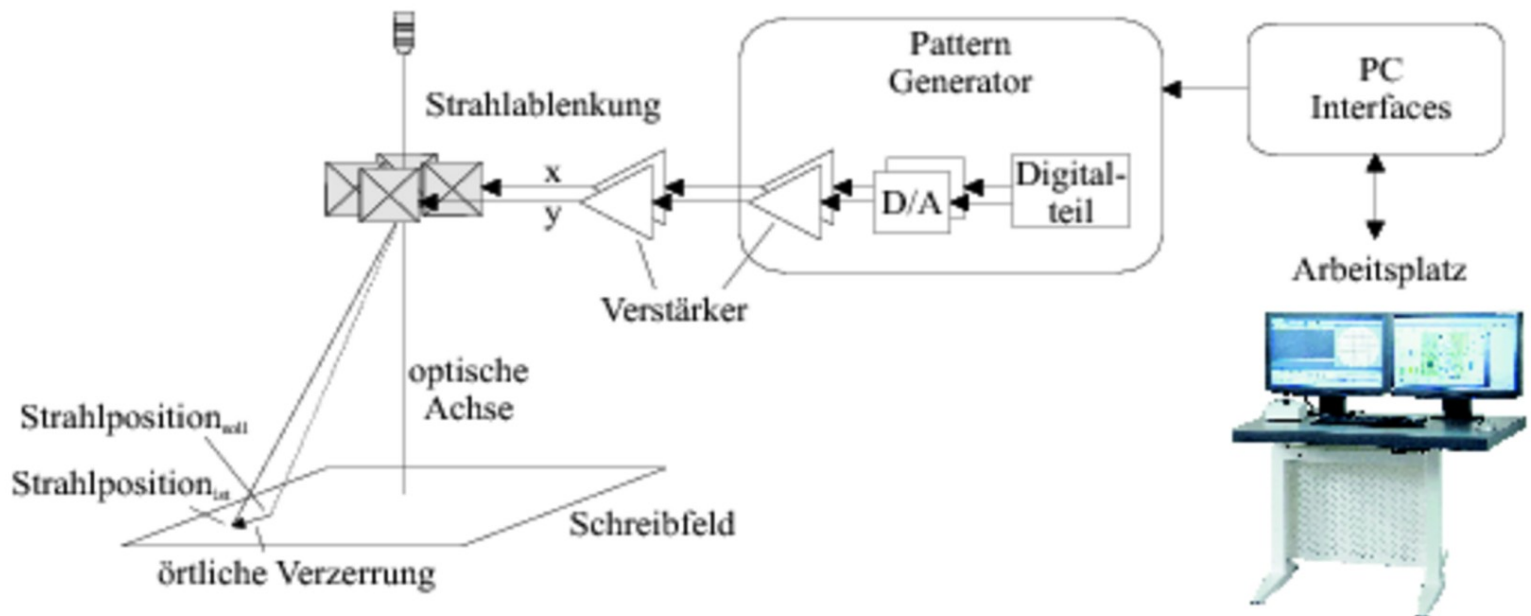


# Logische Entkopplung von speziellem Bus mittels IPIC Interface



# Aktuelle Arbeit

*FPGA Implementierung eines Algorithmus zur Korrektur von Schreibfeldverzerrungen in der Elektronenstrahlolithographie*  
von Mirsad Vejseli. Betreuer: Götze/Marwedel.  
Abgabe: 7.10.2008



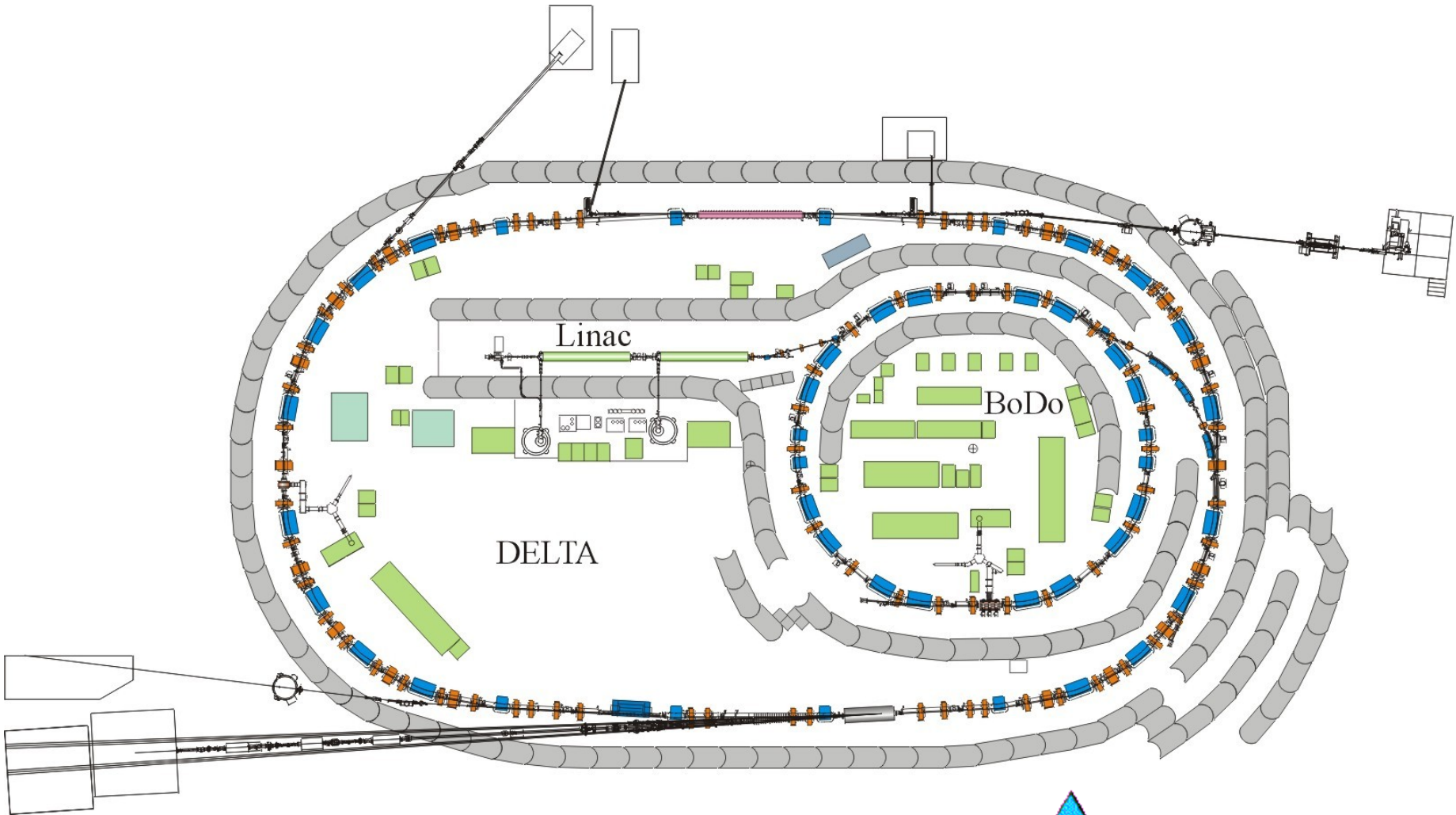
# Strahlkorrektur im Delta-Elektronenstrahling

Diplomarbeit: FPGA  
based data acquisition  
system for fast orbit  
feedback at the electron  
storage ring DELTA von  
Gerrit Schünemann.  
Betreuer: Dr. Hartmann,  
Prof. Marwedel;  
Abgabe: 31.7.2008





# Aufbau von Delta

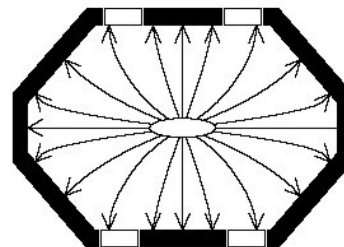
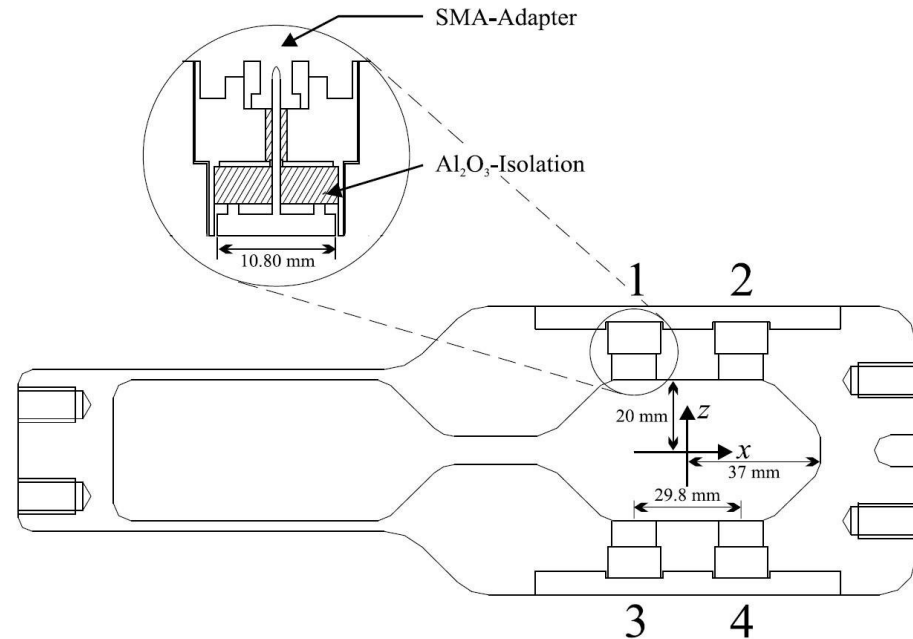
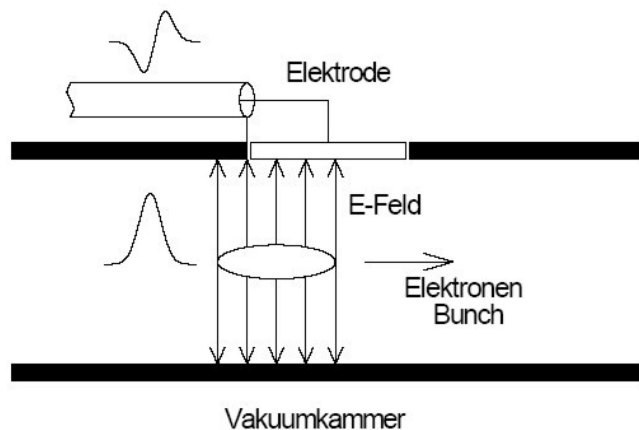


# Elektronenstrahlmessung in der Vakuumkammer

## BPMs

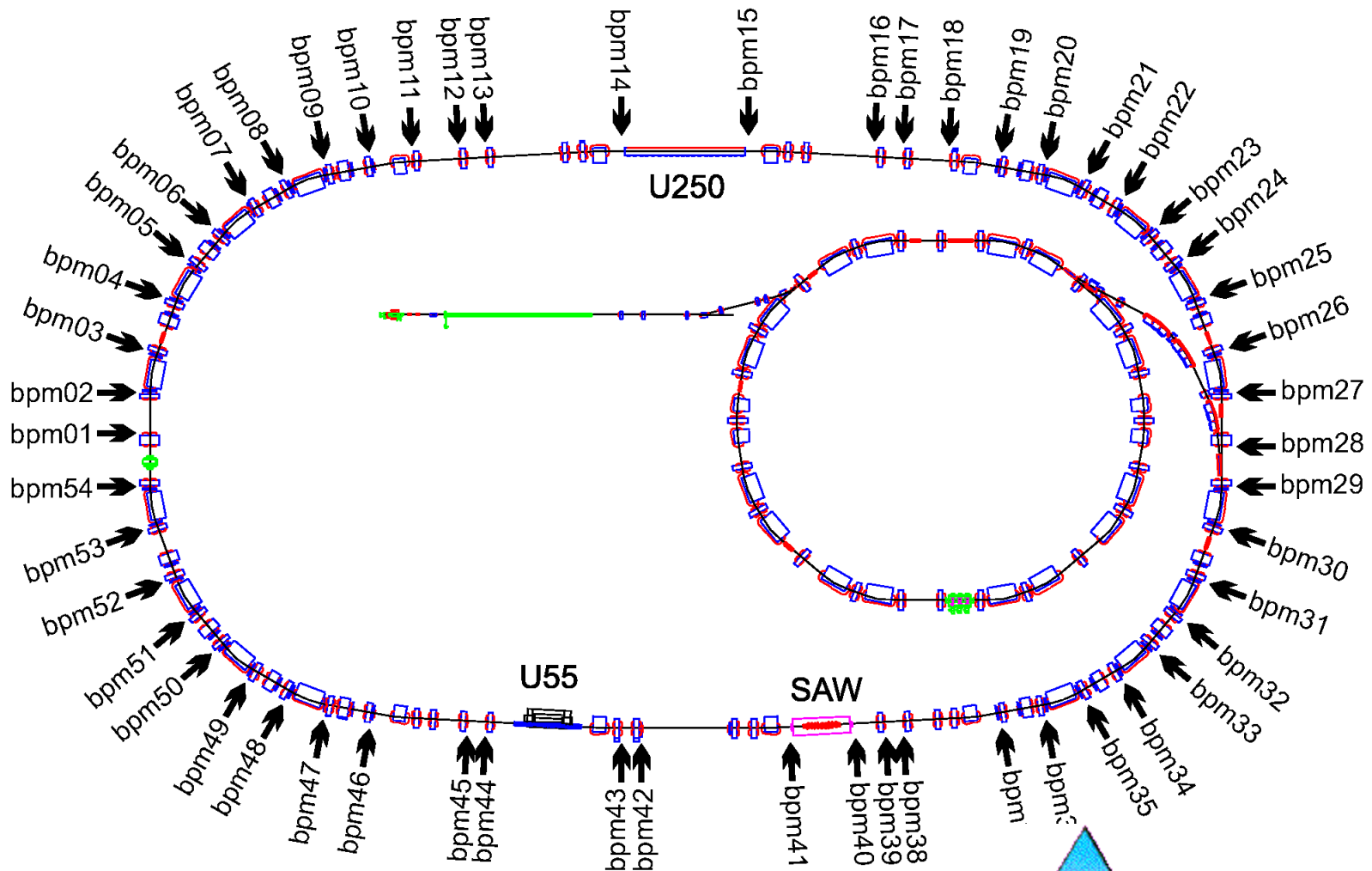
(Beam Position Monitor)

Kammerquerschnitt  
mit BPM-Knöpfen



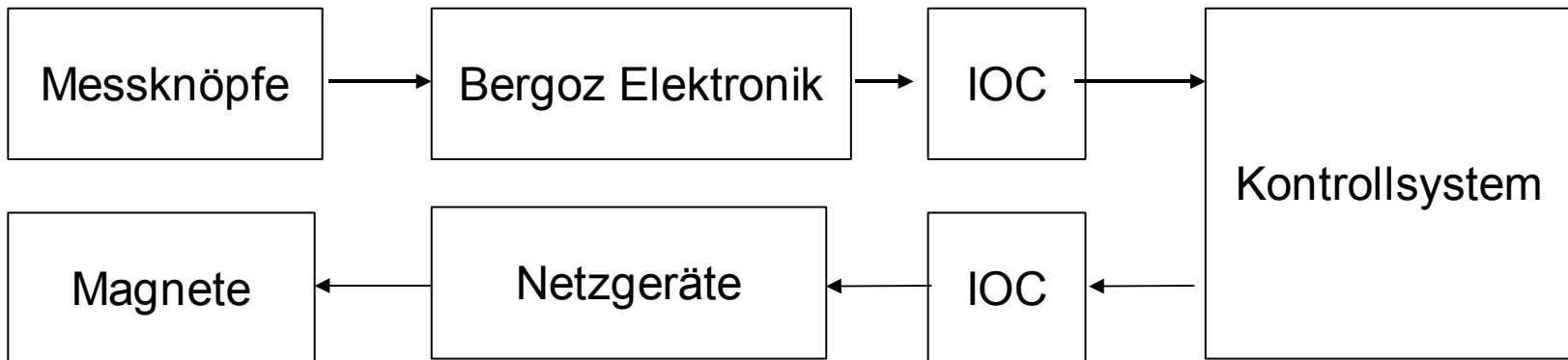
Funktionsweise  
eines BPM-Knopfes

# BPMs am Delta



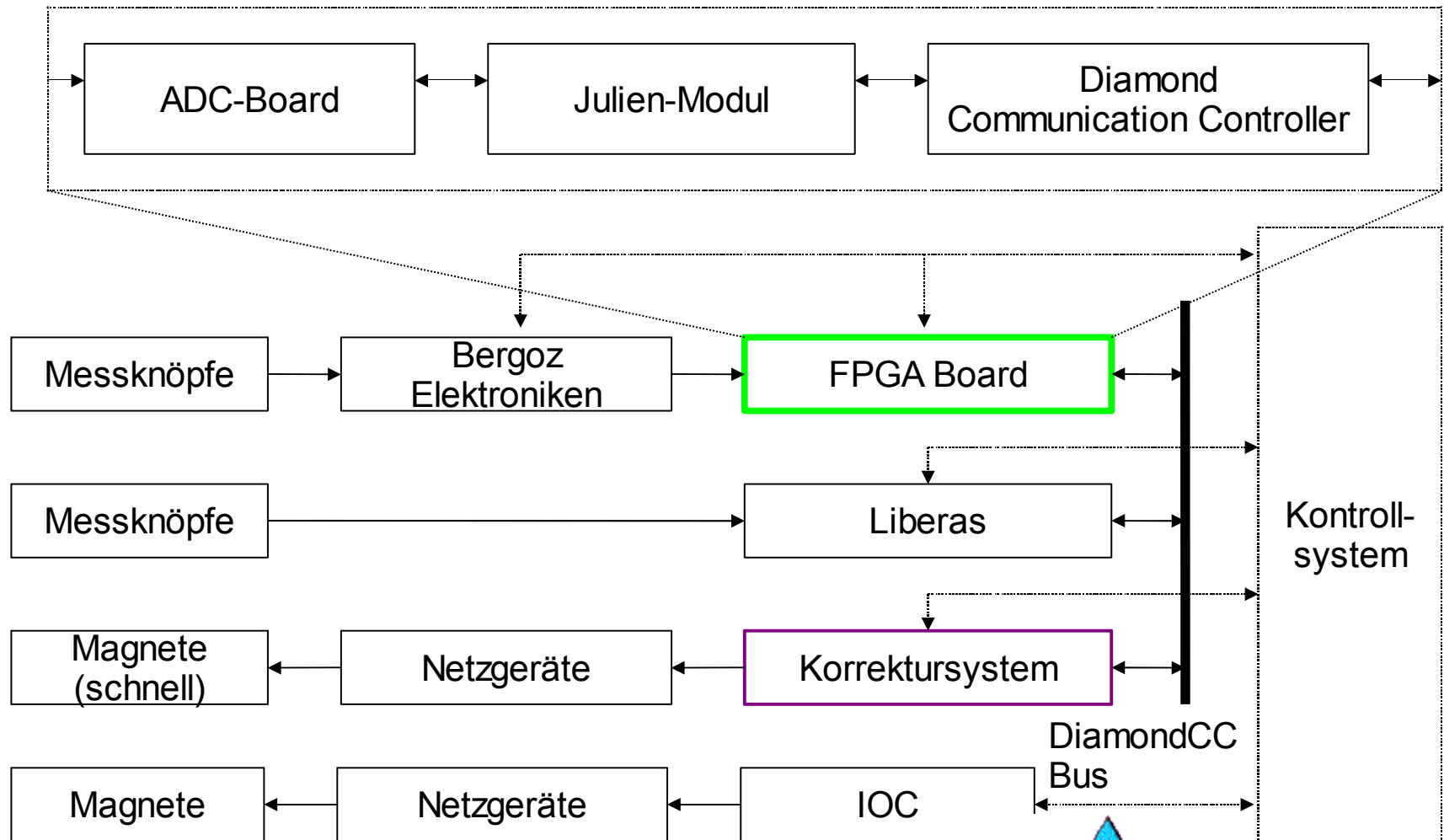
# Langsame Orbitkorrektur am Delta

- Auslesen der Messknöpfe
- Berechnen und Anwenden der Korrekturwerte auf die Magnete
- IOC=Input-Output-Computer -> Bussystem
- Geschwindigkeit am Delta etwa 0.1 Hz



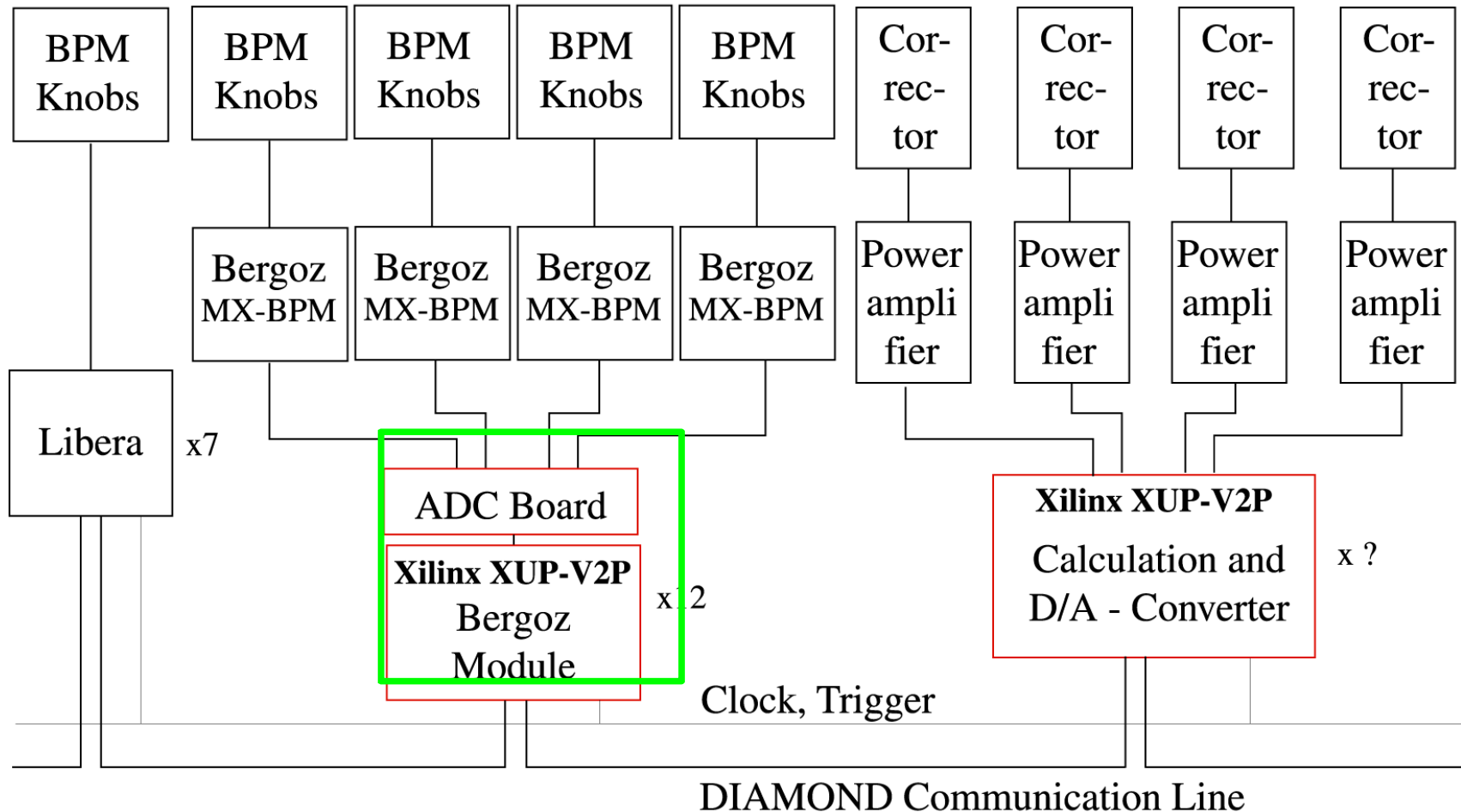
Langsame Orbitkorrektur am Delta

# Planungsdiagramm Diplomarbeit



# Zukünftiges Fast Orbit Feedback am Delta

## Beam



# Weitere Arbeit

---

Diplomarbeit: Lösung partieller Differentialgleichungen mit gemischter Genauigkeit auf *field programmable gate arrays* (FPGAs) von Mirko Sykorra.

Betreuer: Gödecke (FB Mathematik)/Marwedel.

Abgabe: 27.7.2007

Inhalt:

- Implementierung von Gleitkomma-Einheiten in FPGAs
- Kopplung PowerPC/rekonfigurierbare Logik
- Realisierung zeitkritischer Teile in rekonfigurierbarer Logik
- Übrige Teile im PowerPC

# Typical applications

---

- Diploma theses @ Dortmund: real-time computations
- ➔ ■ Graphics accelerators
- Encryption/decryption
- Bio-sequence database scanning
- Network applications (e.g. network intrusion detection)
- Parallel pattern recognition in physics
- Emulation (of new hardware processors)



Source: [www.imm.dtu.dk/visiondag/VD03/grafisk/Visionday-hahl2.ppt](http://www.imm.dtu.dk/visiondag/VD03/grafisk/Visionday-hahl2.ppt)

# An Application Specific Reconfigurable Graphics Processor

DTU



Hans Holten-Lund

IMM, DTU

Graphics Vision Day, 13. Juni 2003



# Overview

- Currently there is a trend for more programmability in graphics hardware.
- Where is the graphics industry headed?
- What about reconfigurable circuits?
  - This could be a comeback for “software” rendering, but not as we know it.
  - Parallel graphics systems?
- Demo:
  - A working graphics processor implemented in an FPGA.

# Current trends

- Past:
  - Fixed function graphics processors.
- Present:
  - Programmable vertex and pixel processors.
- Future:
  - Reconfigurable vertex & pixel datapaths?
    - Modify instruction set of vertex & pixel processors?
  - Reconfigurable architecture?
    - Configurable parallelism?
    - Rasterization algorithm optimizations?
    - Support new graphics algorithms?

# Why reconfigurable?

- Reconfigurable graphics processors could allow:
  - Specialized graphics primitives (e.g. subdivision surfaces)
  - Customized pixel processing (e.g. fragment sorting for correct multilayer transparency)
  - Avoiding software emulation (e.g. when you try to run a DX9 program on DX7-only hardware).
- New ways to process triangles and other primitives:
  - Current GPU's use a fixed triangle processing pipeline.
  - The triangle processing part of the graphics processor should also be programmable, not just vertex processing.
  - Enables: Displacement mapping, procedural geometry, deformable models, collision detection, adaptive level-of-detail, soft shadows, physics simulation, etc...
- New exciting graphics algorithms appear all the time, we would like to have them running in hardware as well!

# Reconfigurable architecture:

- Alternative rendering architectures:
- Tile-based rendering, where triangles are sorted according to screen regions.
  - Tiles can also be rendered in parallel allowing the graphics system performance to scale.
- Framebuffer organization could be more flexible:
  - How about a distributed framebuffer for large display walls?
  - Direct display of a floating-point framebuffer?
- Hardware Ray-Tracing.
  - For global illumination.



# Practical issues with FPGAs:

## ■ Cost issues:

- At low volume, very low cost compared to ASICs.
- At high volume, expensive!

## ■ Speed issues:

- FPGAs are rapidly catching up with standard-cell based ASICs, there is only a factor 5 in clock speed difference today. The gap is getting smaller for each generation.

## ■ Area issues:

- Major concern, as most of the FPGA's chip-area is used for the reconfiguration network. => low silicon area utilization.
- ASIC based graphics processors are 100+ Mtransistor designs using many floating point units, etc.

# Hardware rendered image example

- Image showing the Stanford “Dragon” laserscan rendered using the FPGA hardware implementation of the tile-based Hybris rendering architecture.
- The object contains 870k triangles and was rendered in 2 seconds on an experimental Xilinx Virtex 1000 FPGA board running at 25 MHz.

# “Dragon” rendered by the FPGA

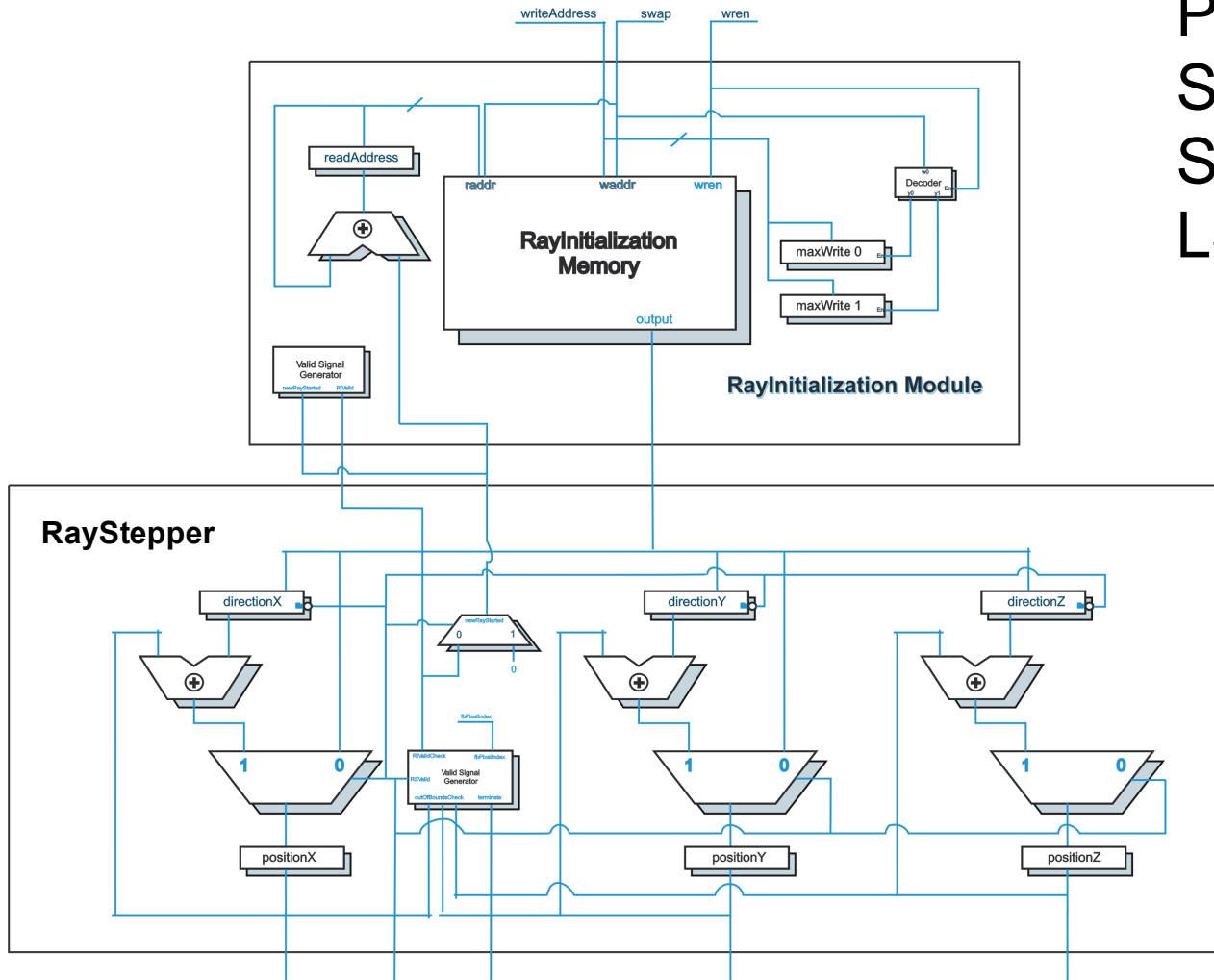






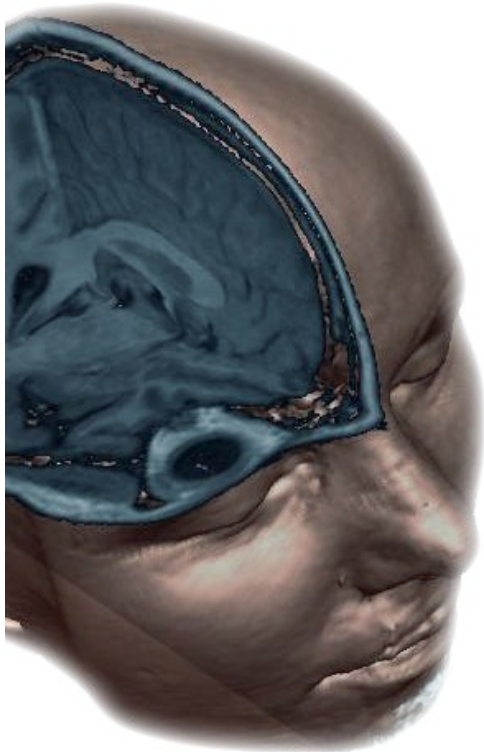
## RTL-Diagramm

Pedram Hadjian,  
Sebastian  
Schmidt: DA  
LS7/LS12





Pedram Hadjian  
Sebastian Schmidt  
Martin Wawro




- Preisträger start2grow 2006



# Typical applications

---

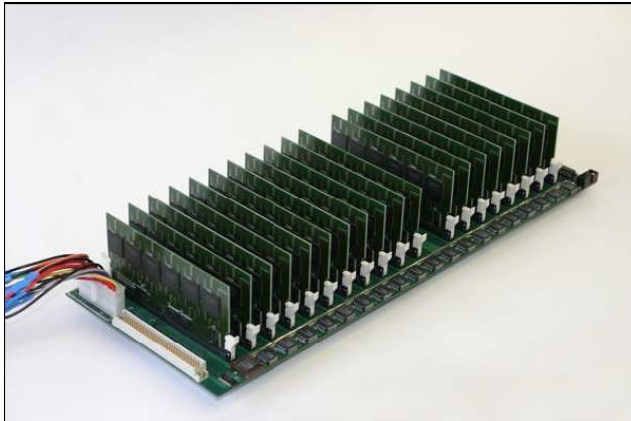
- Diploma theses @ Dortmund: real-time computations
- Graphics accelerators
-  ■ Encryption/decryption
- Bio-sequence database scanning
- Network applications (e.g. network intrusion detection)
- Parallel pattern recognition in physics
- Emulation (of new hardware processors)

# COPACOBANA

---

Machine designed for DES codebreaking  
[Schimmler, Paar, et al. 2006]:

- 20 boards containing 6 Xilinx Spartan FPGAs each
- Cost: < \$10k
- Breaks DES code in 9 days



# Bachelor - Projekt

---

Inhalt des Bachelor-Projekts:

- Beschleunigung von kryptographischen Algorithmen mit rekonfigurierbarer Logik unter Ausnutzung der PowerPC-Prozessoren

# Summary

---

- Simulations
- XPS/EDK
- Applications (1)
  - Diploma theses: real-time applications
  - Graphics accelerators
  - Encryption/decryption