

# Software for development and communication with FPGA based hardware

**Jaroslaw Szewinski, Piotr Pucyk**



PERG Group, Institute of Electronic Systems  
Warsaw University of Technology





# Agenda

- General idea
- System scheme
- Embedded systems
- Development tools
- Future plans



# FPGA features

- Functionality of the FPGA is flexible, and can be changed very often.
- Hardware interface (protocols, buses' widths, etc) may be changed.
- Software must handle all this flexibility.



# Guidelines for software

- Using configuration files when possible (recompilation only in the last resort)
- Modularity:
  - Division into the layers
  - Interfaces defined between layers
  - Implementation as Dynamic Link Libraries (DLL) or Shared Objects (SO)



11/10/2005

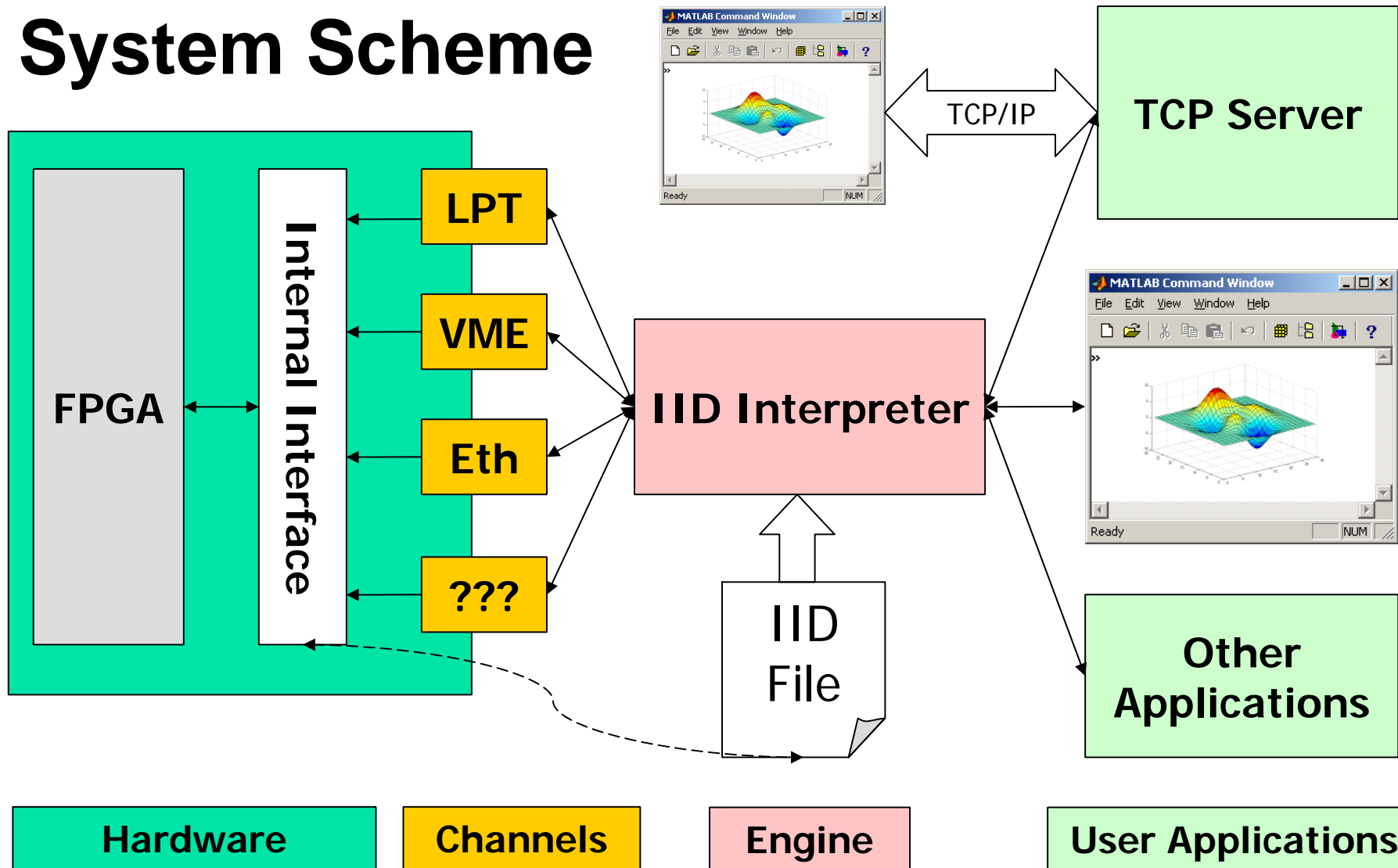
PERG Group, Institute of Electronic Systems  
Warsaw University of Technology

*LLRF05 Workshop*

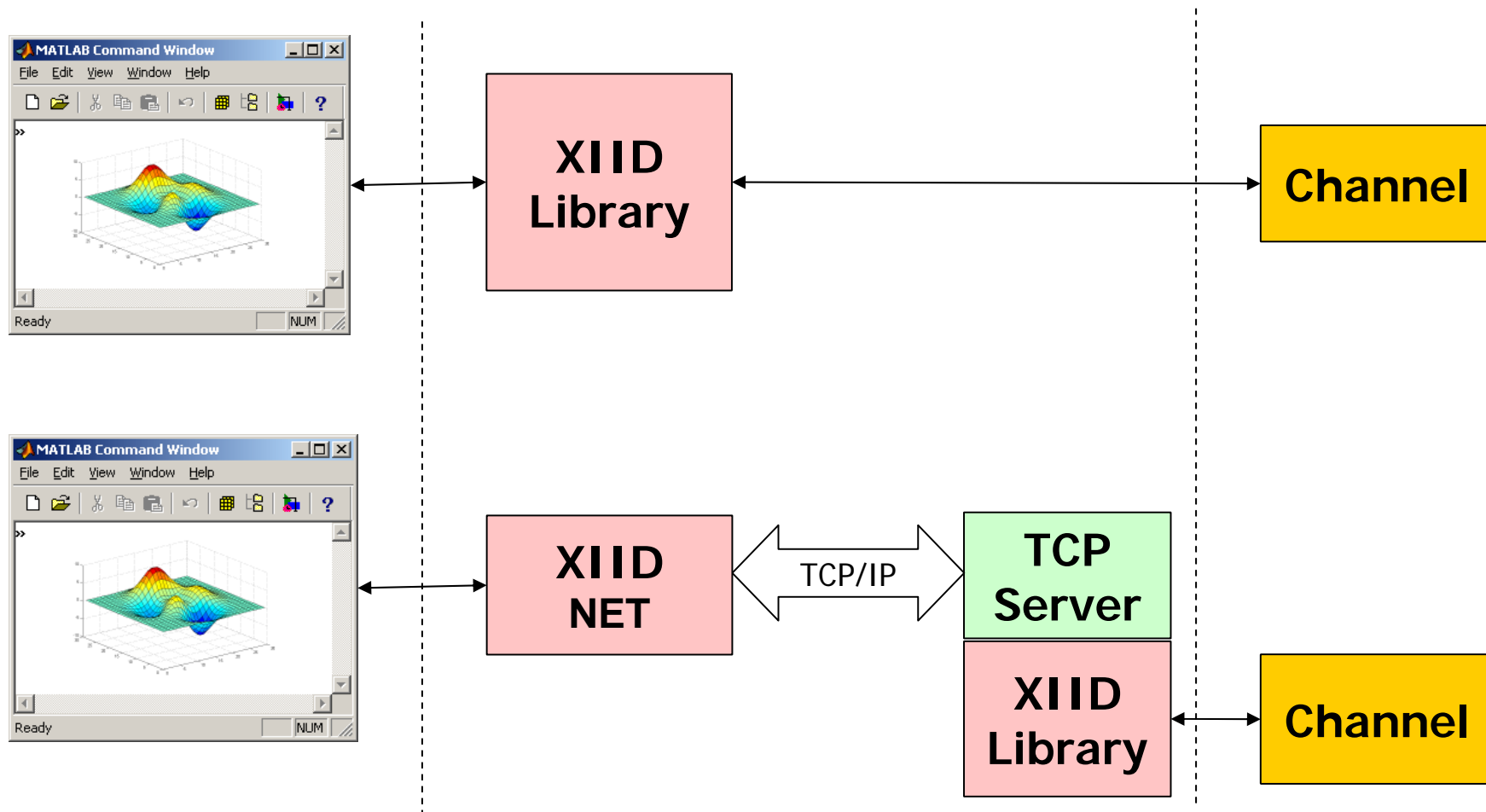
4



# System Scheme



# TCP/IP Connection





# Control Systems

Using these hardware access interfaces two control systems has been set up:

- DOOCS – Distributed Object Oriented Control System developed at DESY
- Our own Matlab based toolset (used mostly for laboratory development)



# Embedded Systems

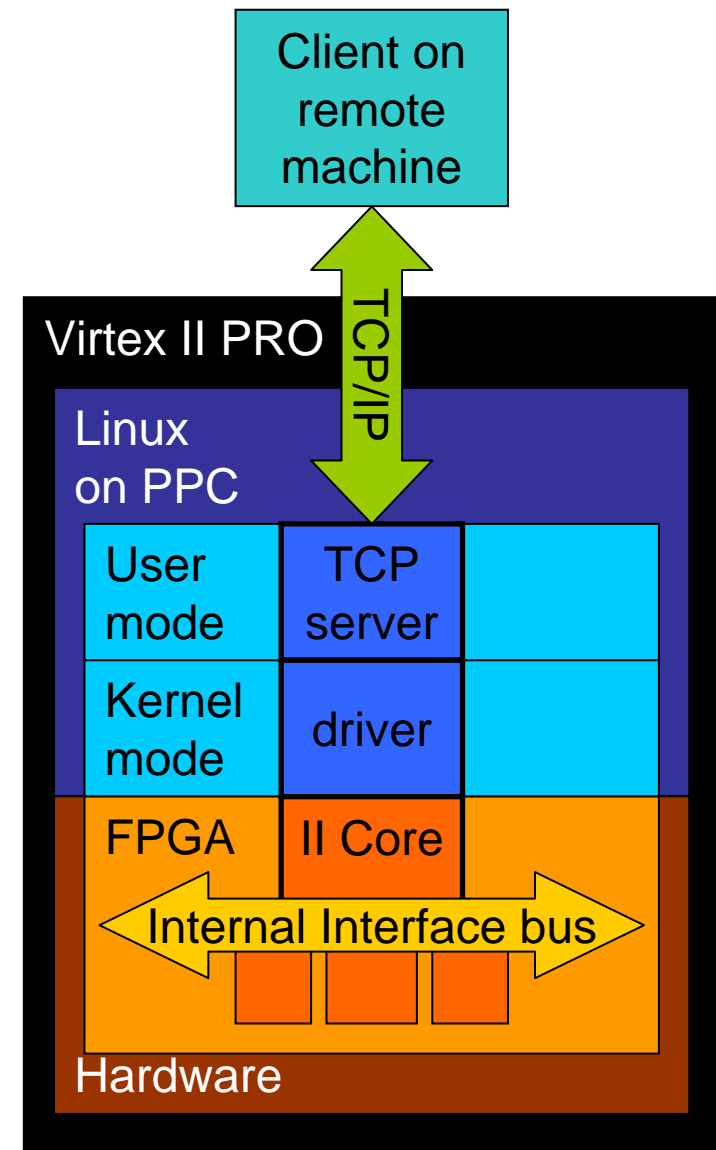
- External RISC CPU - Axis Etrax
- FPGA chip with CPU - Xilinx Virtex II Pro
- Soft processors (described in VHDL) implemented in programmable logic – NIOS, MicroBlaze and others
- Dedicated Linux distributions are available for all examples listed above





# Linux on PowerPC

- User applications access hardware through the driver
- Kernel mode driver has access to the FPGA
- FPGA has defined hardware interface





# Common usage

## ■ Calculations

- non time critical algorithms
- post processing

## ■ Communication

- trough the VME bus
- Standalone, using Ethernet and TCP/IP



# Currently available software

- X11 window system
- DOOCS server libraries
- Scripting languages: perl, tcl
- Secure shell (ssh)
- Apache web server
- Our custom tools (drivers, servers, etc)



# Development Tools

The screenshot displays three windows from a development environment:

- IID Editor:** Shows a tree view of 'PAGE REGISTERS' and a table of parameters.
 

Name	Type	Value	Comment
HARD_ADDR_WIDTH	TVL	16	
HARD_DATA_WIDTH	TVL	32	
LPM_CLOCK_FREQ	TN	40	frequenc...
LPM_STEP_CLK_NUMBER	TN	LPM_...	frequenc...
STEP_CLK_WIDTH	TN	TVLcr...	
LPM_VM_DRV_NUMBER	TN	4	
VM_DRV_WIDTH	TN	TVLcr...	
LPM_CAVITY_STEPS	TN	1000...	*[1 us] = ...
CAVITY_STEPS_WIDTH	TN	TVLcr...	
LPM_AVER_SHIFT_MAX	TN	3	
AVER_SHIFT_WIDTH	TN	TVLcr...	
CAV_DELAY_SIZE	TN	4	
GAIN_BEAM_SIZE	TN	12	oryginaly...
CTRL_GAIN_DOT_POS	TN	5	
CTRL_COMP_DOT_POS	TN	16	
LPM_DSP_WIDTH	TVL	18	original...
- VME over EPP:** A control panel with fields for VMEWAddr (1), VMERdAddr (2), VMEAddrMode (3), TimeOut (6), UserRegister / Addr32 (7), and EPPOperMode (4). It includes 'Read' and 'Write' buttons for each field and a 'Status' section with checkboxes for ProtoErr, CheckErr, VMEBusErr, VMEDtack, and VMETmOut.
- Register Diagram:** A schematic showing three registers: Status Register (bits S0-S7), Data Register (bits D7-D0), and Control Register (bits C0-C7). The registers are connected to a bus structure.



# Future Plans

- Porting EPICS to our Linux distribution
- Running VxWorks on PowerPC
- Research on Altera's NIOS processor
- Completing full communication layer for the VUV-FEL experiment

