JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

# Compute Node Design for DAQ and Trigger Subsystem in Giessen

Justus-Liebig University in Giessen

# Outline

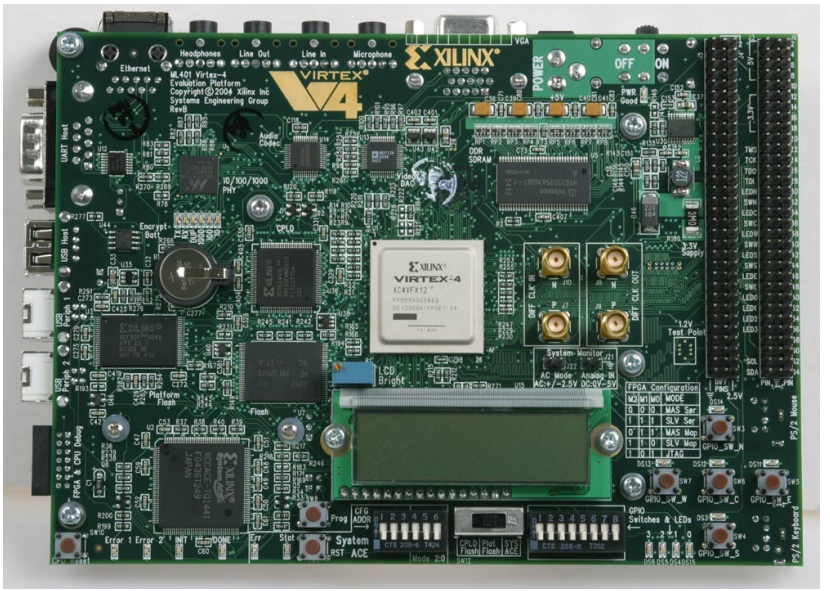
- Design goals
- Current work in Giessen
  - Hardware
  - Software
- Future work

# Design Goals

For compute networks and nodes in PANDA, they are expected to have:

- High computing and communication capabilities to process large amount of data.
- Highly universal, configurable and scalable platform for multiple applications.
- Convenience to be modified and upgraded in-field.

# Present Work in Giessen




- Prototype design
- Based on ML403 commercial board from Xilinx.

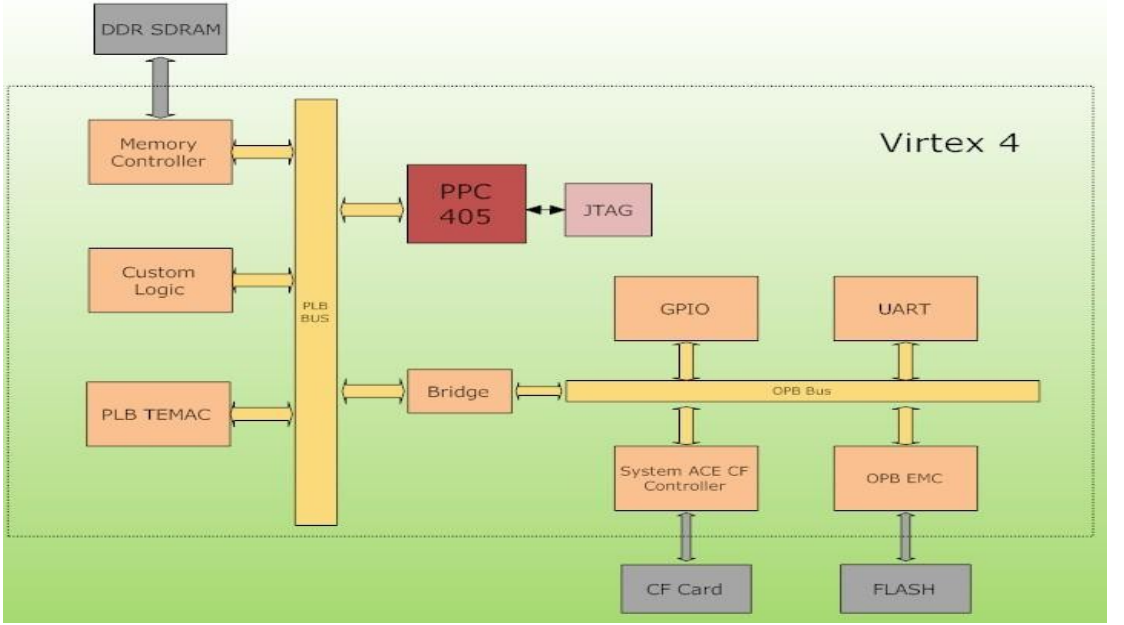
# ML403 Evaluation Board

- **Xilinx Devices:**
  - Virtex 4 XC4VFX12-FF668-10 FPGA
- **Memory:**
  - 64 MB DDR SDRAM
  - 8 Mb SRAM
  - 8 MB Linear Flash
  - 32 Mb Platform Flash
  - System ACE CF card
  - 4 Kb IIC EEPROM
- **Interfaces:**
  - 10/100/1000 RJ-45 Ethernet Port
  - RS-232 Serial Port
  - JTAG
  - 3 USB Ports
  - 2 PS/2 Connectors (Keyboard/Mouse)
  - 2 Audio (Microphone/Head Phone)
  - General Purpose I/O: Buttens and LEDs
  - ……
- **Display:**
  - 16 x 2 Character LCD
  - DB 15 VGA Display
- **Clocks:**
  - 100 MHz Oscillator
  - 2 Clock Sockets

# Virtex 4 FX FPGA

| Device | Configurable Logic Blocks (CLBs)(1) | | | | XtremeDSP Slices(2) | Block RAM | | DCMs | PMCDs | PowerPC Processor Blocks | Ethernet MACs | RocketIO Transceiver Blocks | Total I/O Banks | Max User I/O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Array(3) Row x Col | Logic Cells | Slices | Max Distributed RAM (Kb) | | 18 Kb Blocks | Max Block RAM (Kb) | | | | | | | |
| XC4VSX25 | 64 x 40 | 23,040 | 10,240 | 160 | 128 | 128 | 2,304 | 4 | 0 | N/A | N/A | N/A | 9 | 320 |
| XC4VSX35 | 96 x 40 | 34,560 | 15,360 | 240 | 192 | 192 | 3,456 | 8 | 4 | N/A | N/A | N/A | 11 | 448 |
| XC4VSX55 | 128 x 48 | 55,296 | 24,576 | 384 | 512 | 320 | 5,760 | 8 | 4 | N/A | N/A | N/A | 13 | 640 |
| XC4VFX12 | 64 x 24 | 12,312 | 5,472 | 86 | 32 | 36 | 648 | 4 | 0 | 1 | 2 | N/A | 9 | 320 |
| XC4VFX20 | 64 x 36 | 19,224 | 8,544 | 134 | 32 | 68 | 1,224 | 4 | 0 | 1 | 2 | 8 | 9 | 320 |
| XC4VFX40 | 96 x 52 | 41,904 | 18,642 | 291 | 48 | 144 | 2,592 | 8 | 4 | 2 | 4 | 12 | 11 | 448 |
| XC4VFX60 | 128 x 52 | 56,880 | 25,280 | 395 | 128 | 232 | 4,176 | 12 | 8 | 2 | 4 | 16 | 13 | 576 |
| XC4VFX100 | 160 x 68 | 94,896 | 42,176 | 659 | 160 | 376 | 6,768 | 12 | 8 | 2 | 4 | 20 | 15 | 768 |
| XC4VFX140 | 192 x 84 | 142,128 | 63,168 | 987 | 192 | 552 | 9,936 | 20 | 8 | 2 | 4 | 24 | 17 | 896 |

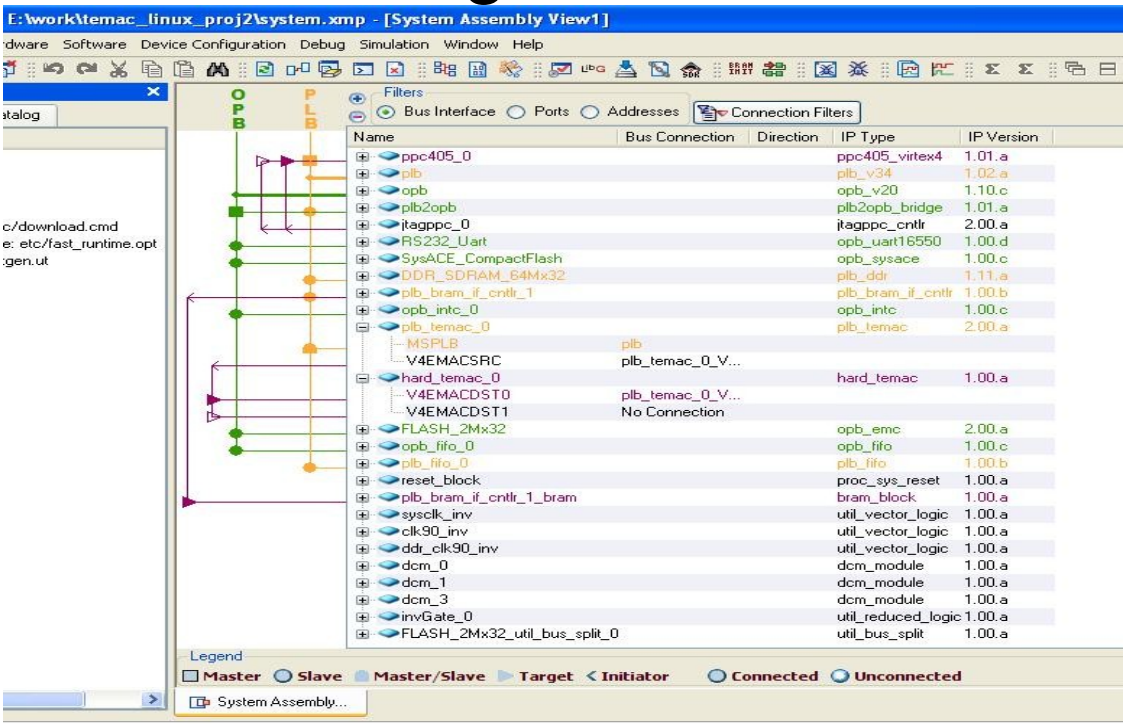
# Hardware Platform



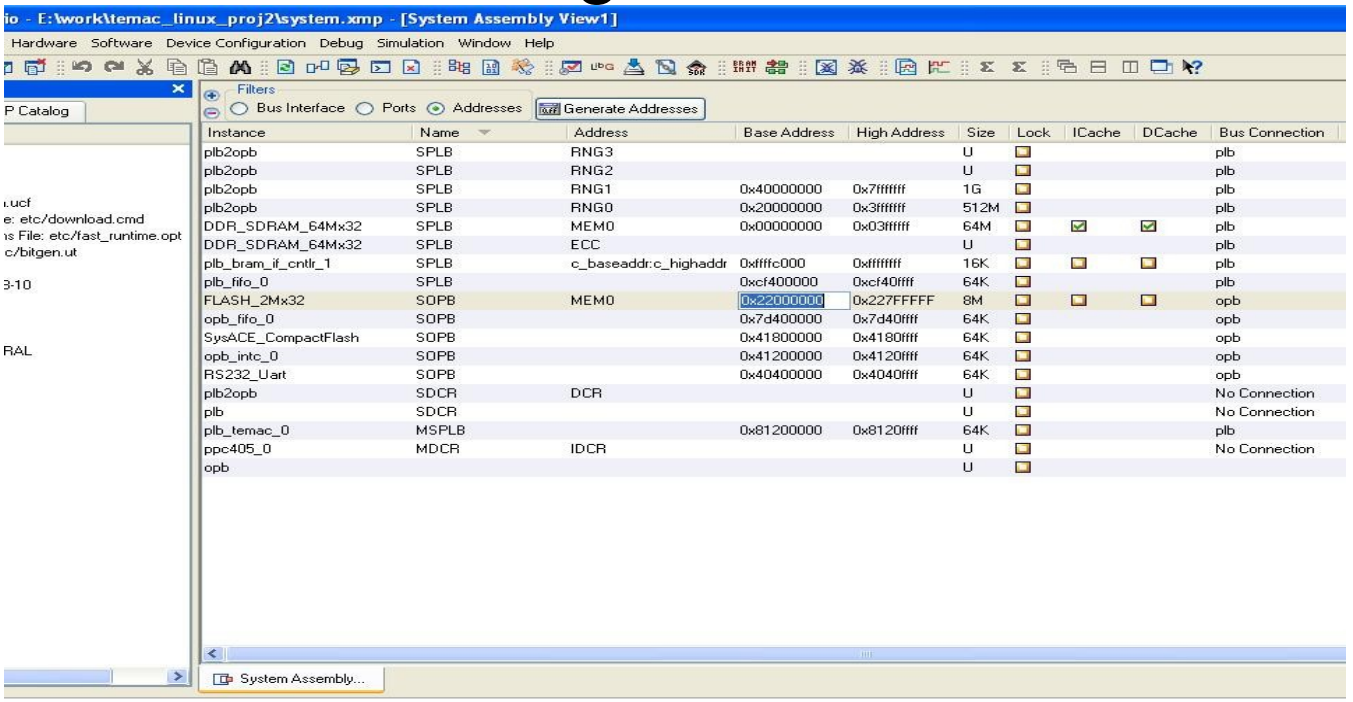
# Hardware Platform

- Most components are inside the FPGA, such as Memory Controller, Tri-mode Ethernet, and Custom Logic.
- The programmable capability of FPGA makes it easy to modify and scale the system.
- Via custom logic design, the hardware platform could be universal for multiple applications.
- Custom logic could be designed with HDL (VHDL or Verilog) for different purposes, for instance, event-select algorithm.
- Design software:
  - EDK 8.1: high level embedded systems design.
  - ISE 8.1: logic design.
  - Modelsim 6.0: simulation.

# Design in EDK

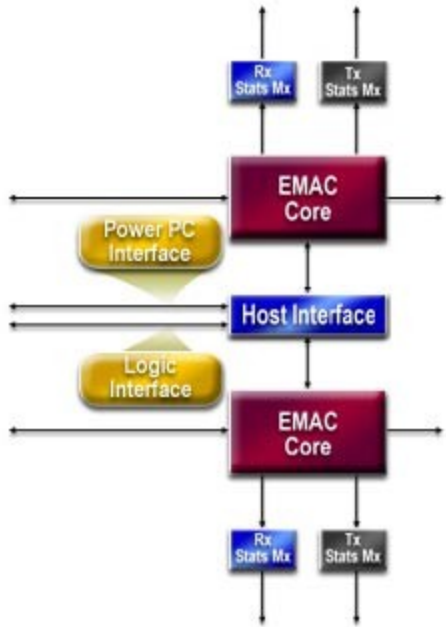


# Design in EDK



# Tri-mode Ethernet

- 2 Tri-mode Ethernet MACs in V4FX12.
- 10/100/1000 Mbps modes. (1000 Mbps is most interesting for us.)
- Gbit Ethernet provides guaranteed communication performance for interconnected networks.



# Flash Memory

- FPGA configuration data and operating system image are both stored in flash memory chips.
- The hardware platform could be updated by this method: 1. power on and boot the system. 2. update the configuration data in flash. 3. reboot the system. (talk about it later in detail)
- Also the operating system image.

# Software

- An open source Linux 2.6 runs on PPC 405.
- Totally free and well maintained by many people around the world.
- Many device drivers and other resources from Internet.
- Powerful network support, suitable for our case.

# Software

With the OS support,

- the hardware and software platform could be universal for multiple uses via different application programs.
- physicists could write their own application programs with high level programming languages, such as C and C++, without the knowledge of circuit design.
- applications could be more portable among different architectures.

# Kernel Compilation

- Getting the kernel sources
  - www.kernel.org
  - www.penguinppc.org
- Configuring the kernel
  - ARCH := ppc
  - Select the features which are to be included in the kernel.
- Building a PowerPC cross-compile toolchain
  - Cross-compiler: powerpc-405-linux-gnu-gcc
- Then, compiling...

# Network File System

- Generating the root file system
  - busybox
- Mounting the root file system via NFS
  - “Diskless” booting and the on-board flash memory space could be saved.
  - The file system could be easily changed through PC operations, without frequently copying files into CF card or flash chips.
  - Convenient especially for in-field file modification.

# System Upgrading

- The OS kernel image and the hardware configuration data are both stored in the flash chips on board.
- After the booting of Linux, the flash memory could be addressed via its device driver. Then the configuration data and kernel image could be upgraded in Linux.
- Reboot the system and hence the updated hardware and OS begin working.
- Operated remotely and need no download cable.

# Current Progress

- Now Linux 2.6 is running on the ML403 platform, with Gbit Enet data transfer, NFS, reserved memory for buffering incoming data, and many other features supported.
- Host (PC) and client (board) transceiving application programs have been realized to transfer packets along the dataflow with UDP protocol successfully.

# Future work

- Measuring the detailed latency and throughput parameters of the network
- Well-designed transceiving programs for Enet data transfer
- Custom logic design for specific algorithm
- Many things to do...

# Thank You!