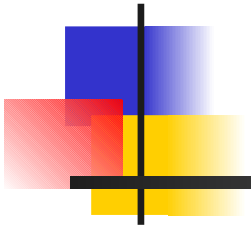
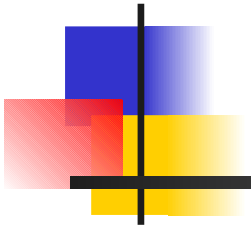


Current Status and Outlook of the DAQ System Design for HADES, BESIII and PANDA

Ming Liu

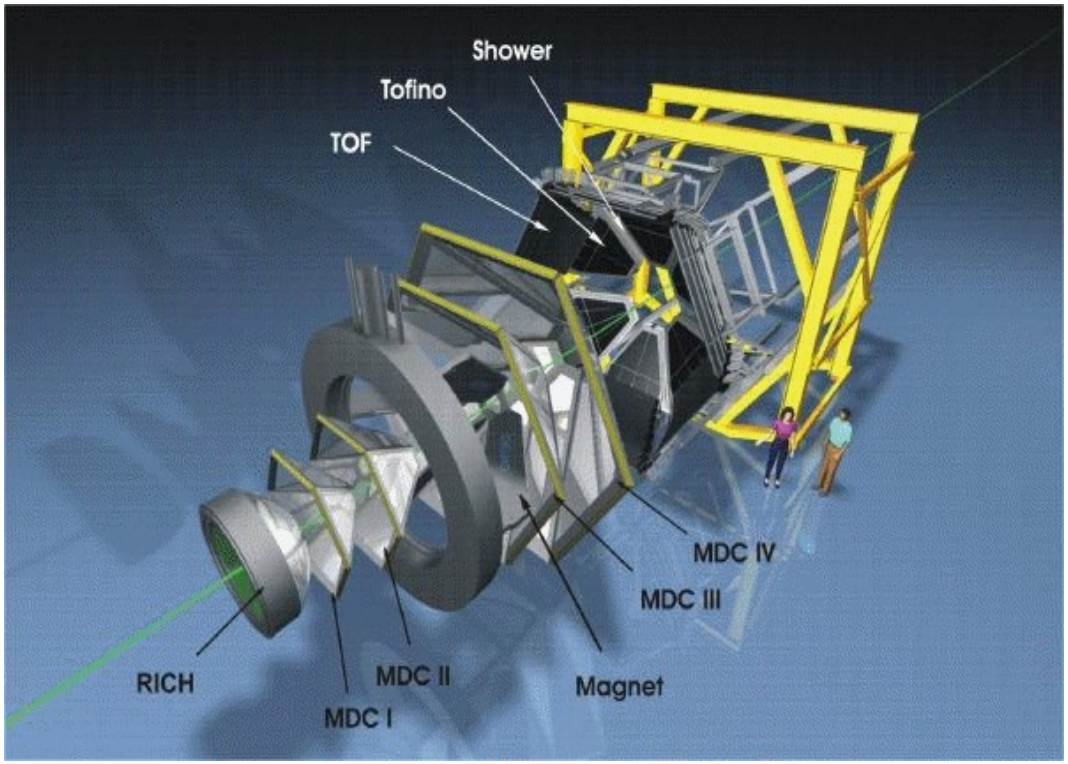


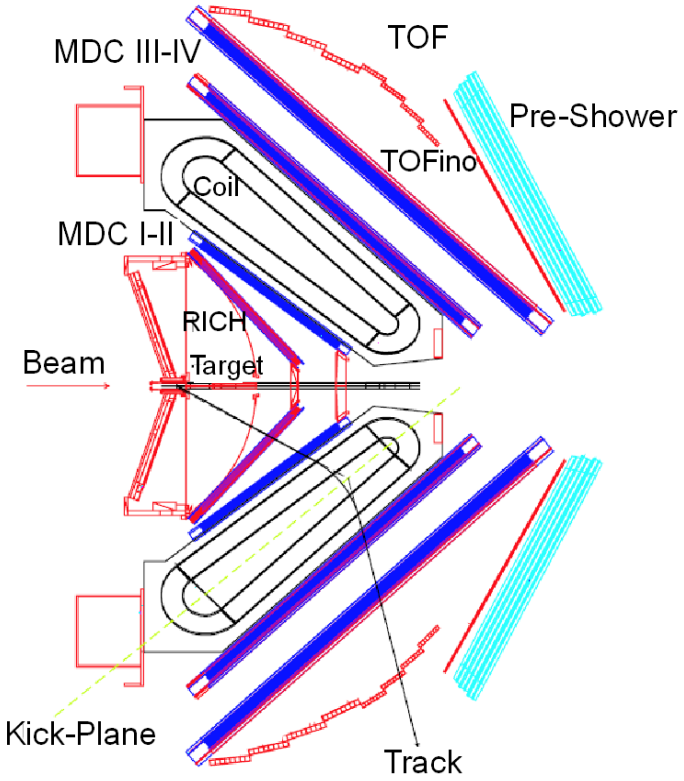
Outline

- Background & Motivation
- Computation Platform for DAQ
  - Network Architecture
  - Compute Node Architecture
- FPGA Node Prototype Design on Xilinx ML40x board
  - HW Design (platform + feature extraction processors)
  - SW Design (embedded Linux + drivers + applications)
- MDC Track Recognition and Reconstruction
- Current Status and Outlook

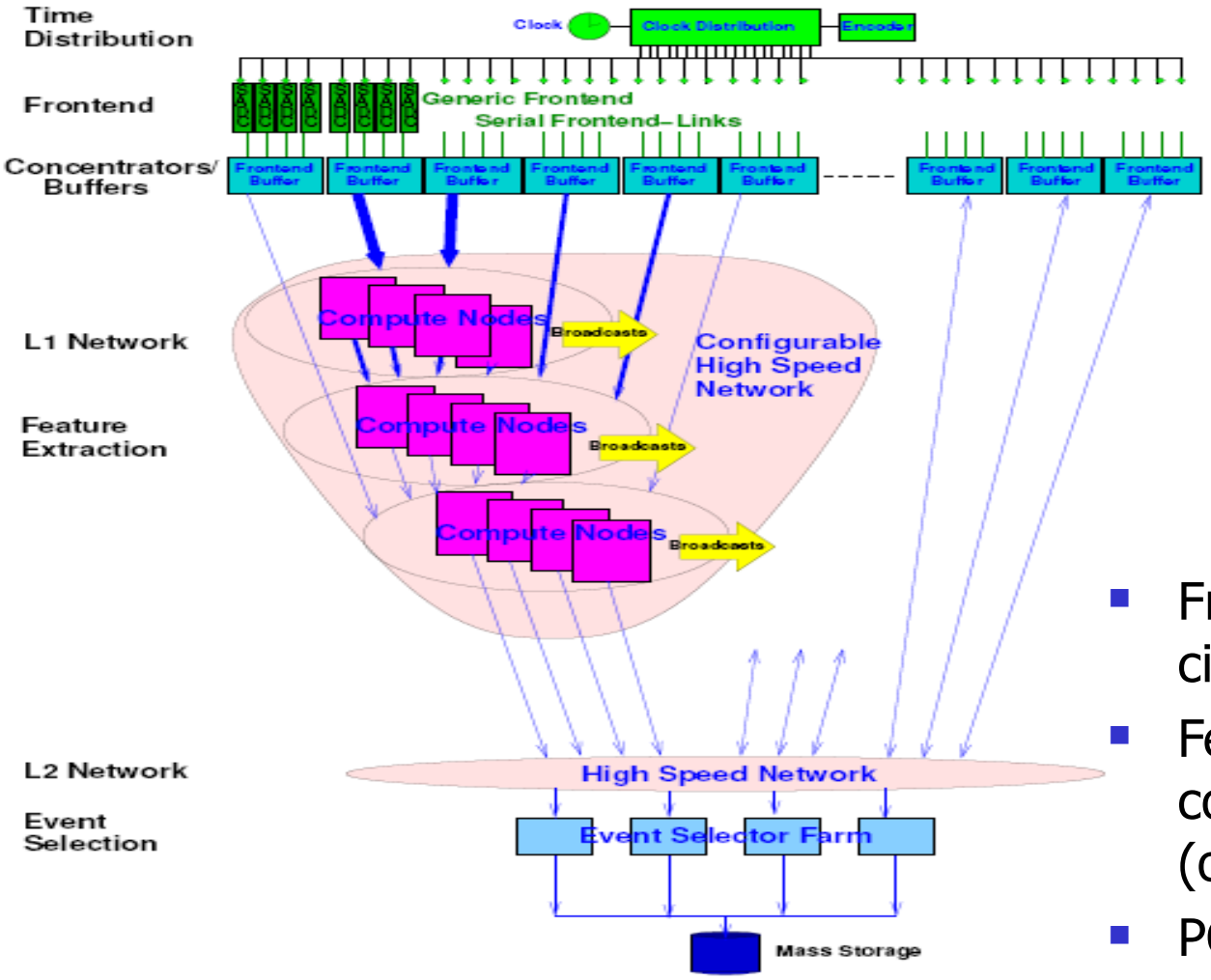
Detector System in Physics Experiments

- Detectors: RICH, MDC, TOF, Shower ……
- Signals generated when particles fly through the detectors

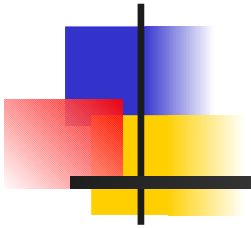




Data Acquisition (DAQ) System



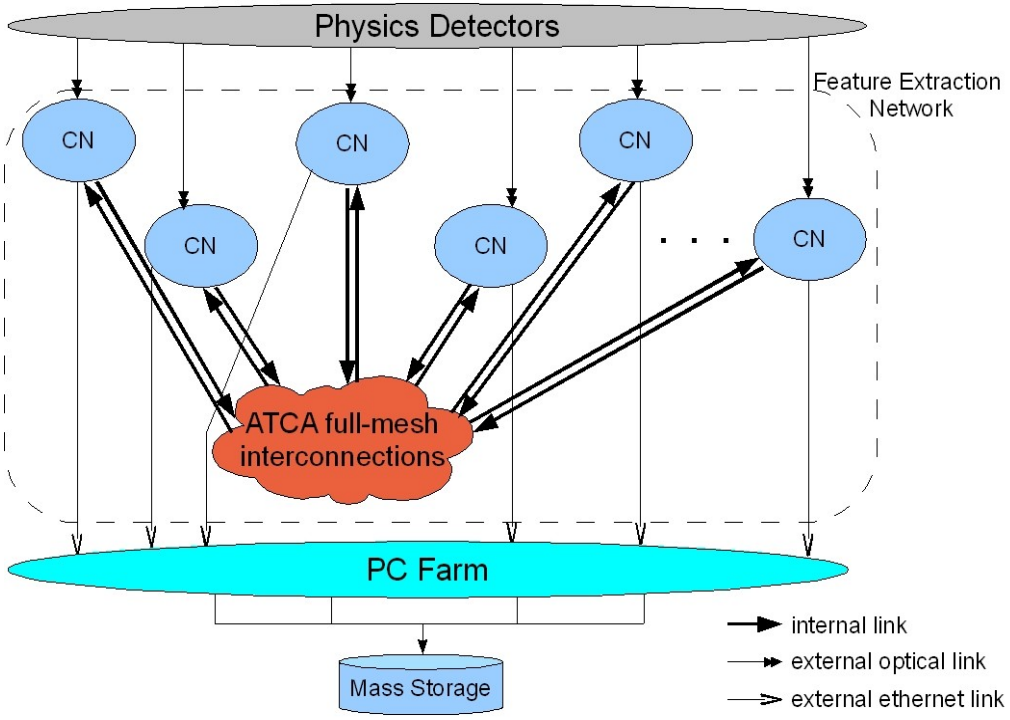
- Front-end electronic circuits
- Feature extraction computation network (our focus)
- PC farm & mass storage



Motivation

- The DAQ system in modern nuclear and particle physics experiments desires:
  - High data processing capability. (reaction rate up to 20MHz, data rate up to 200 GB/s)
  - Various on-line feature extraction algorithms. (Ring recog., MDC tracking, etc.)
  - To be general-purpose for different experiments. (HADES, BESIII, PANDA, etc.)
  - To be highly and remotely reconfigurable.
- Aim:
  - To design the system, based on the platform FPGA technology and HW/SW co-design.

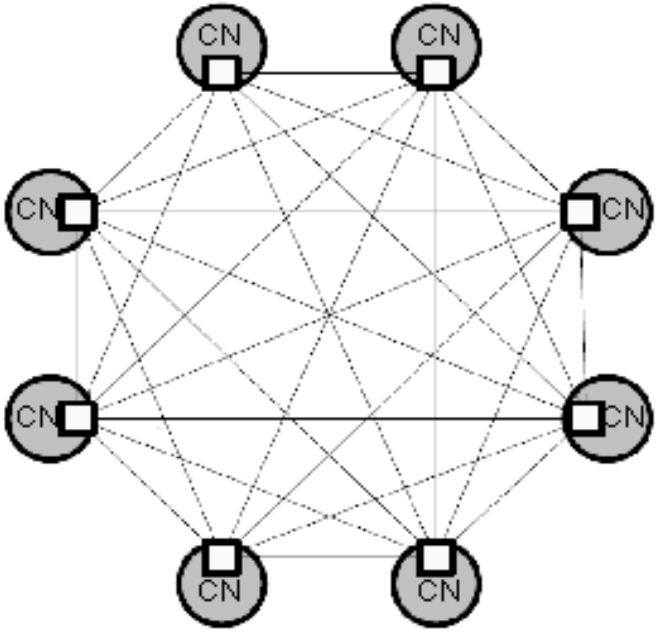
Network Architecture



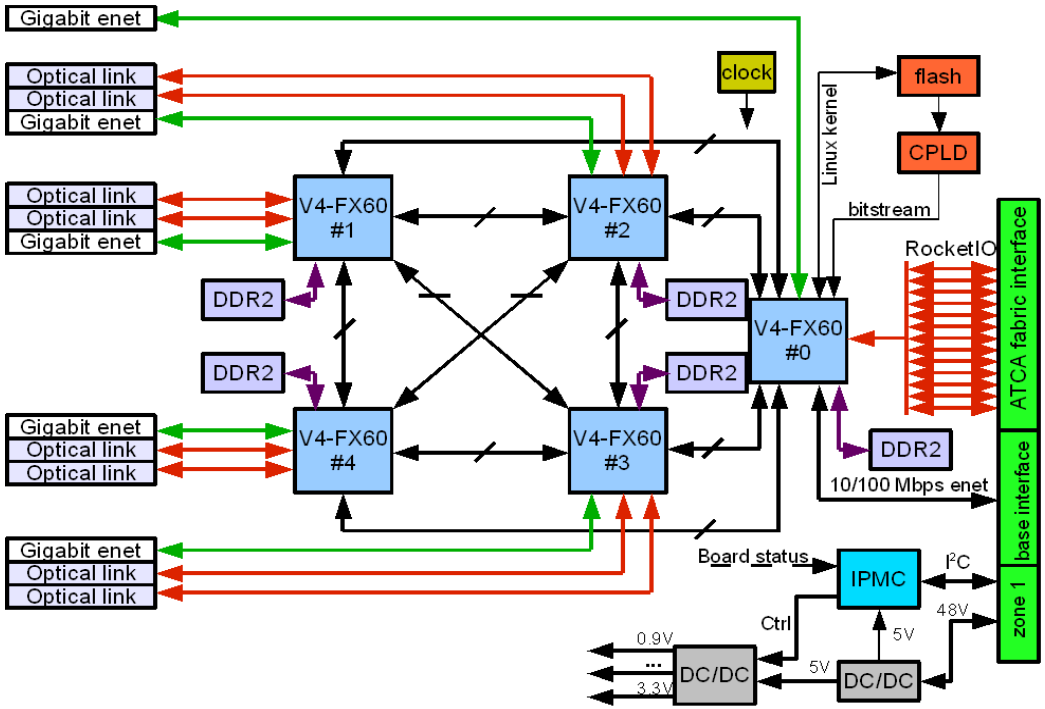
- Feature extractions implemented in the computation network
- Compute node boards internally interconnected by the full-mesh ATCA backplane (communications and correlations among feature extraction computations)
- External interconnections with detectors & PC farm:
  - Optical links
  - Gigabit Ethernet

ATCA Full-mesh Backplane

- Full-mesh backplane network
- High flexibility to correlate results from different algorithms
- High performance

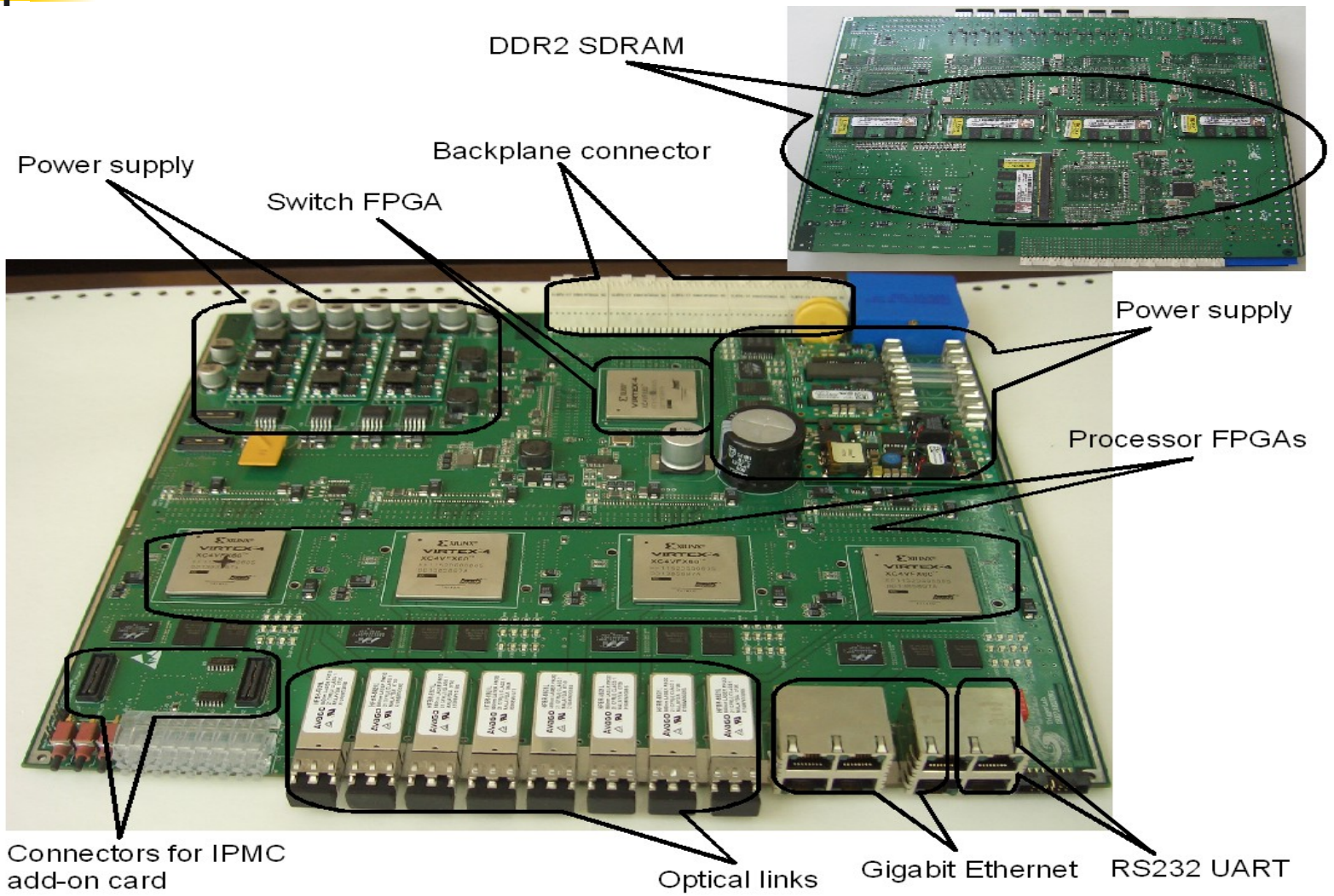


Compute Node (CN) Architecture

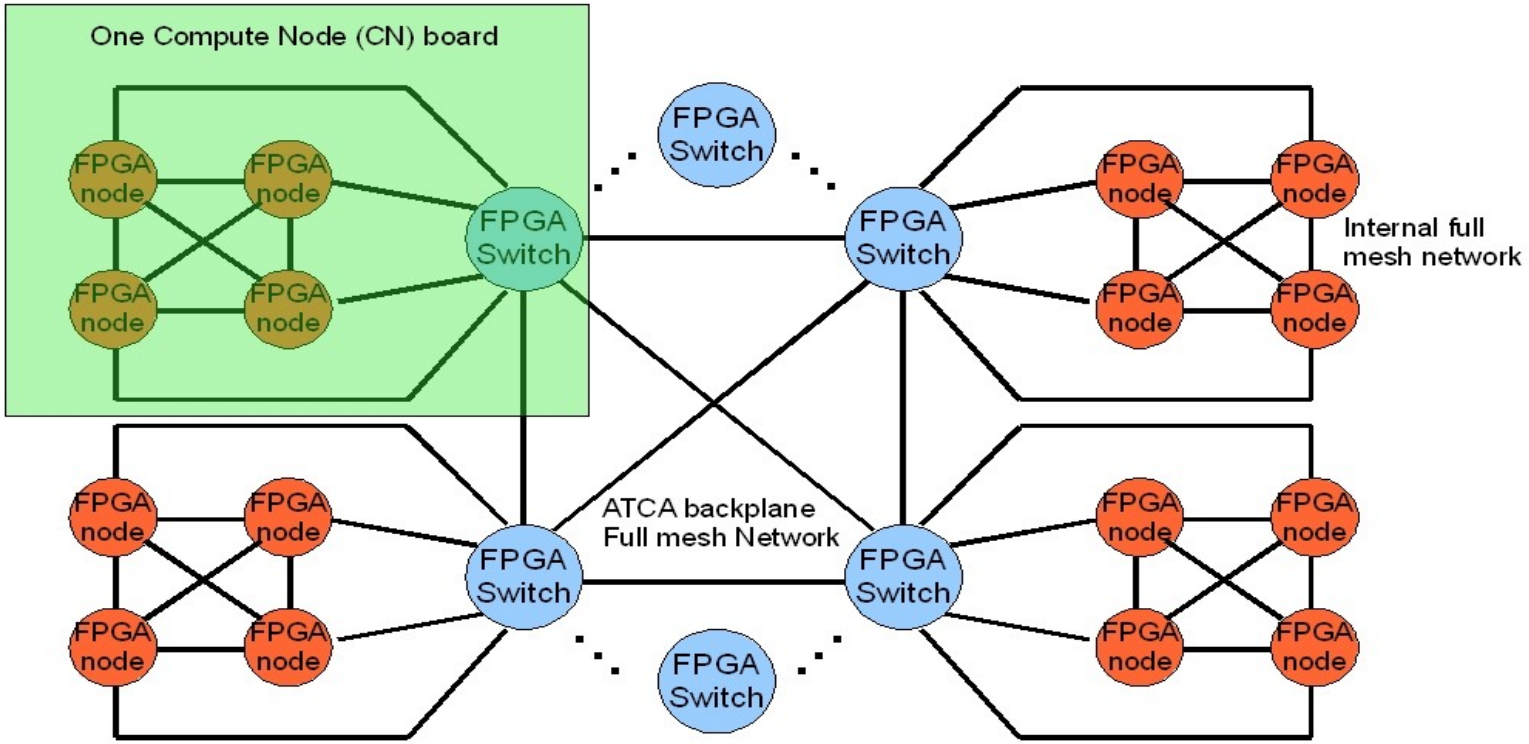


- First version board with 5 Xilinx Virtex-4 FX60 FPGAs
- 4 FPGAs as algo. processors
- 1 FPGA as switch+processor
- Full-mesh communication on-board
- External links:
  - Optical links
  - Gigabit Ethernet

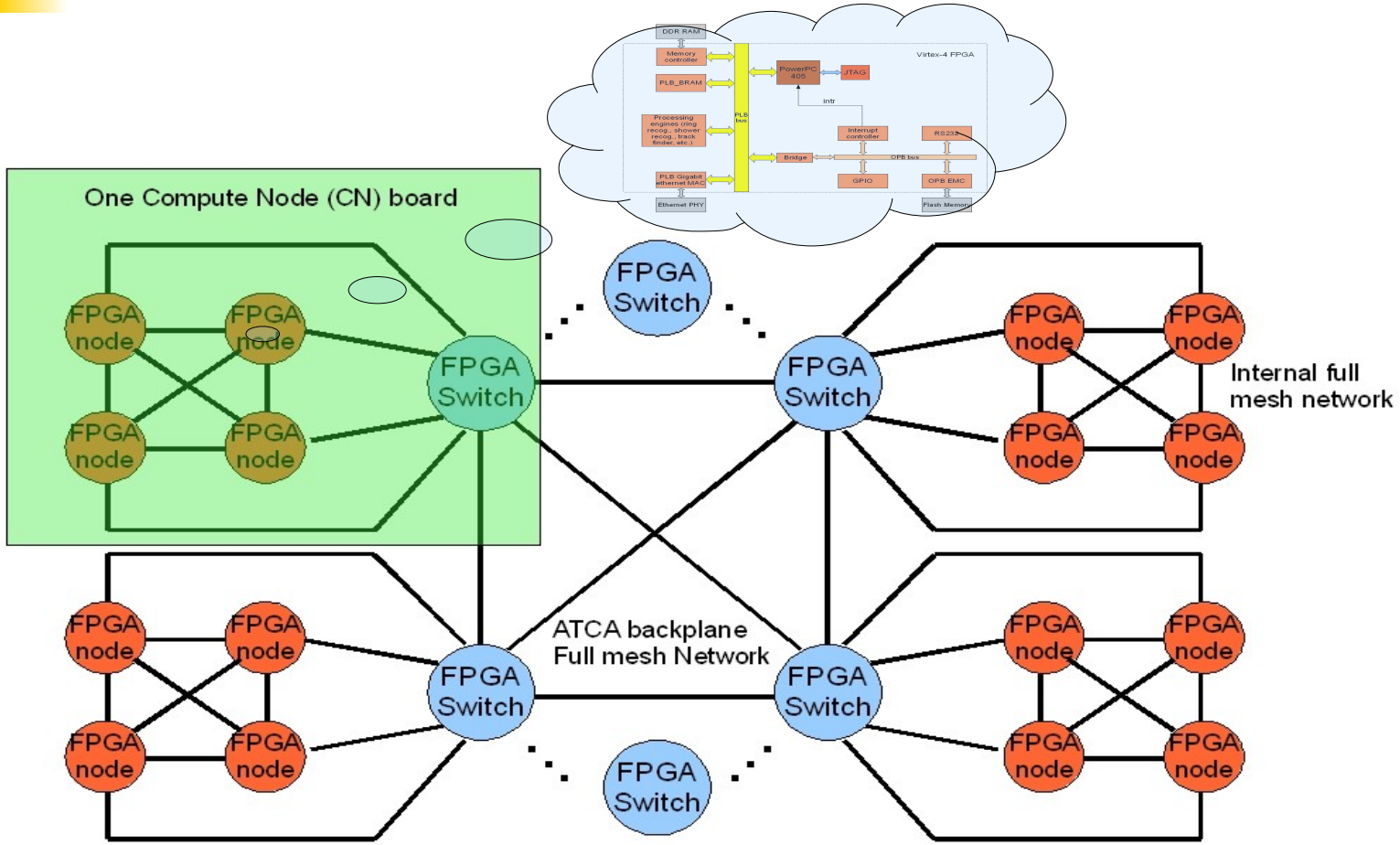
Compute Node PCB

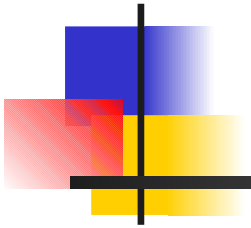


System Topology



System Topology

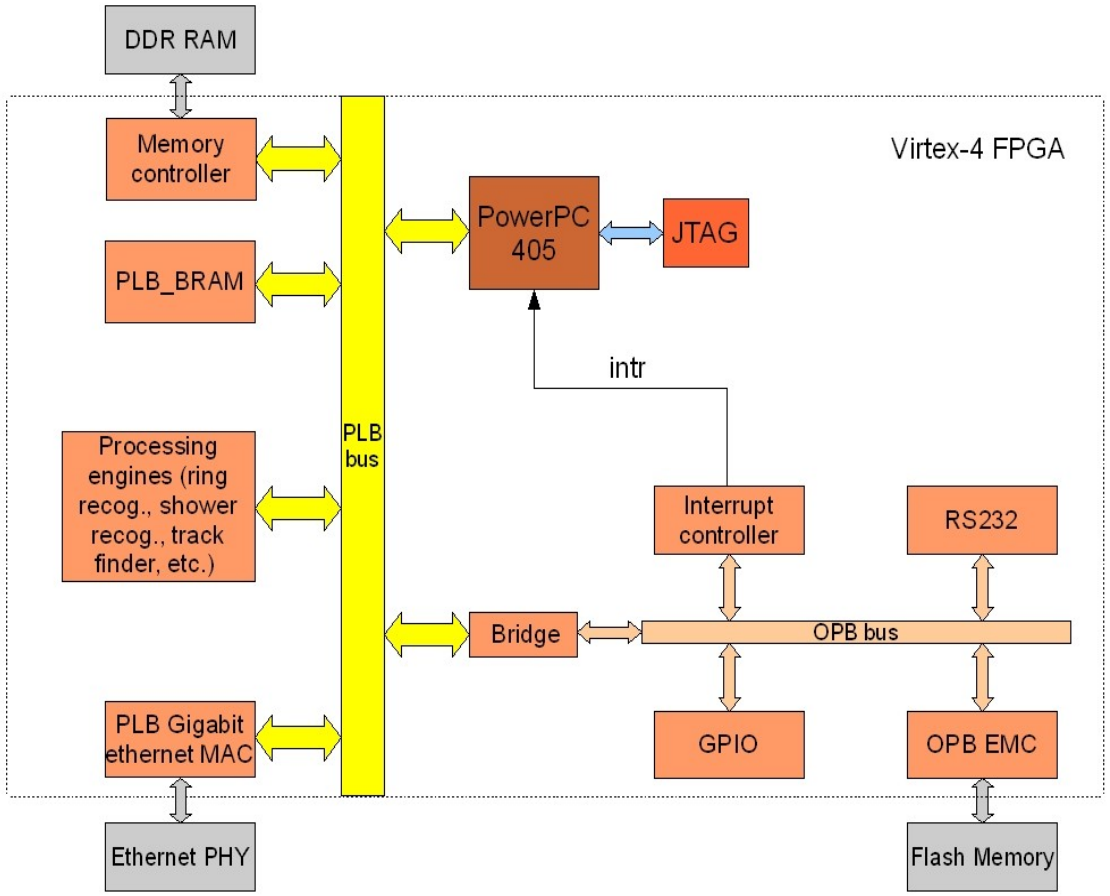




Partitioning Strategy

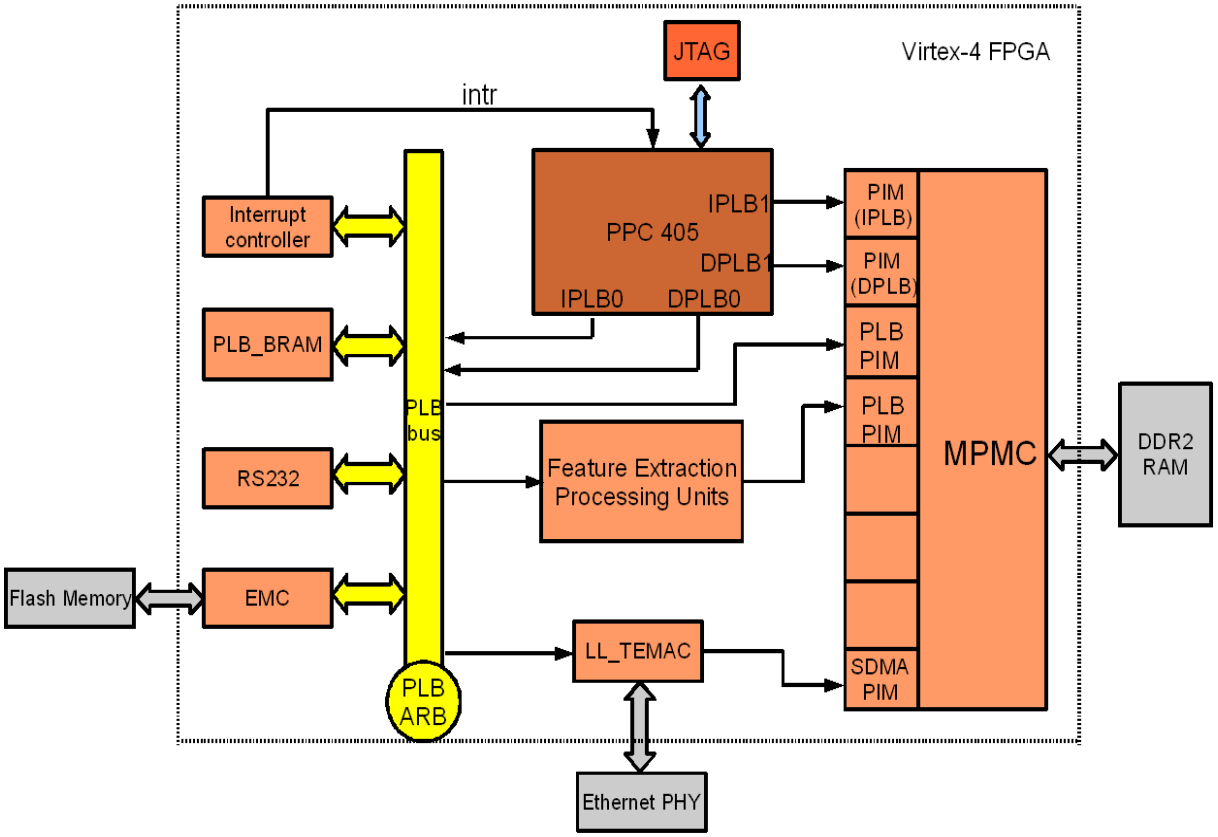
- Computation-intensive algorithms implemented in the FPGA fabric for high performance.
- Parallel and pipelined computation in HW.
- Slow controls in SW (OS + Applications):
  - To remotely upgrade the HW and SW designs.
  - Network test and measurement.
  - To display and modify the experimental parameters.
  - ……
- Soft TCP/IP stack in Linux OS.

Bus-based HW Design (FPGA Node)



- Bus-based platform
  - PLB (fast)
  - OPB (slow)
- PowerPC 405 CPU
- Algo. Processing engines
- Other peripherals:
  - Gigabit Ethernet
  - DDR memory
  - Flash memory
  - RS232
  - ……

LocalLink-based HW Design (FPGA Node)

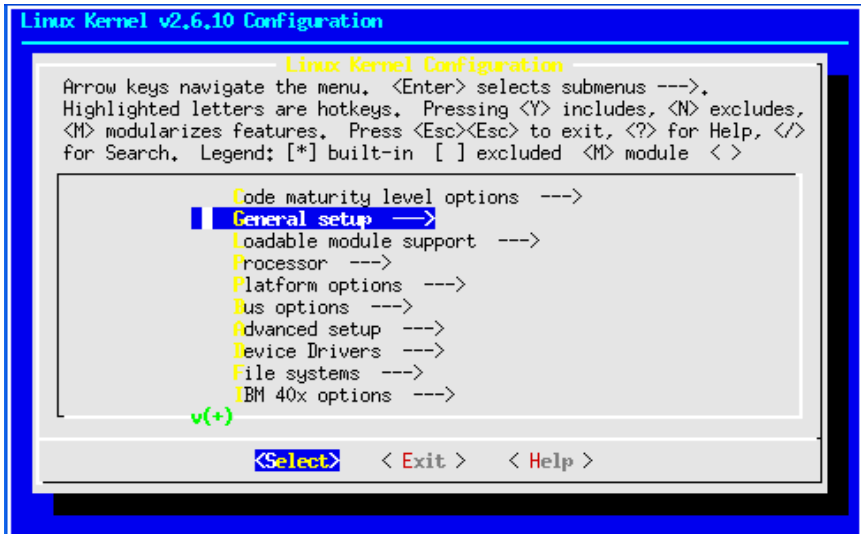


- LocalLink-based platform
- Multi-Port Memory Controller (8 ports)
- Heavy traffic avoided on the PLB bus
- Direct access to the memory from the device
- Large performance improvement expected

SW Design

- Open-source embedded Linux on the PowerPC 405
  - Source code downloaded from internet
  - GNU Cross-compile tool set for PPC405 architecture
  - Communications with Linux fans around the world
- Device drivers:
  - For Ethernet, UART, Flash memory, etc.
  - For the customized processing units
- Applications for slow controls:
  - High level scripts
  - C/C++ programs
  - Java programs on the VM

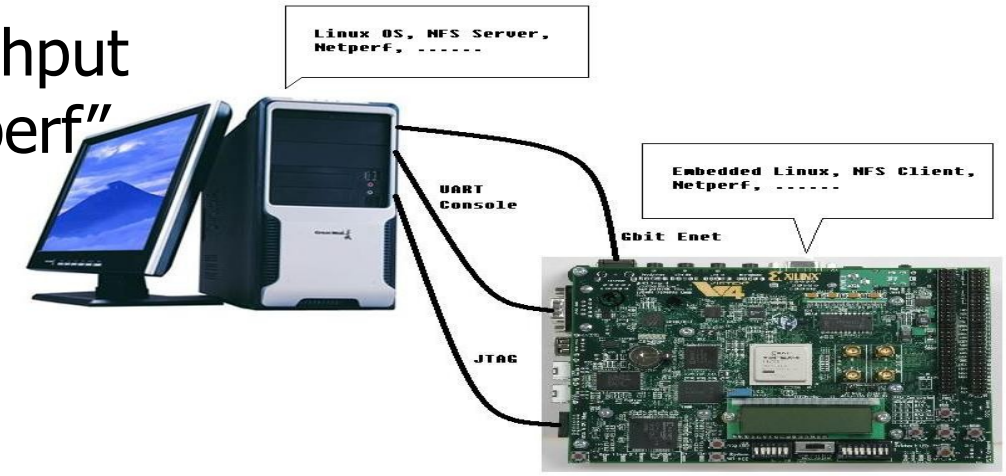
Linux OS Porting



- Kernel configuration
  - Configure useful kernel options
  - Enable necessary device drivers
- Cross-compile with GNU tools
  - Specify the HW adress mappings to the kernel by BSP files
  - Compilation errors have to be fixed
- File system generated by Busybox
  - /bin, /dev, /home, /etc, ... directory hierarchy
  - Scripts for system booting
- The entire OS is free!!! -> No official support from companies

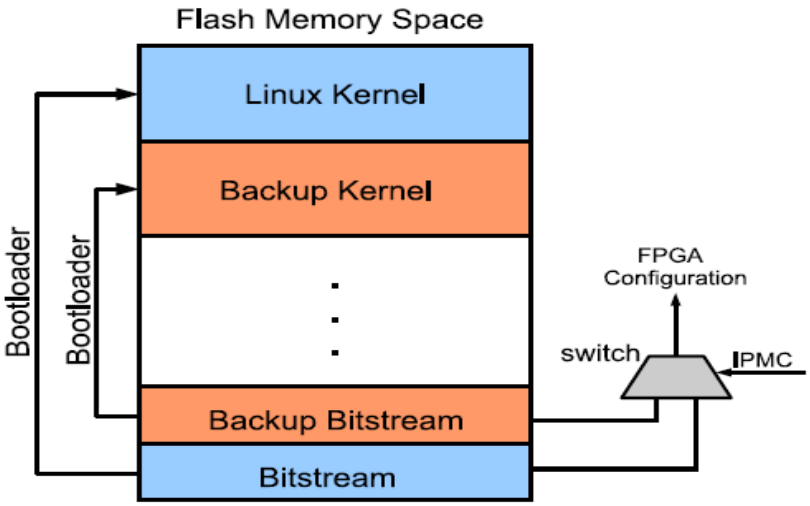
Application Programs on Linux

- With the support of the OS and device drivers, application programs are easily developed to control the system.
  - Only for performance-noncritical control functions, such as system reconfiguration & upgrade, experimental condition displays, etc.
  - Useful for remote configuration and operation when using Ethernet socket programming techniques
  - Mainly in C and cross-compiled on the host PC
- Example: Ethernet throughput measurements with "Netperf"

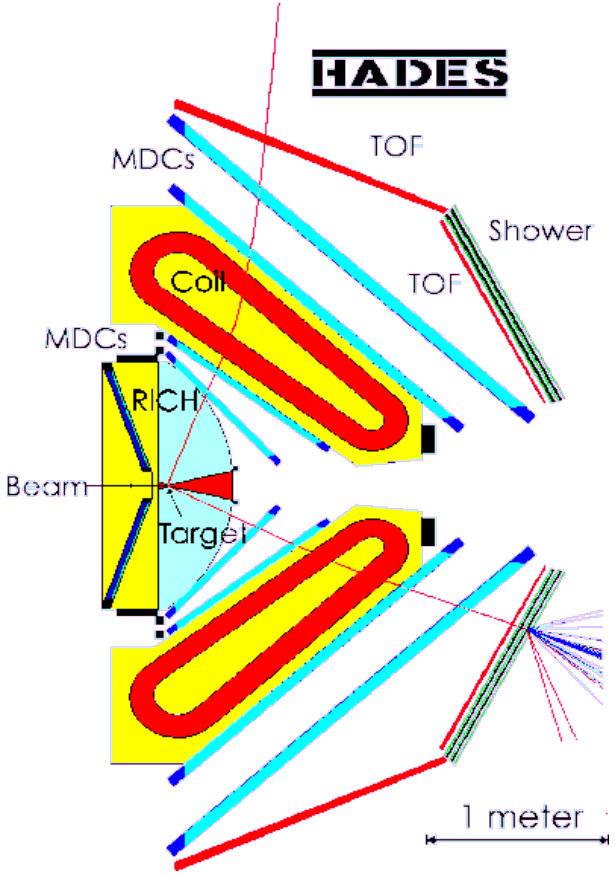


System Configuration Storage and Upgrade

- The OS kernel image and the FPGA bitstream 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.
- Remotely reconfigurable capability
- Backup mechanism to guarantee the system alive

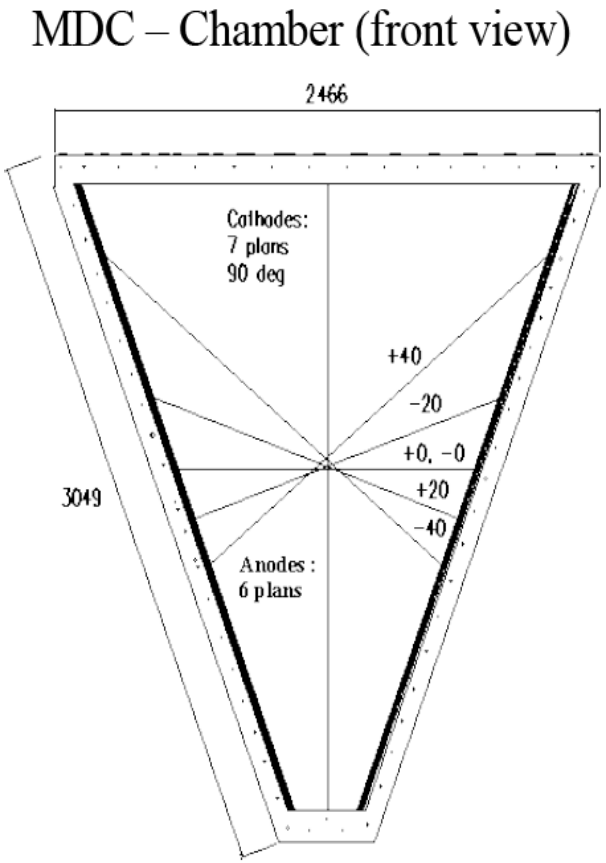


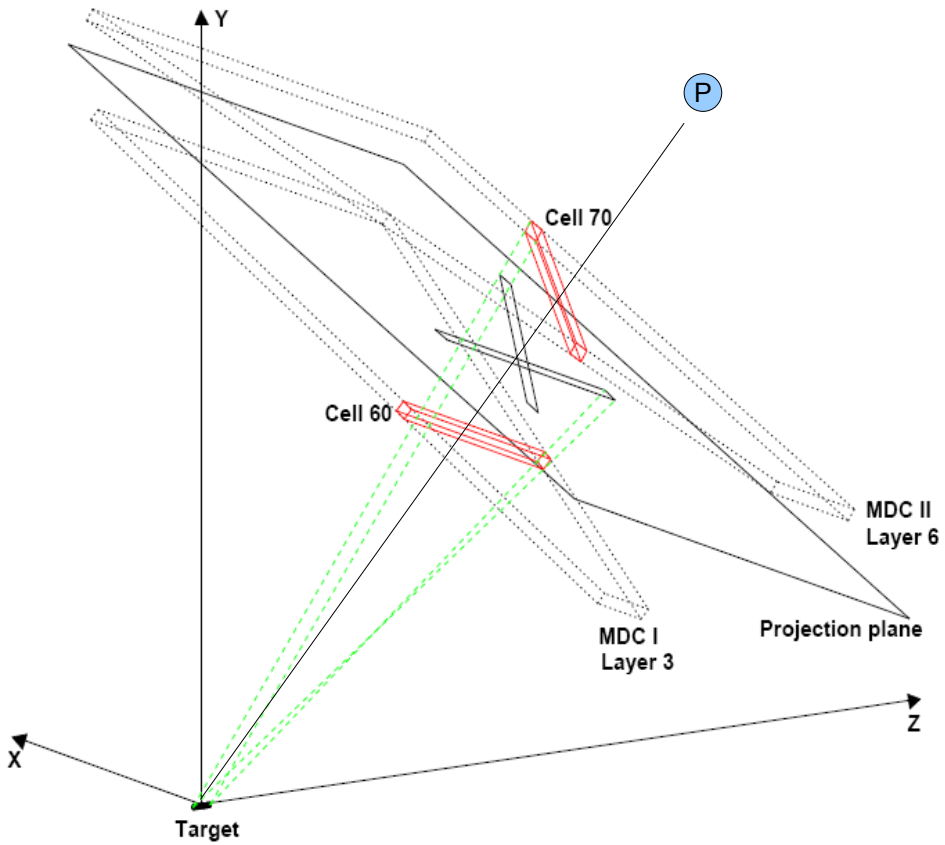
Tracking in MDCs



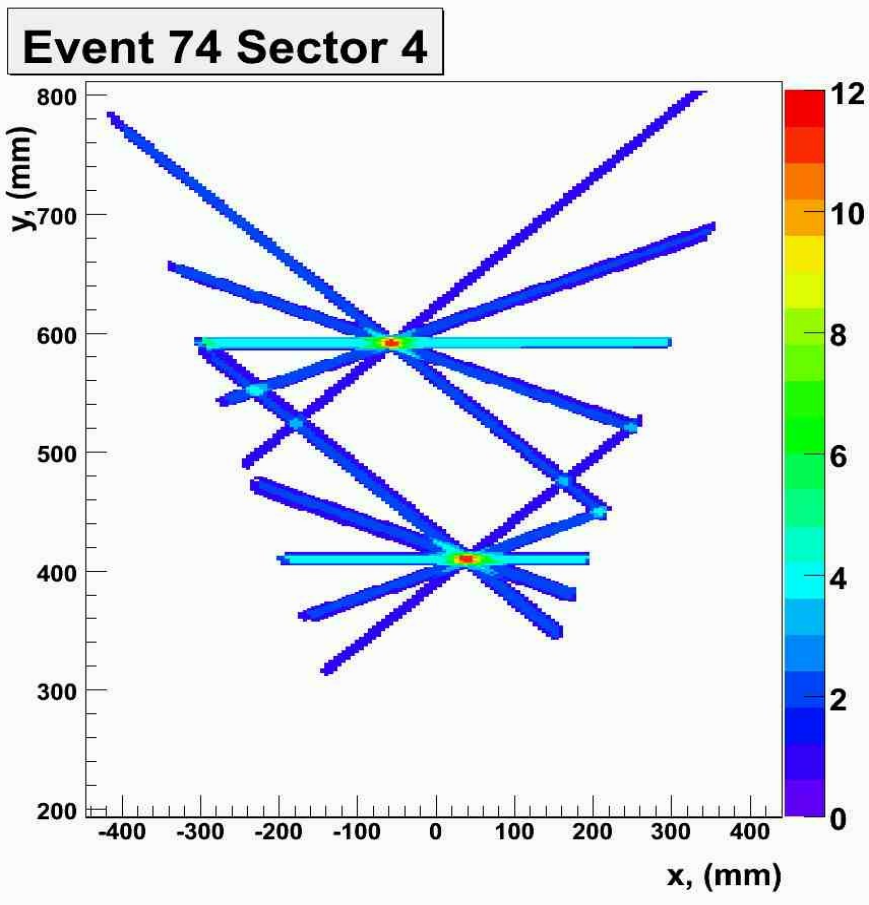
- 4 MDCs
- Particles' tracks bended in the magnetic area
- Straight line tracks from target to MDC II, and from MDC III to MDC IV
- Inner and outer tracks pointing to RICH and TOF respectively and helping them to find patterns

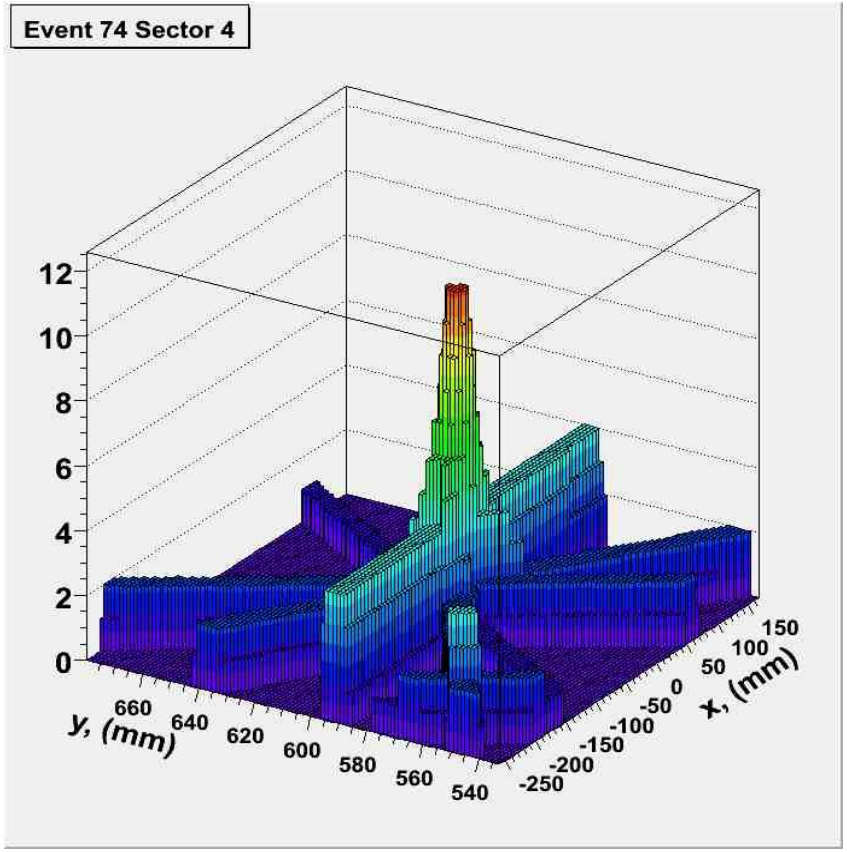
Tracking in MDCs





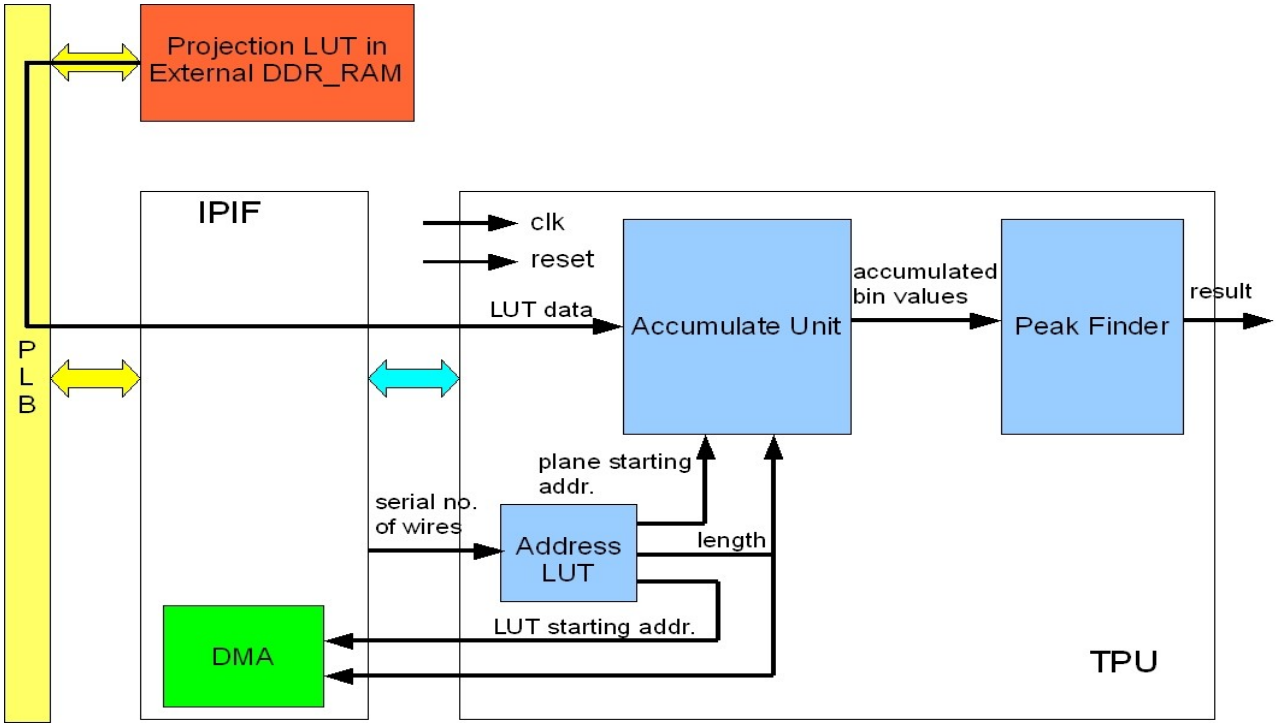
Tracking in MDCs





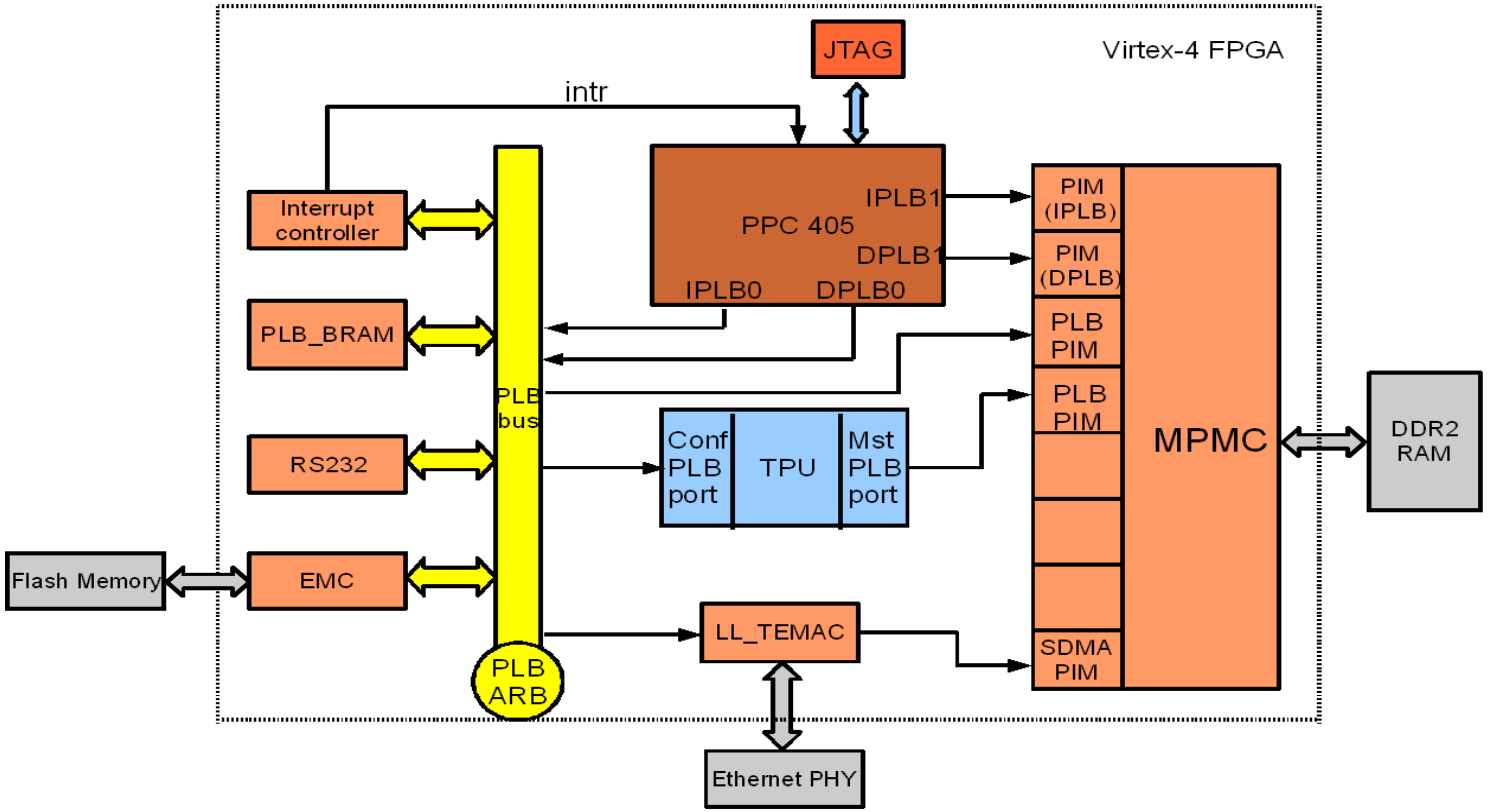
Tracking Processing Unit (TPU) Design

Current bus-based design:



Tracking Processing Unit (TPU) Design

Future LocalLink-based design:



Synthesis Results

Resources	TPU	compute node platform	PLB-IPIF	system with TPU (sum)
4-input LUTs	5175 out of 50560 (10.2%)	8531 out of 50560 (16.9%)	2900 out of 50560 (5.7%)	16606 out of 50560 (32.8%)
Slice Flip-Flops	1715 out of 50560 (3.4%)	5724 out of 50560 (11.3%)	1640 out of 50560 (3.2%)	9079 out of 50560 (18.0%)
Block RAMs	41 out of 232 (17.7%)	18 out of 232 (7.8%)	0	59 out of 232 (25.4%)
DSP Slices	0	8 out of 128 (6.3%)	0	8 out of 128 (6.3%)

Table 1. Resource consumption

- Resource utilization is acceptable for Virtex4 FX60.
- Timing limitation: 125 Mhz.
- We choose 100 Mhz, matching the speed of PLB.

Performance Measurements

- A C program running on the PC as the software reference
- Measurement setup: 30 fired wires/sub-event, 5.7 Kbits LUT/wire in average (1,510,256 bytes/2110 wires)
- **SW performance** = 0.82K sub-events/s (10K sub-events/12.2 s)
- When DMA_done interrupt used, **HW performance** = 0.83K sub-events/s
- When DMA_done polling used, **HW performance** = 4.5K sub-events/s (5.5 times speedup)
- In theory, speedup of around 20~30 is expected according to the cycle-accurate simulation
- Some work should be migrated from the software to pure hardware for higher speedup. (e.g. The DMA_done interrupt handler in the driver is cycle-consuming.) Optimization needed!!!

Current Design Status & Outlook

Current Status:

- First version CN PCB produced and under the test
- Inner tracking system developed and being optimized
- Other modules being developed
  - Ring recognition
  - TOF processing
  - Event building
  - Intelligent Platform Management Interface Controller (IPMC)
- Inner tracking system developed and being optimized

Outlook:

- All algorithm implementations
- Algorithm partition and distribution for parallel processing
- A complete running ATCA system

Thanks for your comments!