

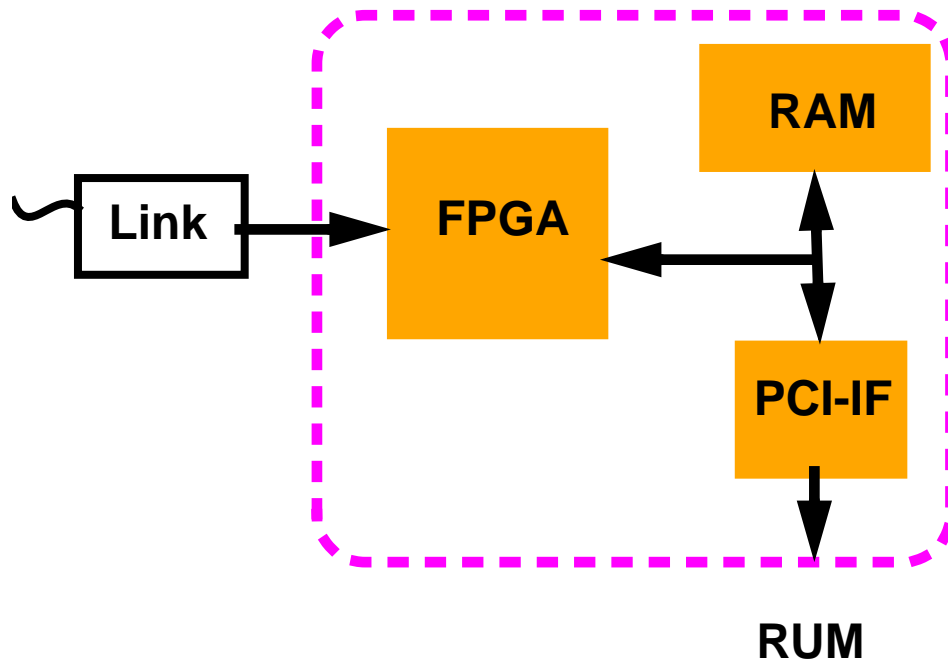
# Status of the Rui-Prototype

- functionality and limitations
- status
- interfaces
- software
- integration plan
- alternatives

# Functionality and Status

- **Functionality**

- receive data from DAQ-Link
- buffer in on board Memory
- transfer via DMA to RUM
- do some consistency checks on data ?



- **Limitations**

- only 32 bit
- low performance ( << 100Mb/sec ! )
- Cannot handle arbitrary event sizes (there must be an upper limit)
- No “real time flow control”
- No checks on data

- **Status**

- implementation in progress
- principle of data-movements involved have been tested.
- main components of logic exist and have been simulated --> need to be integrated and adapted to current Link Design.
- no documentation yet

# Interfaces and Integration Plans

- **Interfaces**

- to software RUM: well defined
- to hardware RUM defined (needs to be confirmed by D. and C.)
- to DAQ-Link: needs to be clarified (Lucien, Attila, Christoph)

- **Software used**

- During hardware development: One-way hacks
- Under construction: OO library to be

integrated in XDAQ

- **Integration + Plans**

- 1) 2 months

- hardware implementation (Christoph)
- OO-lib (Akos?)

- 2) three weeks

Integrate in XDAQ with software RUM (Christoph, Akos + with XDAQ hotline support)

- 3) Tests of the system Rui / Software RUM could start in August

# Alternatives...

- Why, What did we learn?
  - Performance of the generic II based RUI will be limited: below 100Mbytes/sec
  - Using a local RAM on the RUI board kills performance
  - Relying on a large RAM and/or involved algorithms for treating the data in the RUI excludes commercial solutions
  - DAQLink is difficult to operate
- Alternatives:
  - A RUI card without big buffer memory  
-> RUM buffers events and absorb RCN latency
  - LVDS-Link of Dominique:  
similar structure  
High Performance ( 400-800MB/sec)  
available soon
  - commercial link (Myrinet) plus software  
-> Link card in PC which plays the role of RUI and RUM