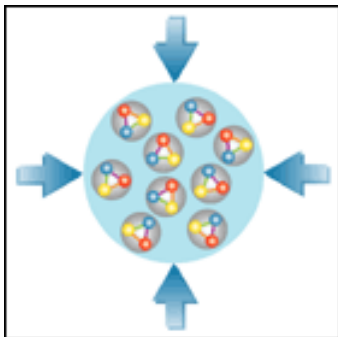

CBM readout controller – ROC

Software development and integration into DABC



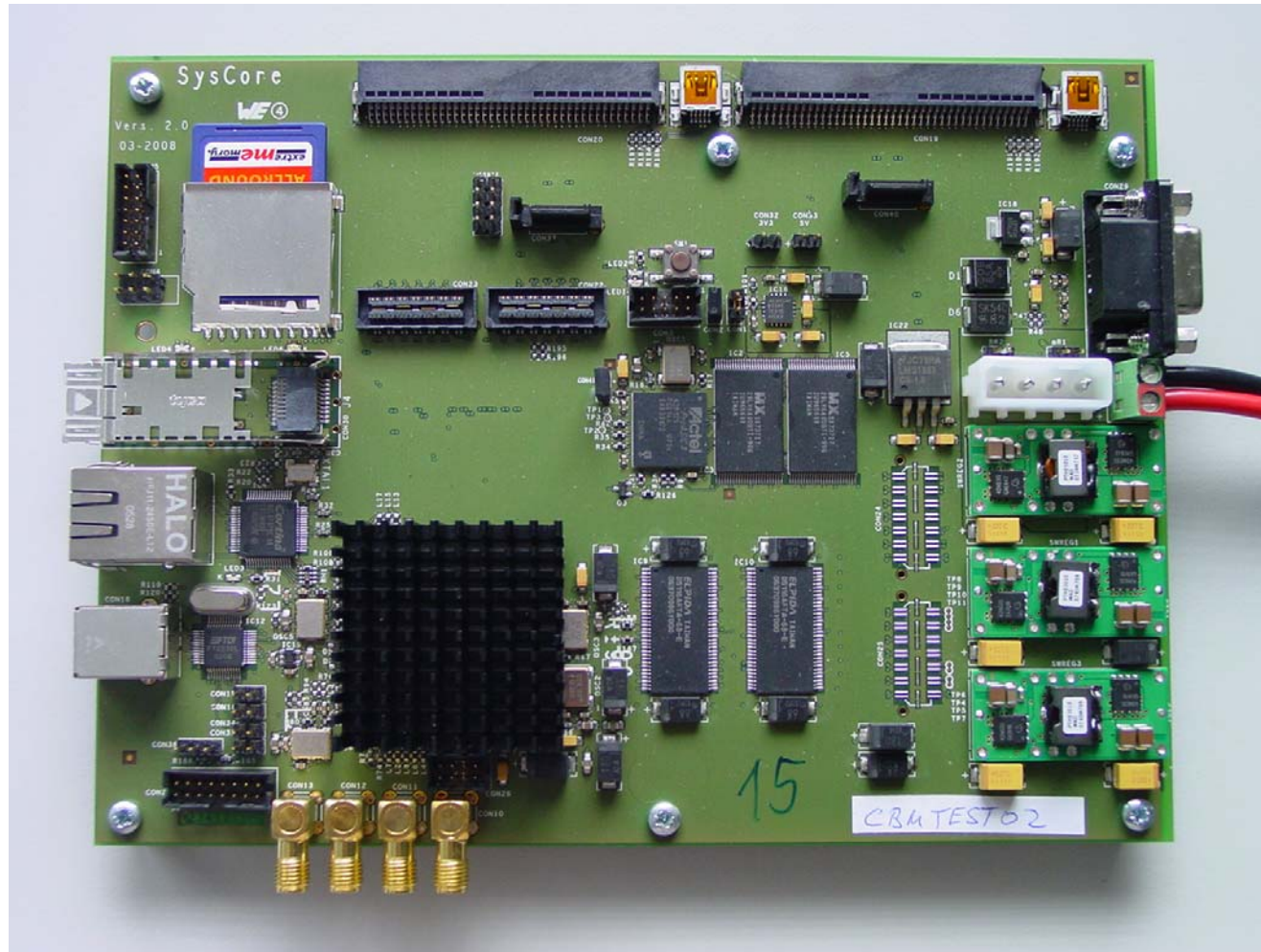
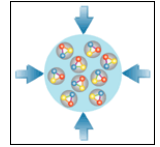
S. Linev, GSI, Darmstadt

EE meeting

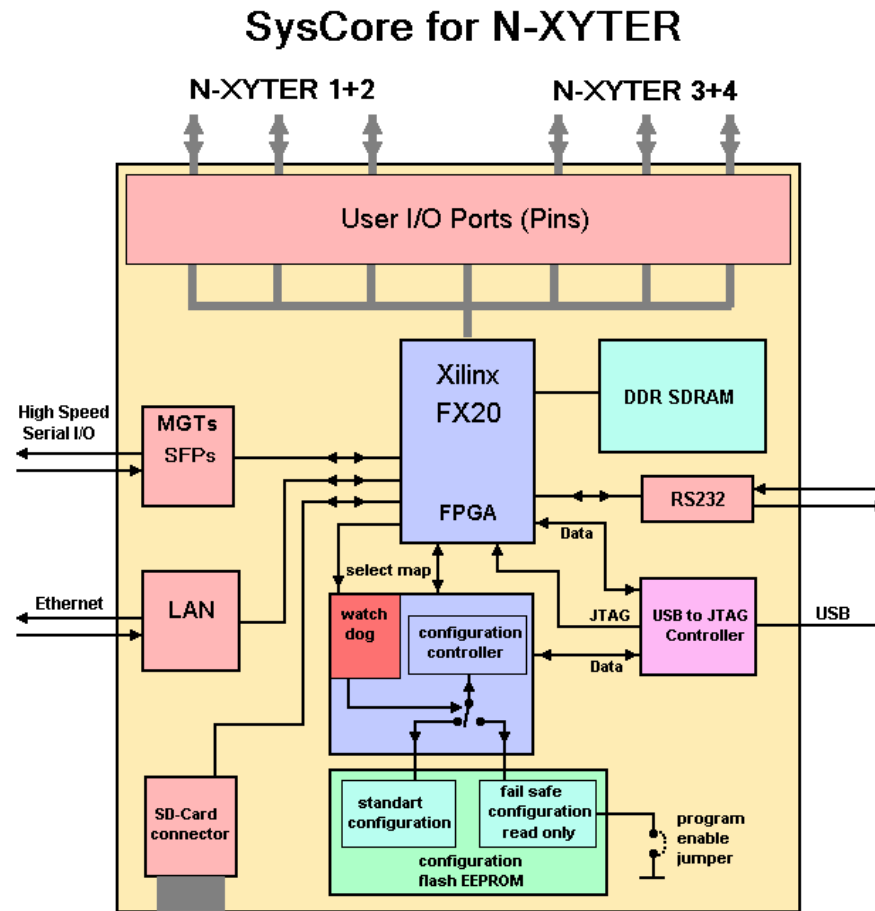
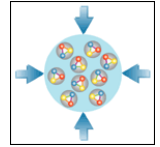
8 December 2008



Readout controller (ROC)



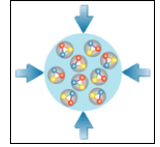
ROC block diagram



dg/kip 2008.02.25 ver.1.04

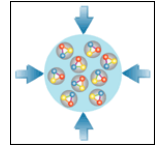
Info: <http://cbm-wiki.gsi.de/cgi-bin/view/NXYTER/SysCoreV2>

First impression



- First version of ROC software package (KNUT) was available in August 2008
- Had a lot of problems:
 - no any kind of blocking readout -> only polling mode was possible -> 100% CPU for nothing
 - UDP as transport → no any kind of retransmission implemented → suspends of data taking after several minutes
 - only half-duplex mode worked in GSI → 0.5% packets lost
 - unreliable controlling interface
- Has to be ready for September 2008 beam time

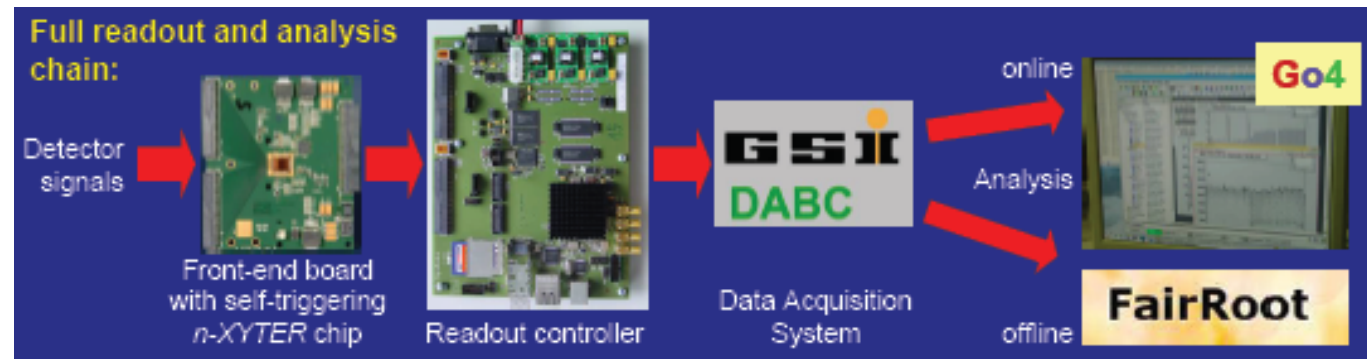
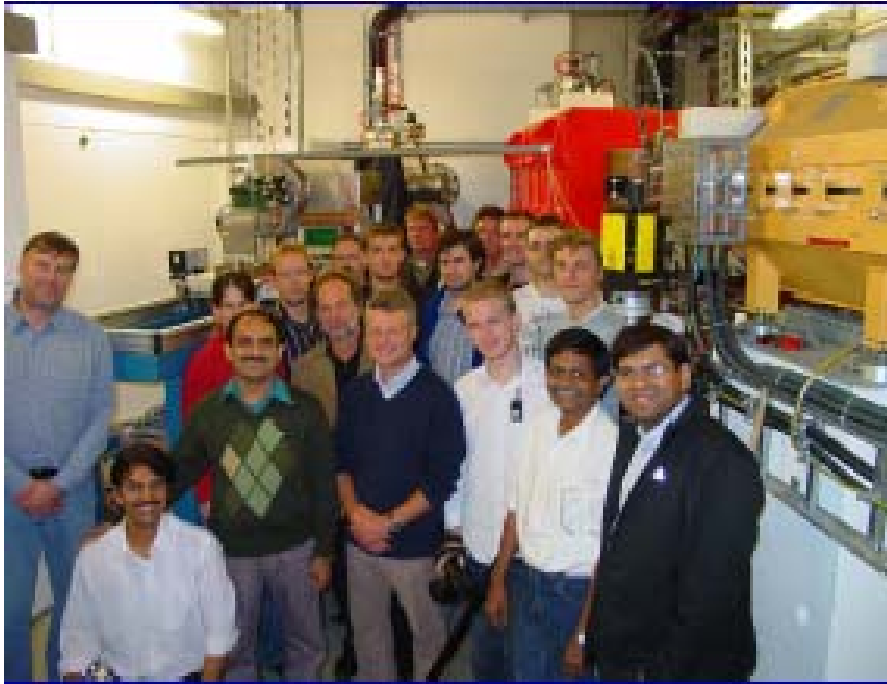
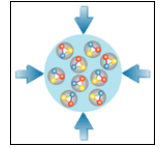
Development before beamtime



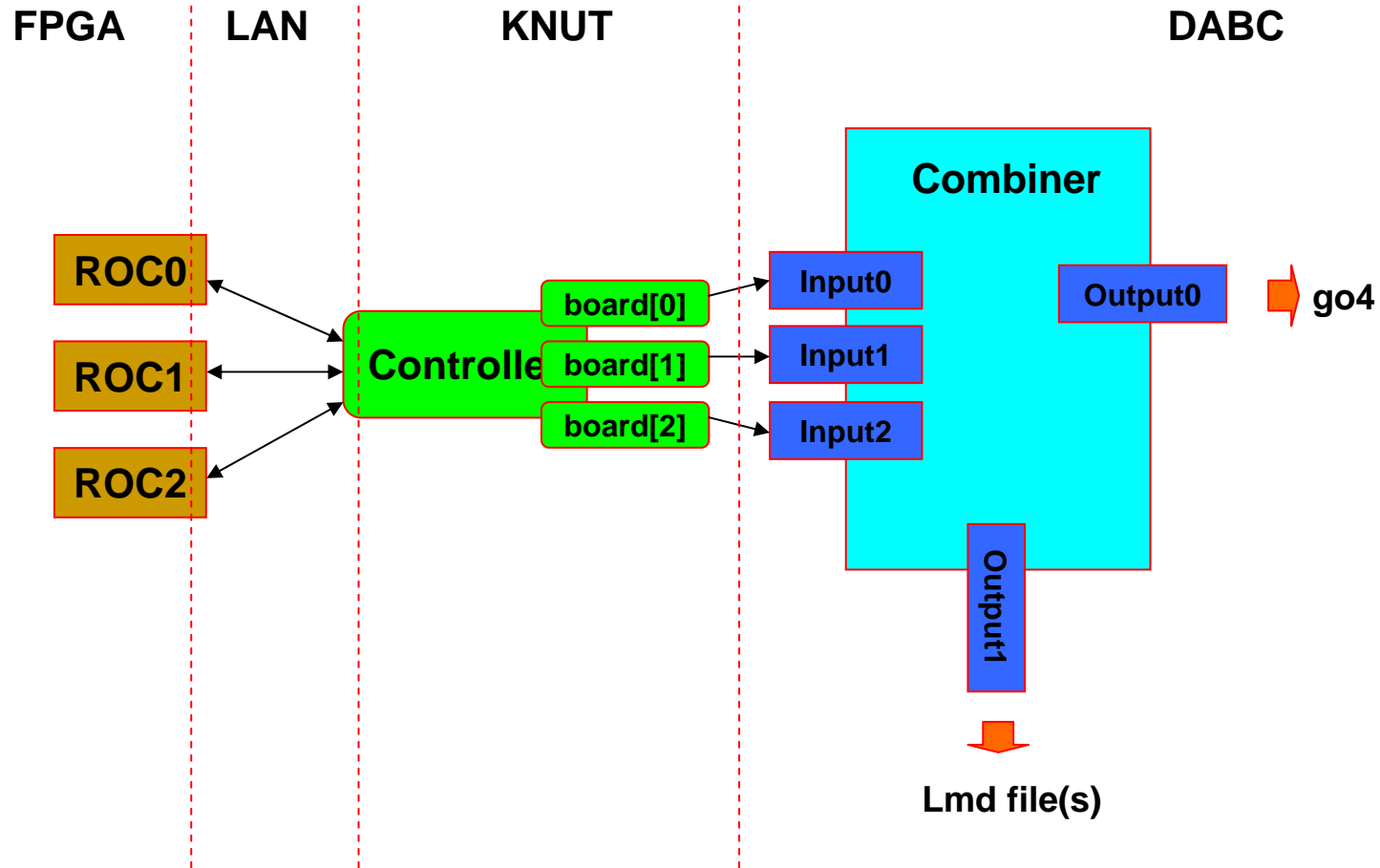
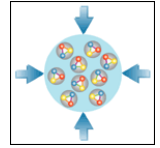
- Support in DABC
 - device and transport implementation
 - readout application with combiner and calibration module
 - storage of data in new lmd format
 - small go4 monitor program, connected to DABC via stream server

- Our changes in KNUT
 - Reliable control interface
 - Blocking data read interface
 - Data retransmissions – recover of lost data packets
 - Via timeout detect and recover lost of data requests

Test beamtime September 2008

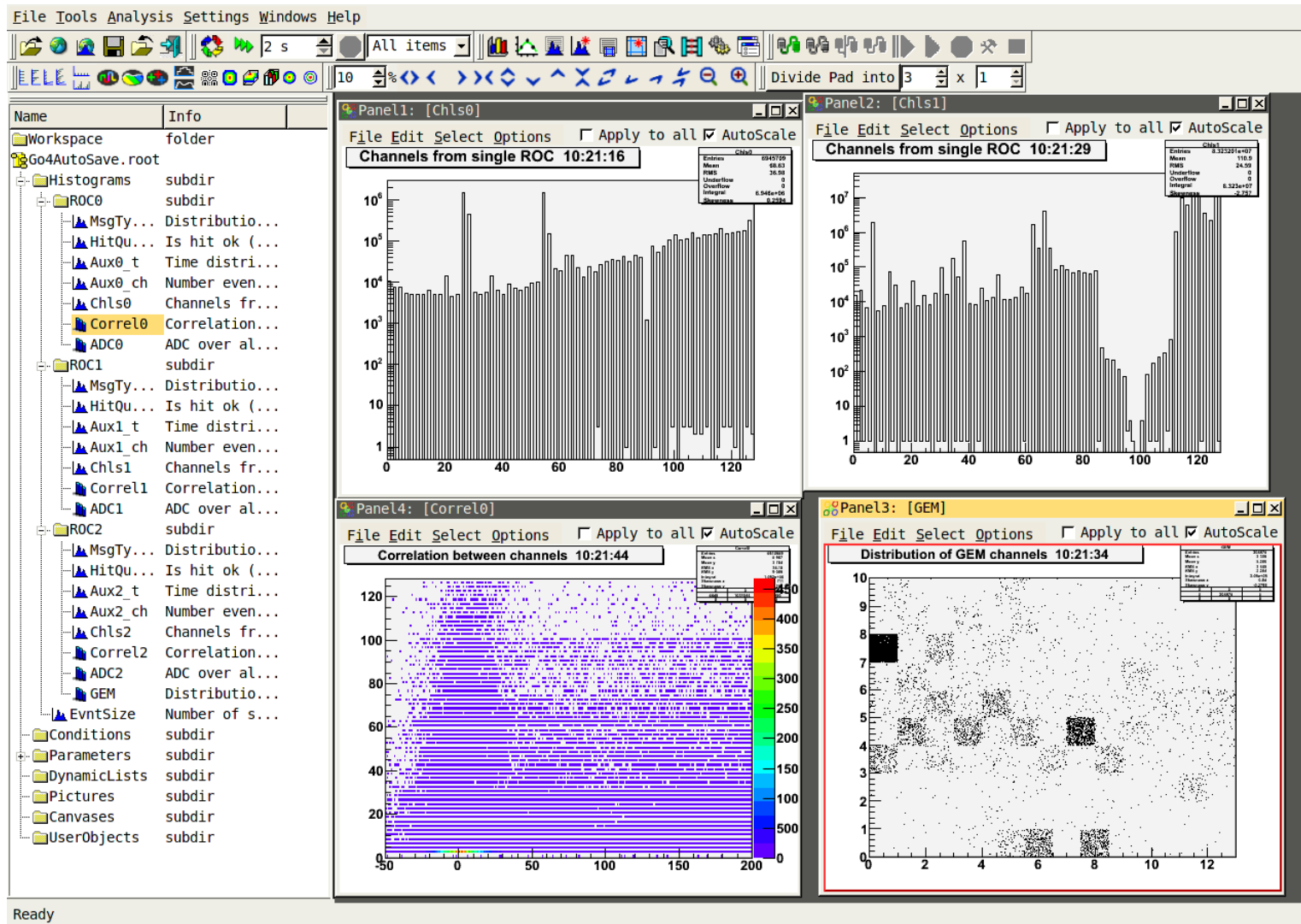
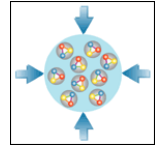


Software in beamtime

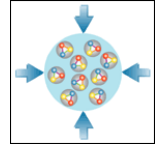


Complete software collection: <https://subversion.gsi.de/cbm/ROC/trunk>

Online go4 monitor



Beam time observations

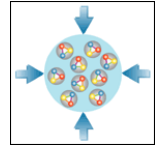


- First run of complete DAQ with 3 ROCs and wired sync signals happened only 30 minute before beam starts – and it was working!
- Overall 24 Gb of raw data was taken
- Sustained data rate ~1MB/s per ROC

- A lot of new problems were discovered:
 - corrupted NXYTER messages, NOPs
 - lost of epoch markers
 - unsynchronized data drop on ROCs, when rate > 1 MB/s

 - no manpower in Heidelberg to fix these and other problems

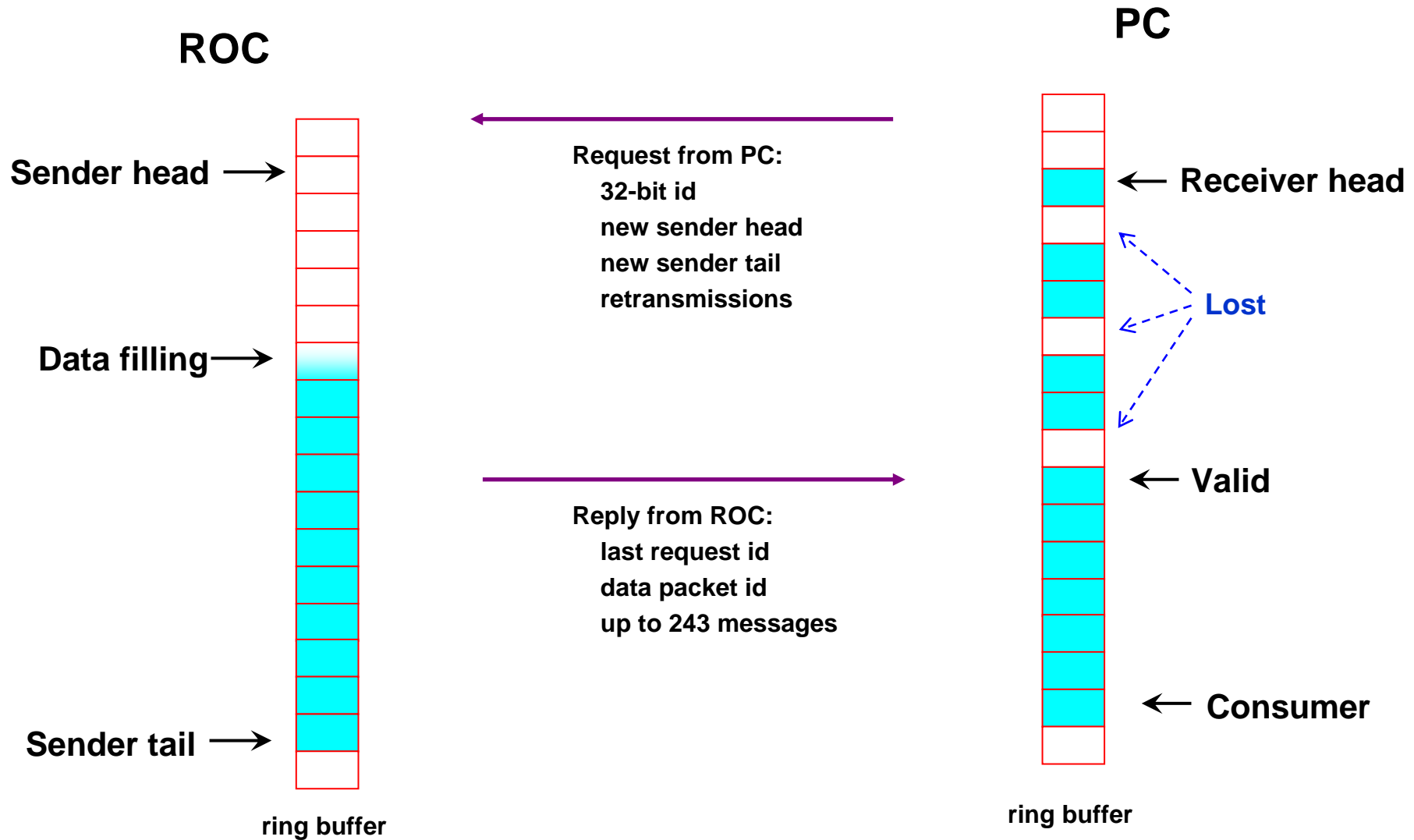
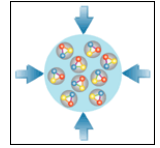
Full redesign of PowerPC code



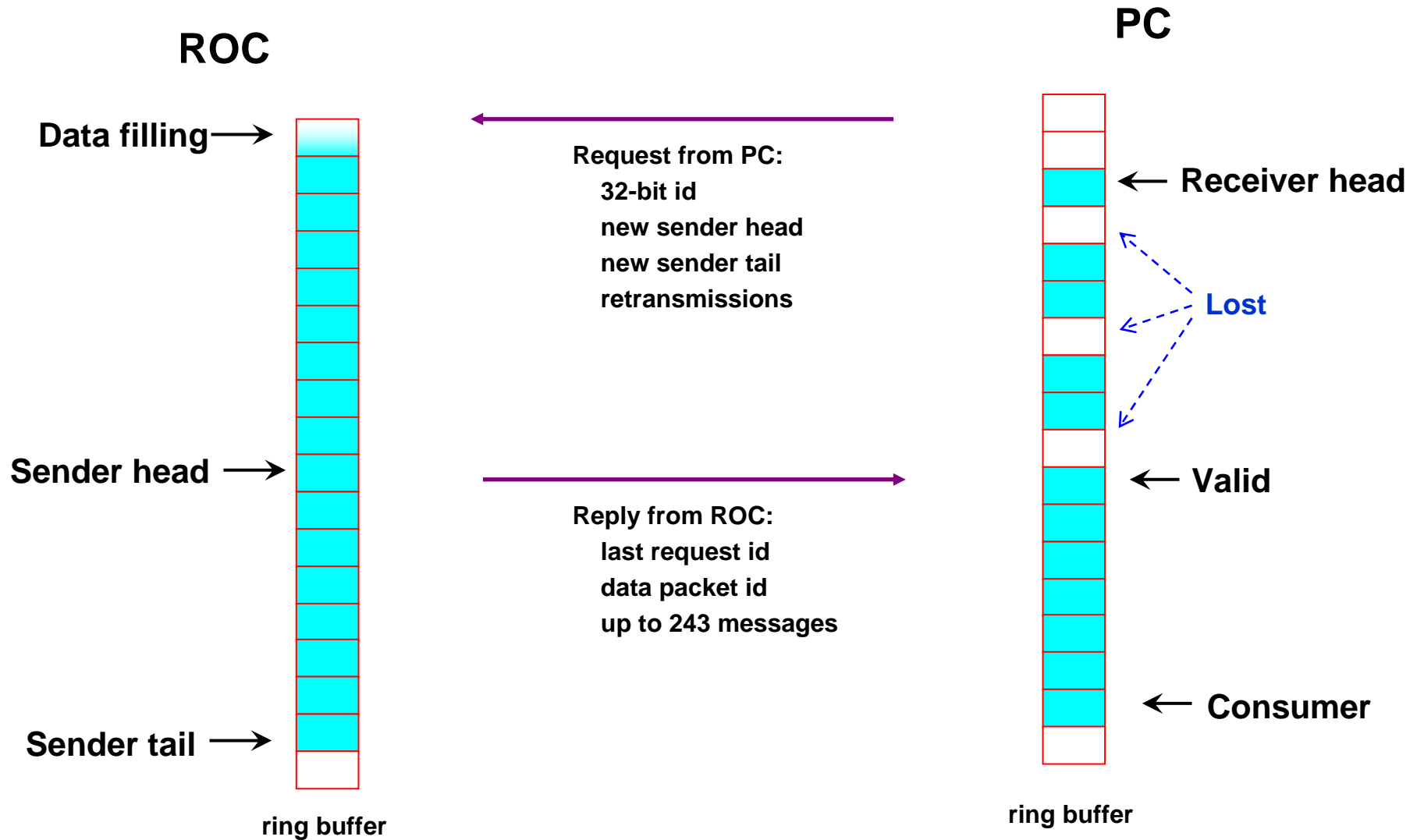
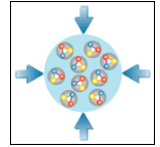
- New concept of mainLoop – 3 "independent" tasks:
 - taking data from FPGA and filling 110 MB buffers
 - dispatching requests, coming via Ethernet
 - sending data to PC

- Fix of half duplex problem with TEMAC Ethernet adapter
- Master and observers operations
- New transport protocol
- Possibility to update software and config files on SD-card
- Refactoring of source code

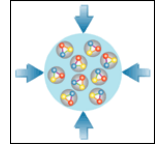
New request-reply transport protocol



When data taking faster than sending



Conclusion



- After 2½ weeks of work:
 - fully reliable transport
 - ~7 MB/s sustained data rate (12.3 MB/s maximum)
 - fixes lot of hardware and software issues

- Near future plans:
 - use DMA in Ethernet adapter to get 12 MB/s sustained – modification and fixes in FPGA code required

- Far future plans:
 - use optic for data, control and time synchronization
 - connect to PC via so-called Active Buffer Board (ABB)