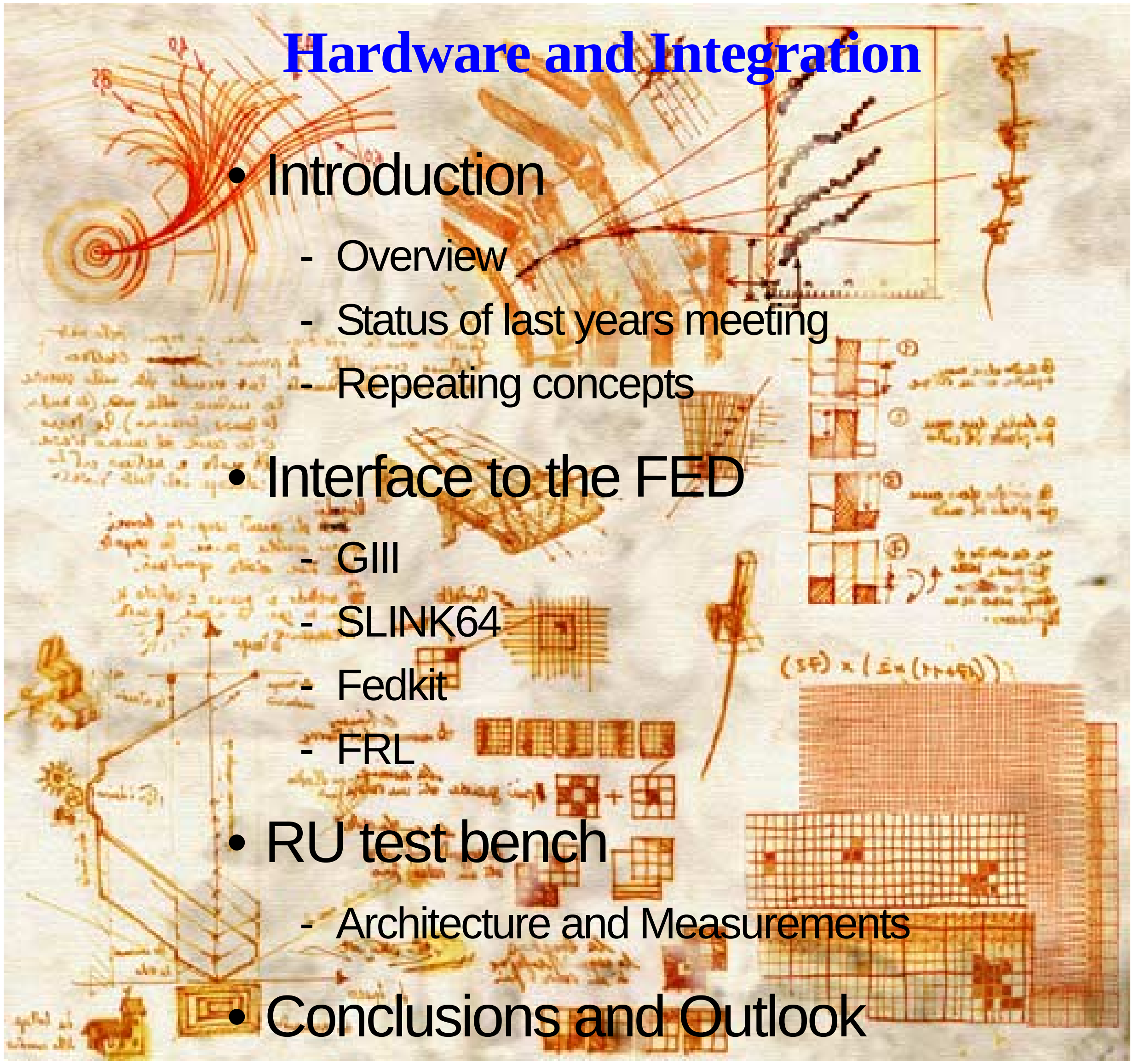
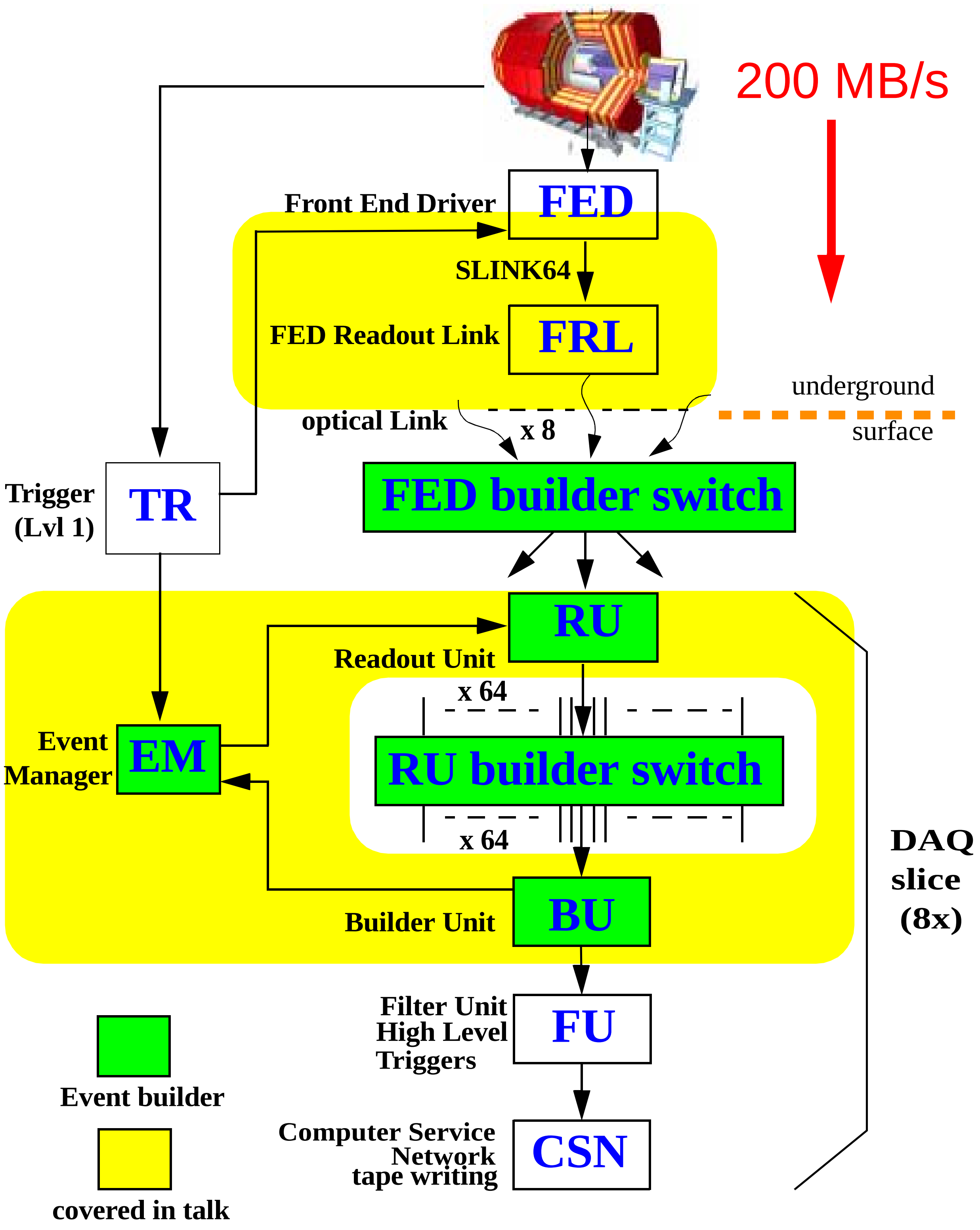
DAQ Column



Hardware and Integration

- Introduction
  - Overview
  - Status of last years meeting
  - Repeating concepts
- Interface to the FED
  - GIII
  - SLINK64
  - Fedkit
  - FRL
- RU test bench
  - Architecture and Measurements
- Conclusions and Outlook

DAQ Column



200 MB/s

Front End Driver FED

SLINK64

FED Readout Link FRL

optical Link x 8

underground

surface

Trigger (Lvl 1) TR

FED builder switch

Readout Unit RU

x 64

RU builder switch

x 64

Event Manager EM

Builder Unit BU

DAQ slice (8x)

Filter Unit High Level Triggers FU

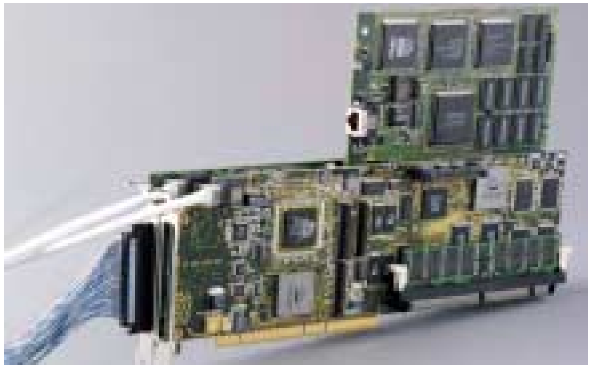
Computer Service Network tape writing CSN

🟩 Event builder

🟨 covered in talk

One year ago: The Status

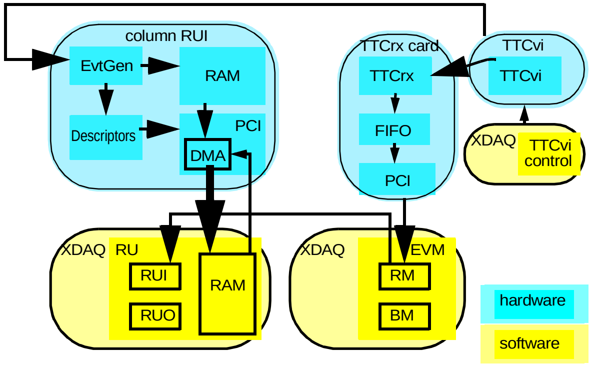
- Components:
  - Optical Link Project
  - Hardware Readout Unit
  - Hardware Builder Unit
- Integration
  - First Column: RU-Input, RU, EVM
  - XDAQ usage
  - Functional tests
  - Low performance
- Old Outlook
  - PC based RU and BU
  - Performance studies for PC with read-DMA and write DMA in parallel



measured: 192 MB/s

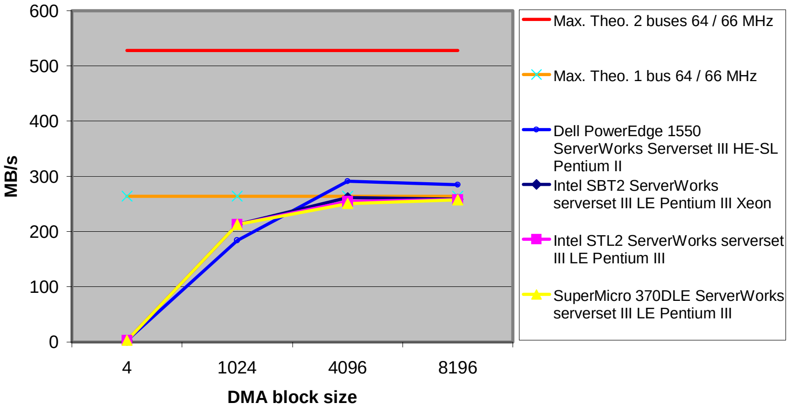
4 kB packet size (now 16kB)

PCI clocked at 33MHz



Hardware and Software integration

Stable operation in XDAQ environment



Reused Components / Concepts

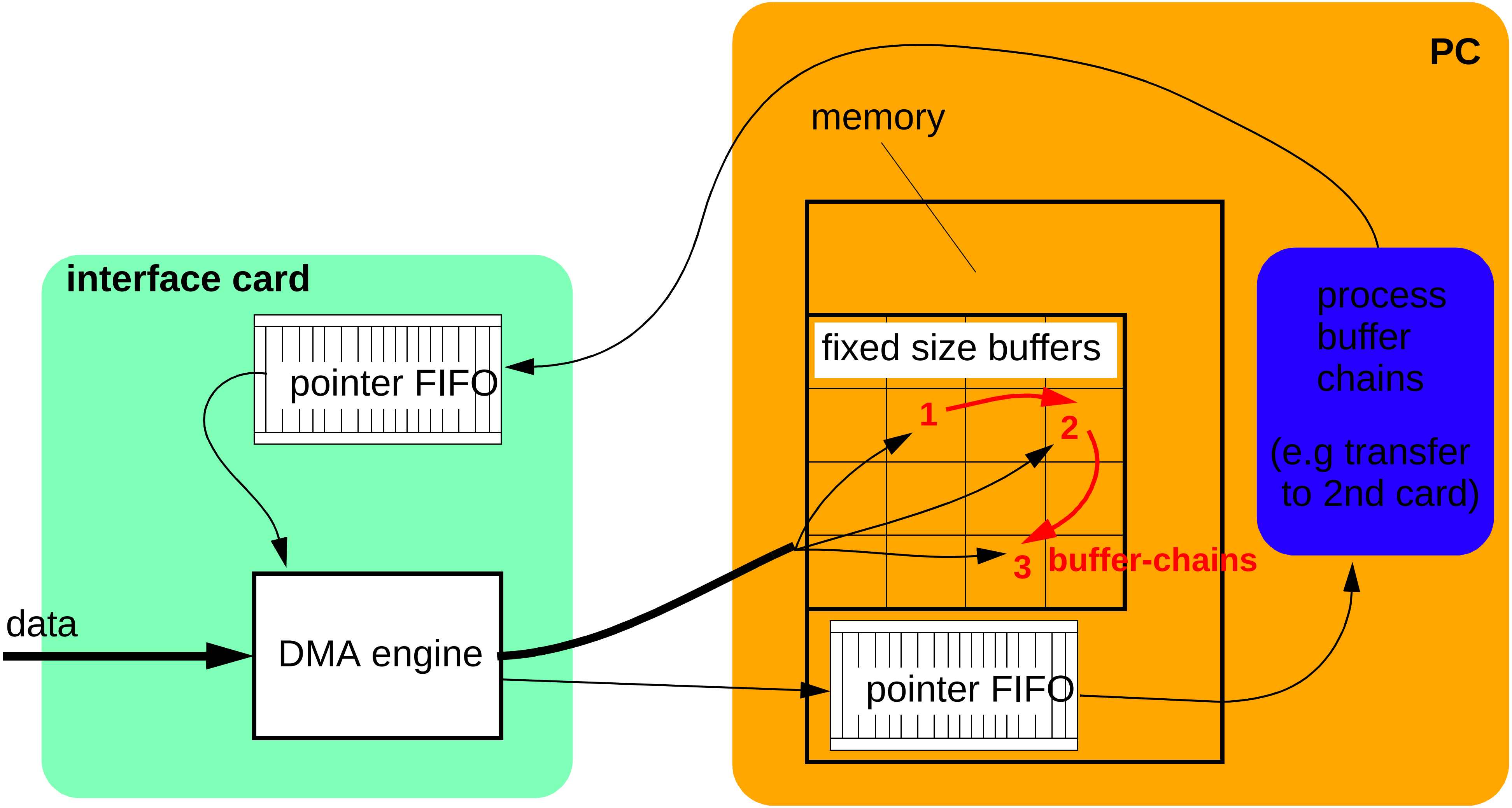
- Hardware: Generic III platform
  - PCI 64 bit / 66MHz “universal card”
  - One FPGA (PCI interface + user code)
  - 32 MB SDRAM
  - Connectors: SLINK64 & multi purpose connector
  - Software Kit to download firmware “in situ”

- Software: the buffer loaning scheme
  - Used to transfer data from external card into PC (or vice versa)
  - Implements “zero copy” scheme:
    - CPU does not copy data (free for more important jobs)
    - In PC only pointers are handled around

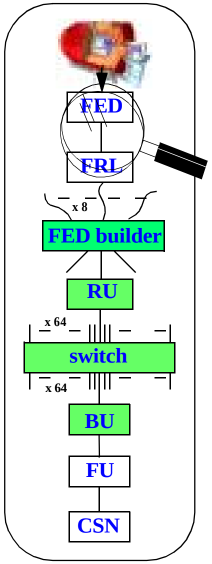
Requirements for data transfer interface: hardware card - computer memory:

- Length of data packet not include in header
- CPU must not be loaded with data copying
- Hardware cards have DMA engines but small memory
- Robustness against fluctuations:
  - Data-volume
  - CPU availability (Linux is NOT Real Time)

Buffer loaning scheme: functional diagram for card to PC transfer



The Interface to the FED: SLINK64



Requirements:

- Easy to use for FED (FPGA friendly protocol)
- Data rate sustained >= 200 MB/s
- Data rate peak >= 400MB/s

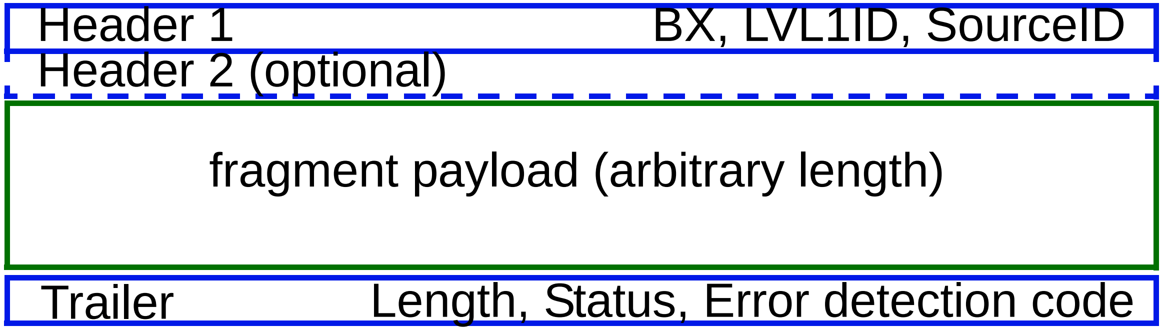
Solution: SLINK64 protocol

- SLINK64 is based on SLINK; modification:

32 bit → 64 bit

40 MHz → max. 100 MHz

- Data format:



Header 1 BX, LVL1ID, SourceID

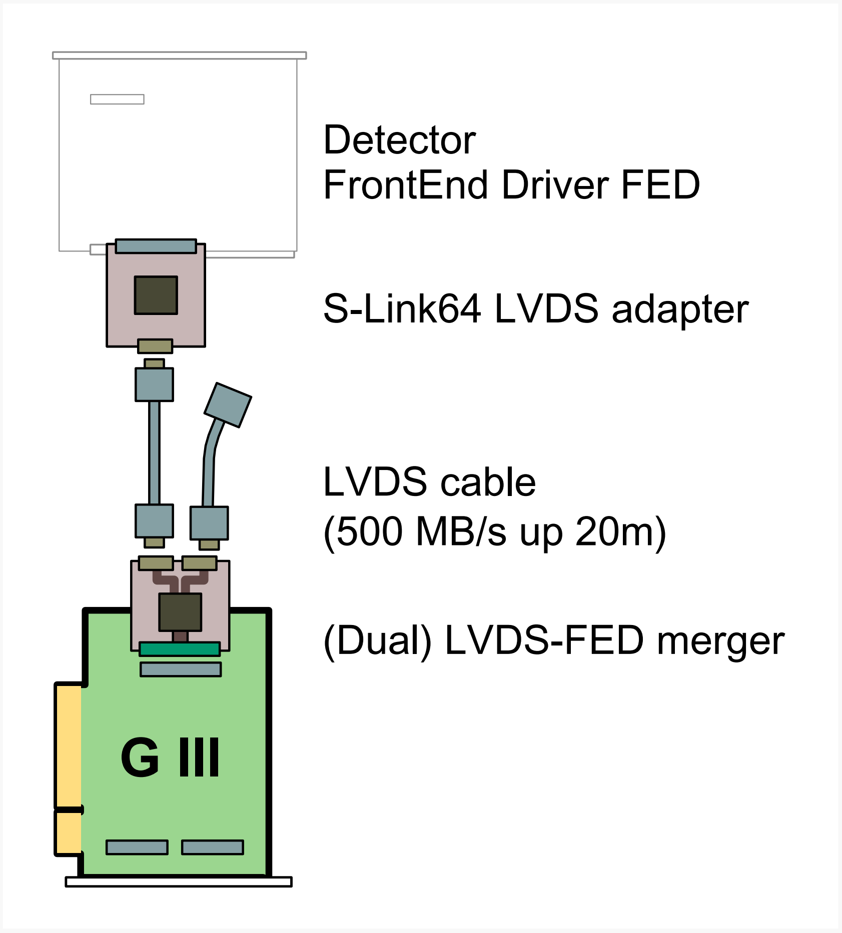
Header 2 (optional)

fragment payload (arbitrary length)

Trailer Length, Status, Error detection code

Implementation and test bench of SLINK64

- LVDS implementation
  - Sender: FPGA, LVDS driver
  - LVDS cable (see below)
  - Receiver:
    - Merges up to two sources
    - FIFOs and FPGAs



Detector
FrontEnd Driver FED

S-Link64 LVDS adapter

LVDS cable
(500 MB/s up 20m)

(Dual) LVDS-FED merger

- Measurement Setup
  - Data generator GIII
  - Data receiver GIII ⇨ PC
  - Random data generator

LVDS cable measurement results

vendor	length [m]	testing time	rate [MB/s] / ν [MHz]	result
AMP	2	1 month	800 / 100	no error
	7.5	8 hours	800 / 100	no error
	2+7.5+7.5	8 hours	528 / 66	no error (cables were connected via home made passive screened boxes)
Amphenol	15	4.5 hours	528 / 66	no error
	20	ε	264 / 33	errors
3M	10	1 hour	528 / 66	no error
	15	1 hour	528 / 66	no error

Conclusion for LVDS - SLINK64 prototype

- Fully functional SLINK64 has been built
- Easy to implement LVDS technology
- Throughput achieved is higher than needed

  528 MB / s at 17 m (400 MB/s required)
- More tests needed
  - Long term test with long cable
  - Use of “standard” test patterns

Remark: Cable paths in control room are not yet defined.

Fedkit

- Facility to test FED interface to DAQ.

Hardware components:

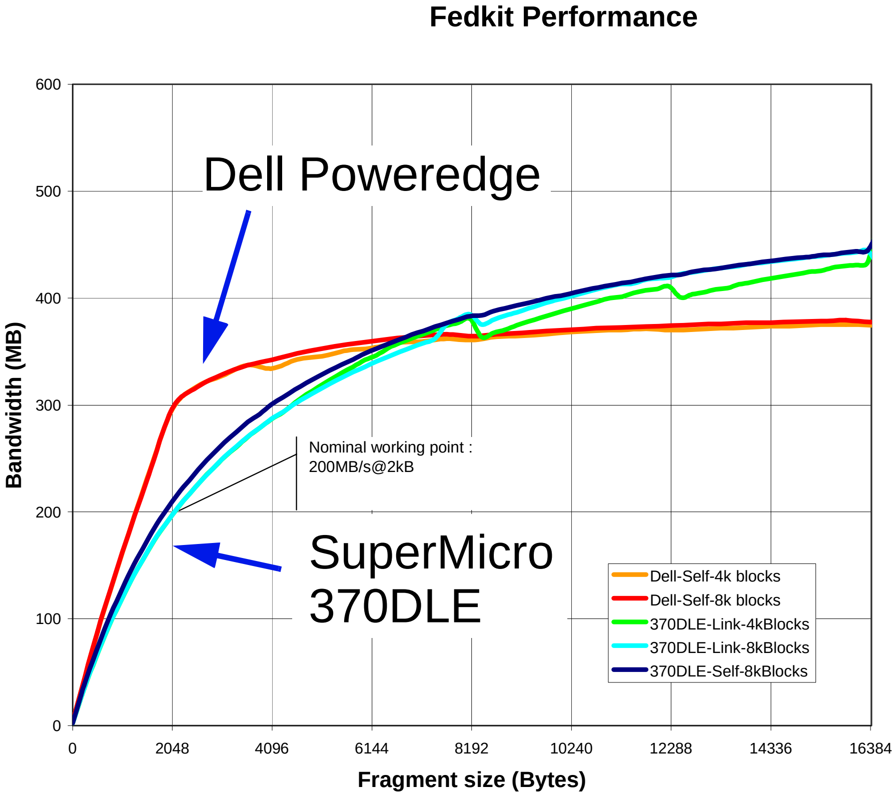
- SLINK64 sender
- Cable
- Receiver card
- GIII to be plugged into Linux PC

Software:

- Driver which handles hardware interaction
- XDAQ application with simple API

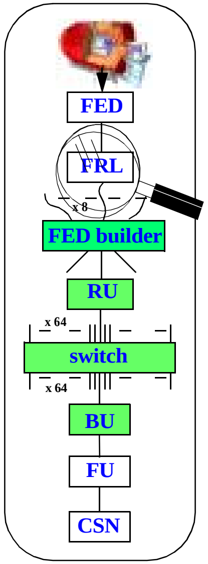
Performance

- Measurement: Fedkit driver only
- Various operation modes
- Performance depends on PC hardware



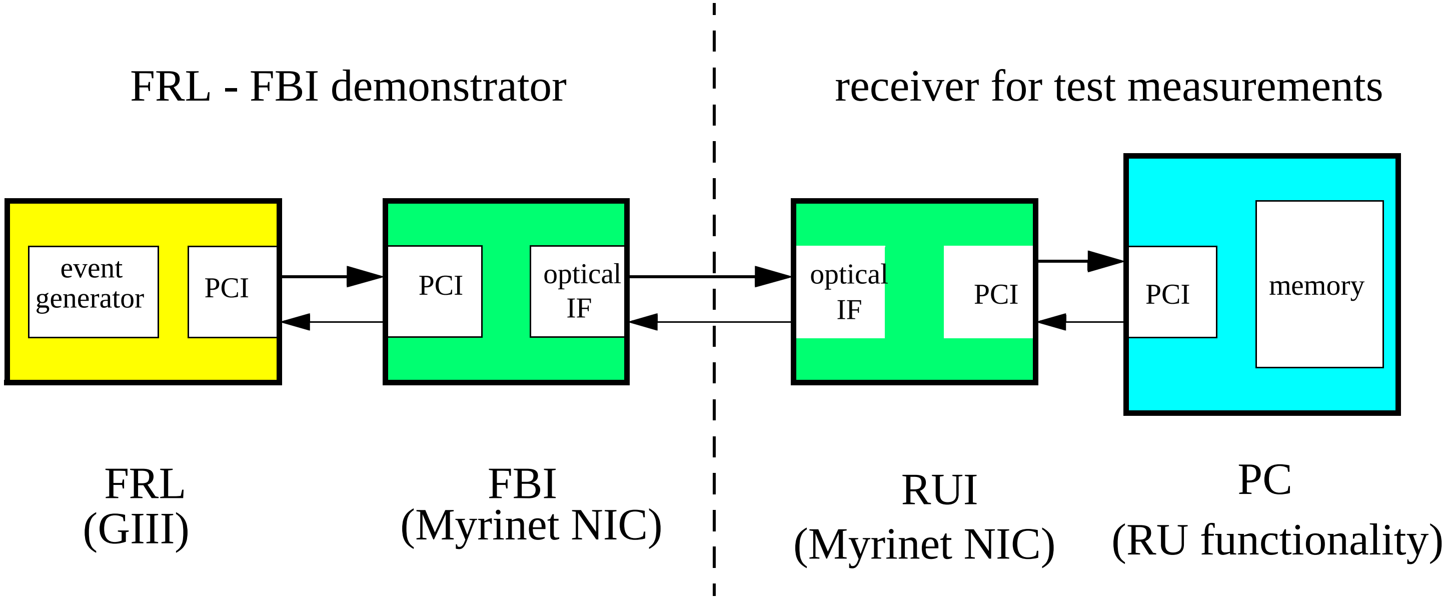
Fedkit in CMS

detector		pieces	date	remark
TriDAS		8 + 16 GIII	spring 2003	FED Builder demonstrator
subdetector Fedkits	Pixel	2	end 2003	
		2	end 2004	
	Tracker	2	12/2002	
	Preshower	3	10/2002	
	HCAL	2	9/2002	
	RPC	1	6/2002	
special	GTP emulator	1	-	
	ECAL	2	-	for link tests
total		23 + 16		



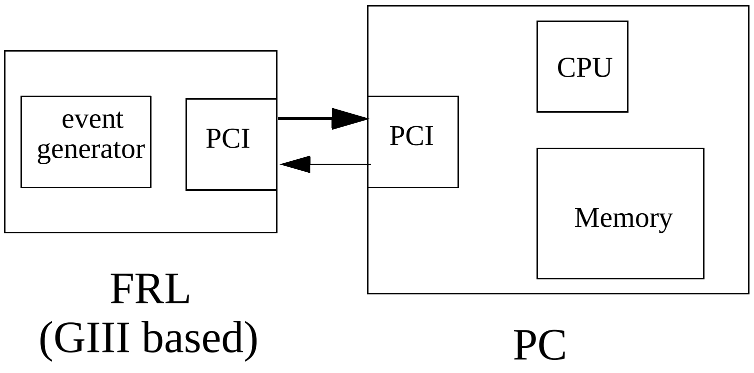
FRL Demonstrator

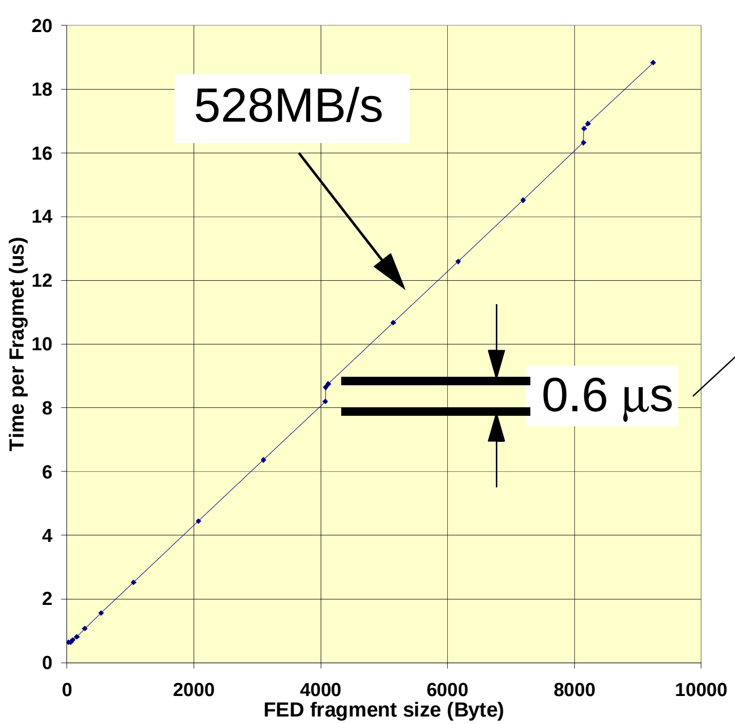
- Scope: Interface FRL, FBI (Front End Builder Input)
- FRL: GIII
- FBI: Myrinet NIC
- Data generator in GIII

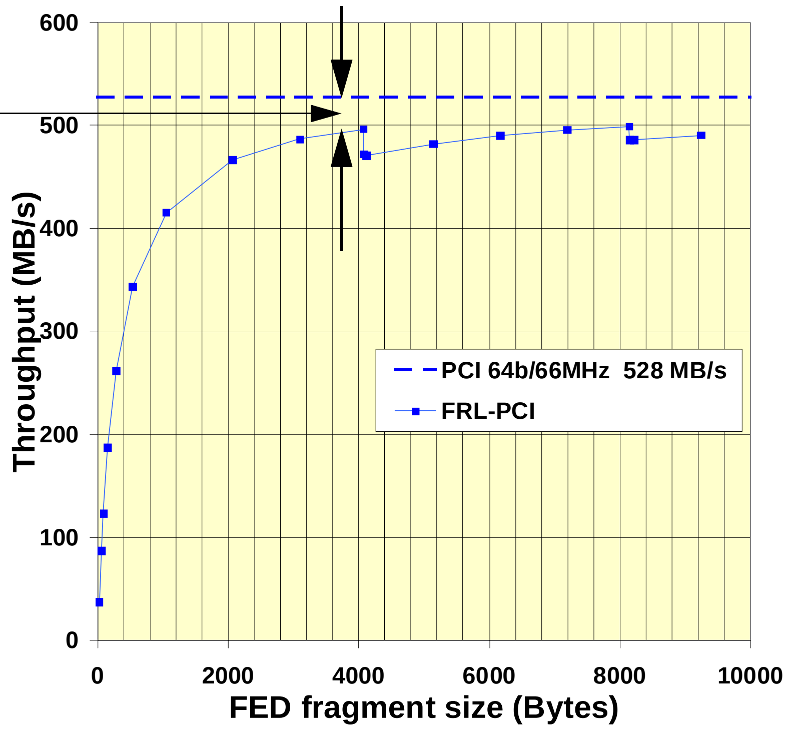


Measurements 1: FRL protocol

- PC emulates NIC card
  - Measures performance of FRL (PC is much faster than NIC)
  - Details: 4 kB pages, 66 Mhz 64 bit PCI-bus, 1024 page entries in GIII, fragment parameters for 512 fragments in GIII FIFO by PC

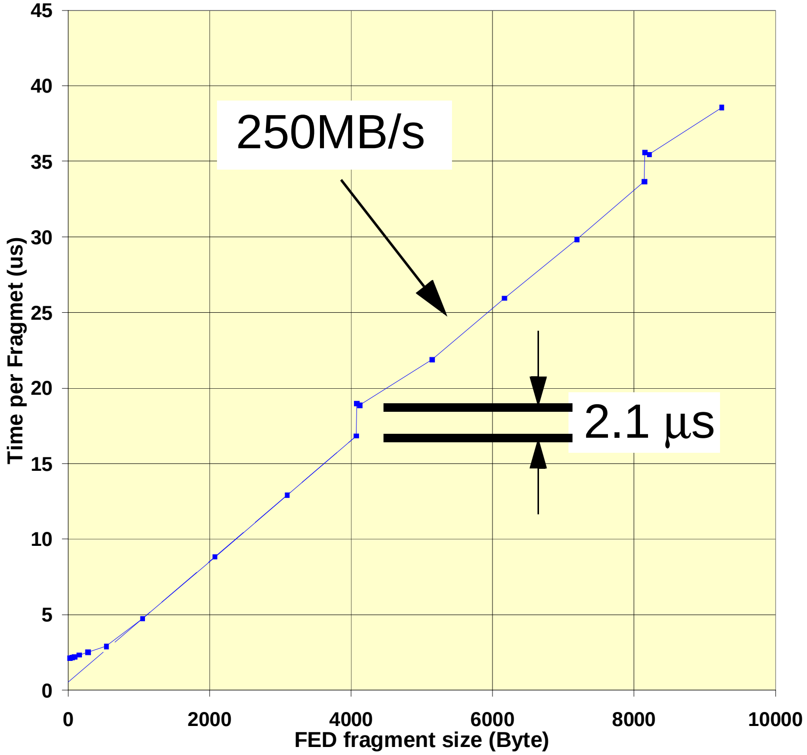


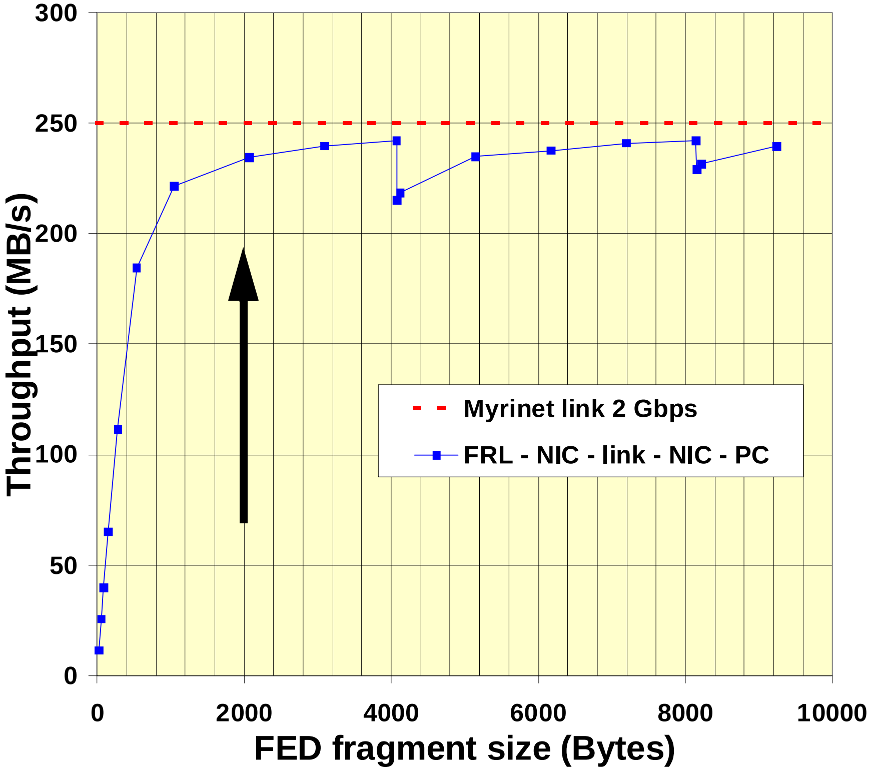




Measurements 2: full test bench

- Test 2: FRL - NIC - NIC - PC
  - Setup is Myrinet limited (theoretical max. 250 MB/s)
  - Offset 2.1 µs is partly shadowed (CPU works while DMA goes on)
  - 0.5 µs “irreducible” offset





Conclusions

- FRL - FBI interface performs close to theoretical limit
  - DMA performance of FRL is at theoretical limit: 528 MB/s
  - Small offset of 0.6 μs / DMA
  - With 2 kB fragments 460 MB/s throughput

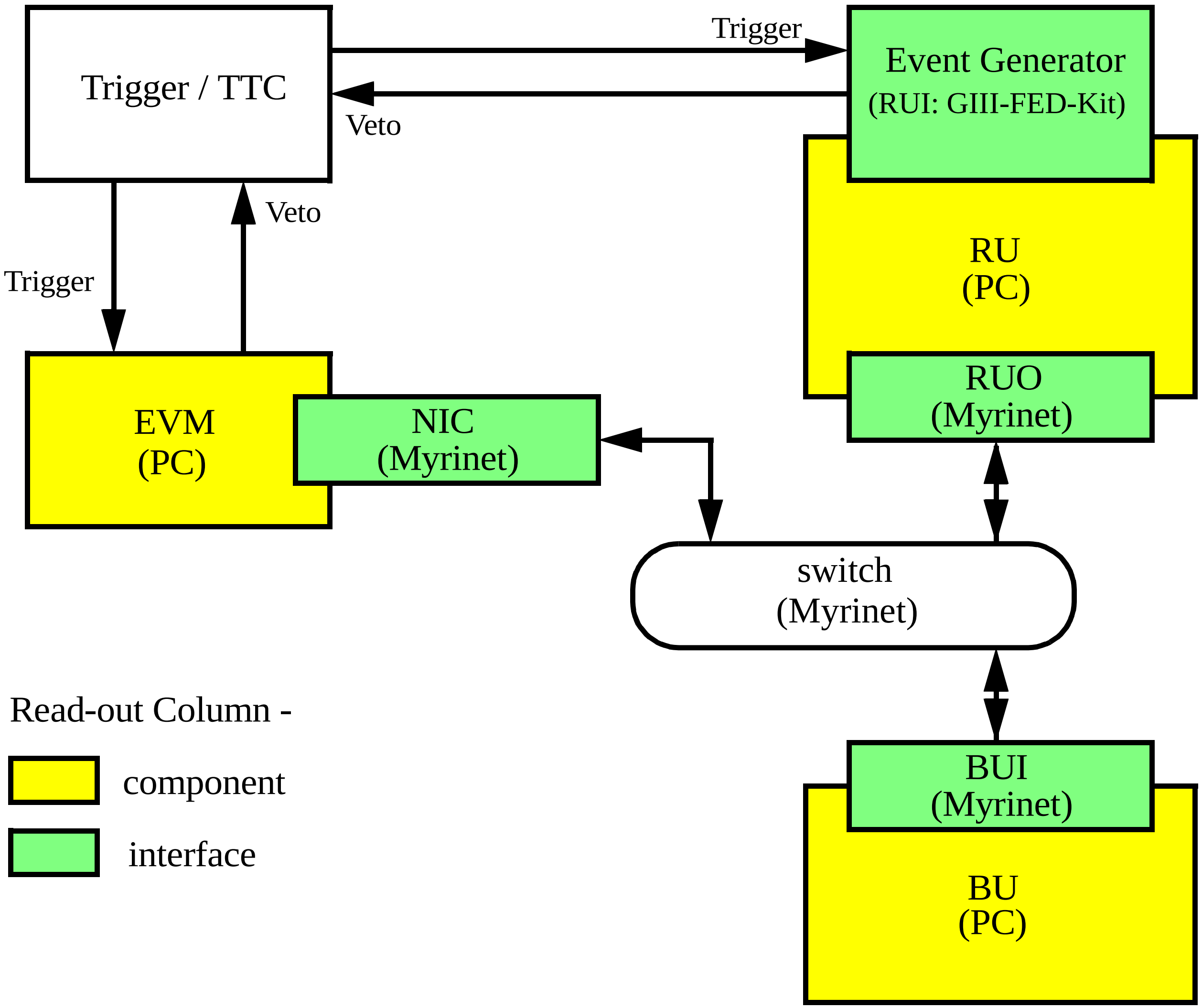
- FRL - FBI - NIC - PC chain
  - Limited by Myrinet performance
  - Bandwidth 230 MB/s for 2 kB fragments

CRUDE: Column for Readout Unit Development

PC based RU demonstrator

- Layout
- Features
- Measurements
- Conclusions

CRUDE layout



Trigger / TTC

Trigger

Veto

Event Generator
(RUI: GIII-FED-Kit)

Veto

Trigger

RU
(PC)

RUO
(Myrinet)

EVM
(PC)

NIC
(Myrinet)

switch
(Myrinet)

BUI
(Myrinet)

BU
(PC)

Read-out Column -

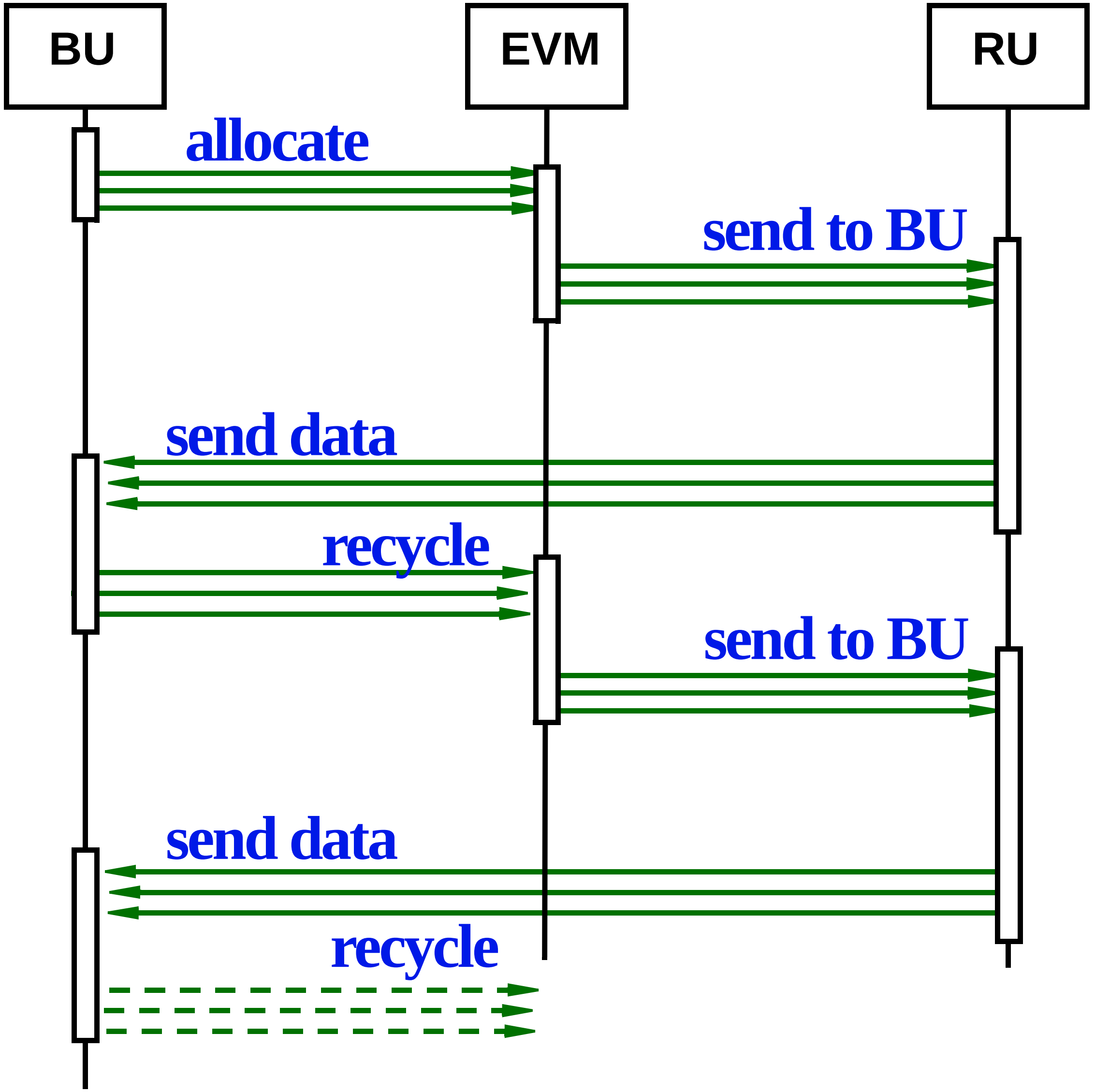
🟨 component

🟩 interface

Test bench features for RU-PC

- XDAQ software environment
- XDAQ applications used and modified as needed
    - Event Manager Protocol: indirect mode
- Data source (RUI replacement): Fedkit with trigger and back pressure
    - Memory allocated by XDAQ
    - Backpressure latency compensated for with internal trigger counter
- Data output: BU
    - Myrinet card
    - GM transport layer (software from Myrinet)
- Implemented data checks:
    - Event numbers (always, in various places)
    - Data can be checked in BU (optional; not done if speed is measured)
    - No error recovery tried

Event Manager Protocol: Indirect Mode



Simplification

- EVM assumes there are always triggers
  - Naive readout of TTCrx setup too slow.
  - Need Fedkit like architecture under development:
    - TTCrx readout via DMA: CPU can work in parallel
    - Multiple buffers: No backpressure if other tasks are scheduled

- Consequences:
  - No latency due to late arrival of trigger in EVM

    ⇒ Less memory required by RU

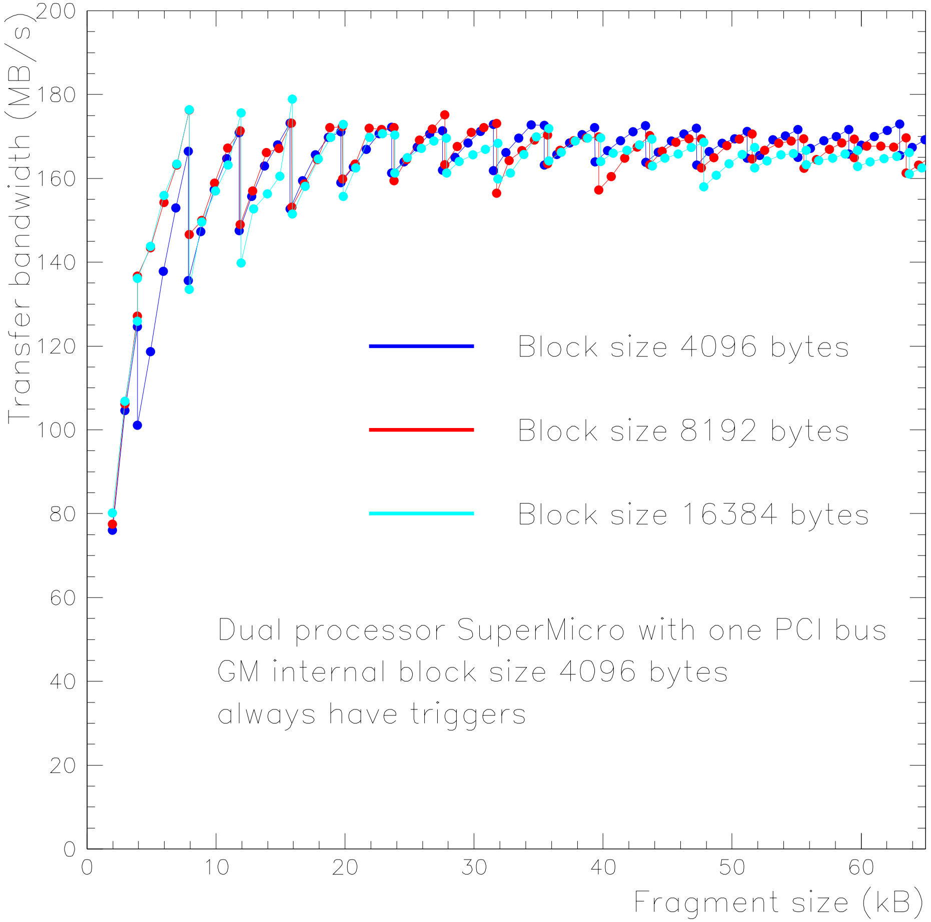
    ⇒ No consequences on RU throughput measurements

Parameters to adjust

- PC type (chipset and motherboard)
- PC - system parameter
  - Bigphys size: DMA - capable RAM to buffer event fragments
- Buffer handling in XDAQ
  - Blocksize: blocks used in Fedkit (RUI), RU. (GM uses fixed blocksize of 4 kB)
  - Maximum number of blocks circulating for data transfer
- Event fragment generator
  - Fragment size distribution: constant / table driven
- EVM protocol
  - RU: bundling of send request
  - BU: bundling of requests

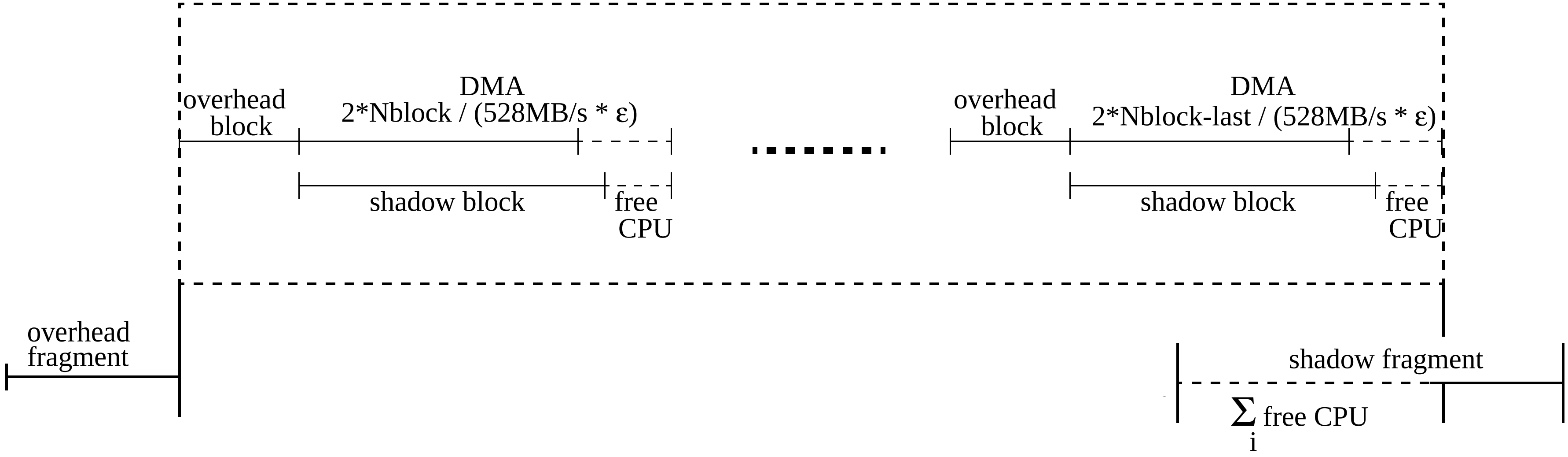
Results: Super Micro different block size

- Fixed parameters
  - 200 MB total buffer (bigphys)
  - 1000 buffers for Fedkit
  - Fixed fragment sizes
  - BU and RU request bundling: 60
- GM works with fixed blocksize of 4 kB -> saw tooth
  - Block sizes multiples of 4 kB
- At 16 kB:
  160 MB/s +/- 13 MB/s



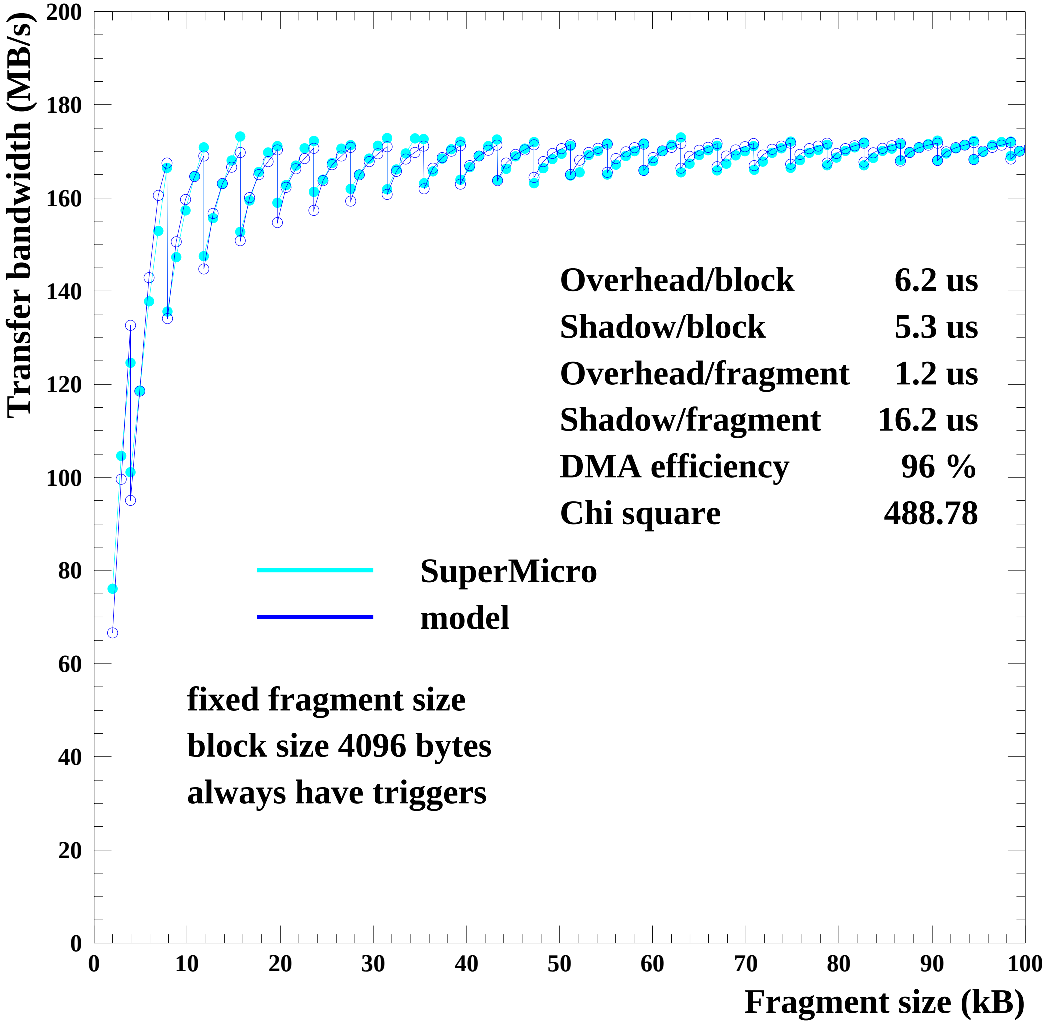
Results: Comparison with a simple model

- Only model PCI bus:



- Try to fit to this model with 5 parameters...

5 parameter fit to the simple model

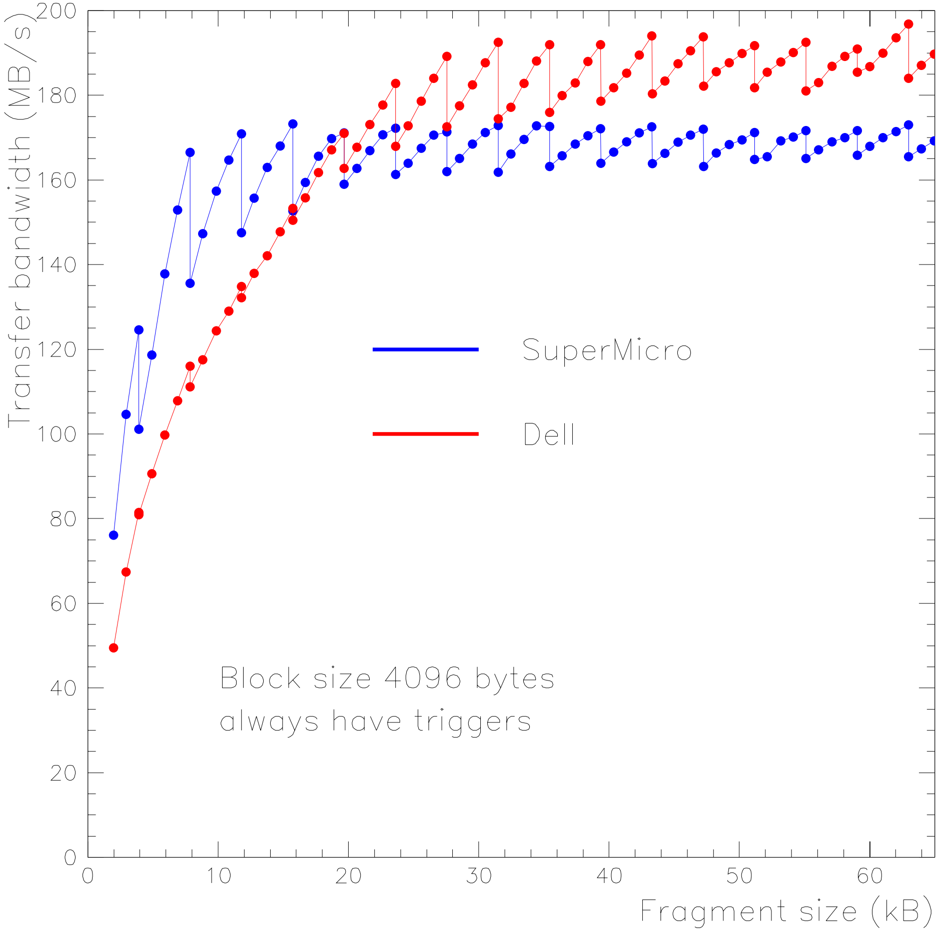


Results: Comparison Dell Power Edge - Super Micro

- High throughput for large fragments
  - DELL has 2 - independent PCI buses
  - Might be reason for higher throughput for large fragments?

- Slow rise:
  - No saw tooth in slow rise (overhead for new block is hidden)

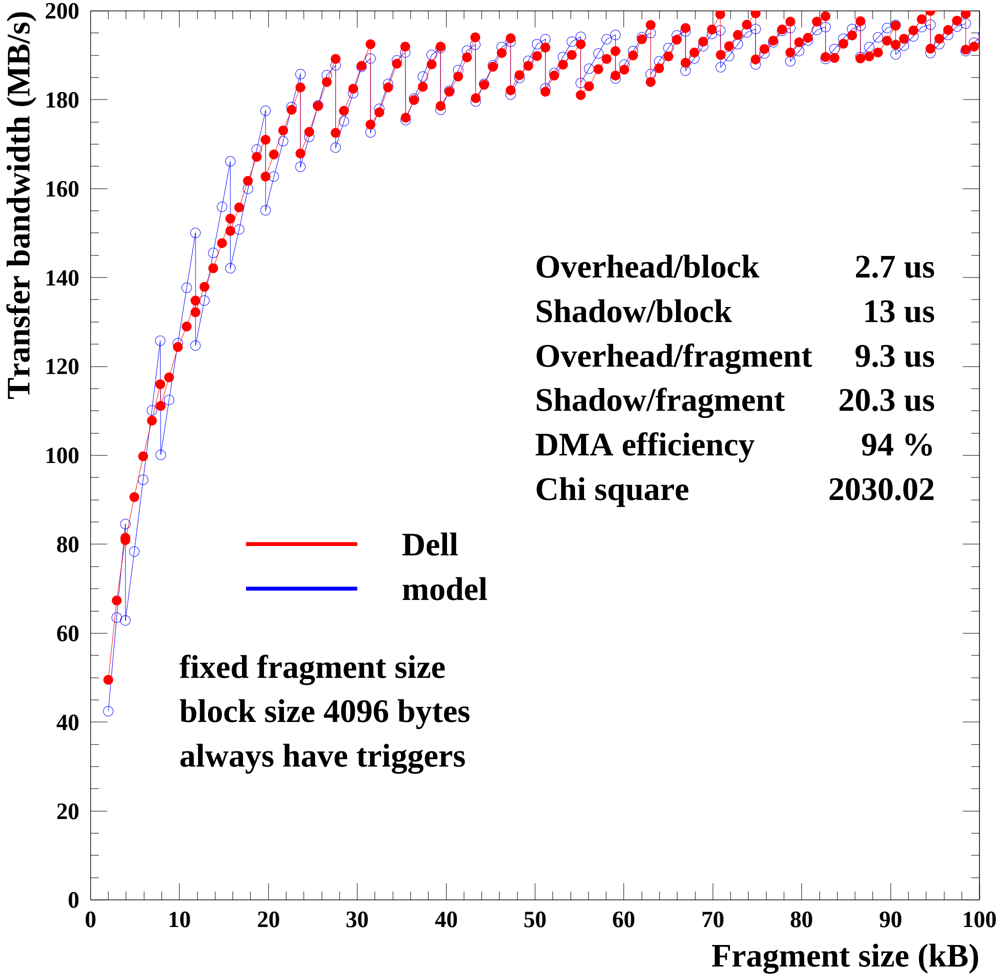
Slow rise for small fragments unexplained



The simple Model fit for the DELL

- Model shortcomings
  - NOT correctly modelled: the two independent PCI buses.
  - With GM point to point transfer limited to 200 MB/s
    ⇒ Network limitation not in model
- Interpretation of fit
  - Large Shadow/block gives large jumps
  - Small overhead/block gives high throughput
  - Model does not describe slow rise details

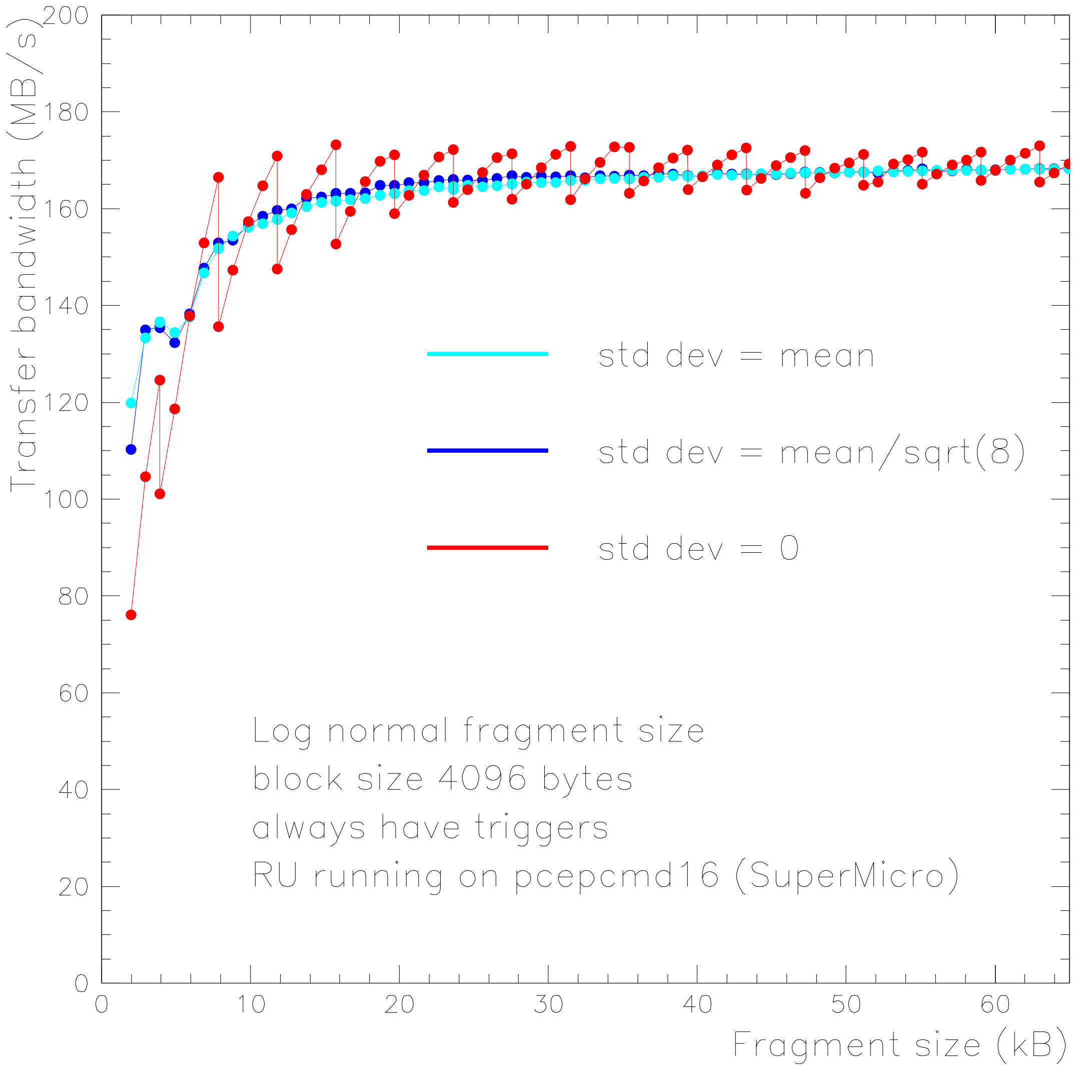
⇒ No clear understanding



Overhead/block	2.7 us
Shadow/block	13 us
Overhead/fragment	9.3 us
Shadow/fragment	20.3 us
DMA efficiency	94 %
Chi square	2030.02

fixed fragment size
block size 4096 bytes
always have triggers

Results: variable fragment size SuperMicro



RU test bench summary

- Two different PCs have been tested

- SuperMicro 370 DLE	160 MB / s with <16 kB fragment>
- DELL Poweredge	145 MB / s with <16 kB fragments>

- Bandwidth 20% from final design goal

Possible improvements:

- Code optimization
- Substitute GM bye in house Myrinet firmware (MAZE)

- Behaviour of DELL Poweredge not completely understood

- Next steps:
  - Implement Fedkit in EVM to run EVM with “real triggers”
  - Implement direct EVM protocol
  - Merge RU-bench with FRL bench
  - Complete system with SLINK - Merger - FRL - Myrinet FED-Builder / RUI - RU - BU
  - Merge with Filter Unit
  - Test with other powerful PCs

Conclusions

Achievements from September 2001 to now:

- SLINK64
  - Designed, built, tested
  - Fully functional
- FEDkit
  - SLINK64 sender, receiver and XDAQ software kit developed
  - Production for FED designers started
- FRL demonstrator
  - Meets bandwidth specifications
- RU demonstrator (PC based)
  - Bandwidth is 20% from design goal

DAQ Glossary used in this presentation

- FED (Front End Driver)
  - last element in subdetector dependent chain; Interface to DAQ System
  - formats data according in standard Frames
  - common hardware interface to DAQ System (Slink)
- SLINK64
  - protocol and physical specification of the interface FED - DAQ
  - easy to handle for FED (writing in a FIFO)
  - 64 bits data width
  - up to 100 MHz clock frequency (not necessarily used in CMS)
- FRL (Front End Readout Link)
  - Interface between SLINK64 receiver and Front End Builder
  - Allows to merge data of up to 2 FEDs
  - Houses programmable NIC which implements the FBI (see below)

- DAQ Link
  - transfers data from underground to surface (O(150m)).
  - require 400MB/s and flow control
  - Interface to FED builder
  - commercial candidates on the market (Myrinet, Gigabit Ethernet...)
- FBI (Front End Builder Input)
  - Interfaces to output of FRL (via PCI bus)
  - Defines the FED-Builder routing
  - Implemented with programmable NIC
- FED builder
  - performs event building for eight FEDs.
  - 1 - 8 outputs connected to 1 - 8 DAQ slices
  - system is scalable from this stage on (a minimal system needs one slice only)

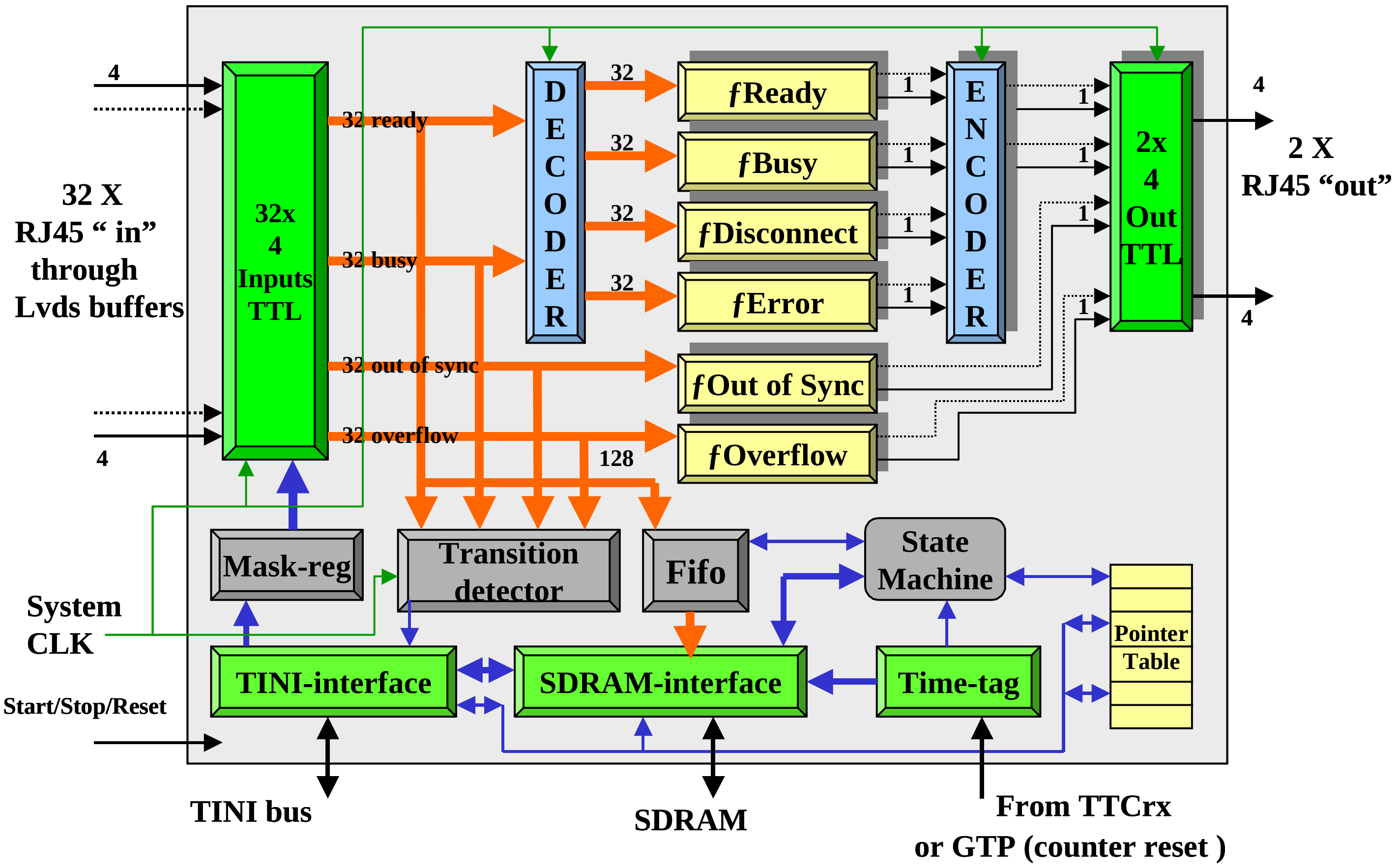
The following items define a DAQ slice:

- EVM (Event Manager)
  - Interface of DAQ System to Trigger
  - implements the Event Builder Protocol, involving the RUs and the BUs (see below)
  - manages the event-identifiers
- RUI (Readout Unit Input)
  - receives and merges fragments from FED builder, pushes them into RU
  - implemented with programmable NIC
- RU (Readout Unit)
  - receives:
    - data fragments of the FED Builder (approx. 16 kB)
    - for each Lvl1 Trigger a Trigger Identifier from the Event Manager
    - readout requests from the Builder Unit
  - tasks:
    - the RU contains the buffer memory of the DAQ system
    - associates event fragments with unique identifiers needed for event building
    - perform event building with the RU-Builder protocol.

- RUO (Readout Unit Output)
  - transfers data from RU to RU Builder
  - implemented by Myrinet card
- RU Builder
  - Myrinet switch to perform event-building
  - 64 inputs and 64 outputs
- BU (Builder Unit)
  - sends requests for an event identifier to EVM
  - requests event fragments for that identifier from RUs
  - builds entire events
  - forwards events to Filter units
- FU (Filter Unit)
  - performs level two trigger decision by processing data of entire event.
  - monitors data quality online

FMM Fast Merger Module

- implements logical tree of sTTS signals
- in design phase


4
32 X
RJ45 " in"
through
Lvds buffers
4
32x
4
Inputs
TTL
32 ready
32 busy
32 out of sync
32 overflow
D
E
C
O
D
E
R
32
32
32
32
128
ƒReady
ƒBusy
ƒDisconnect
ƒError
ƒOut of Sync
ƒOverflow
1
E
N
C
O
D
E
R
2x
4
Out
TTL
4
2 X
RJ45 "out"
4
System
CLK
Start/Stop/Reset
Mask-reg
Transition
detector
Fifo
State
Machine
Pointer
Table
TINI-interface
SDRAM-interface
Time-tag
TINI bus
SDRAM
From TTCrx
or GTP (counter reset)