

MBS release V7.0 and recent developments

*Super-FRS EC Meeting
17-19 January 2024*

M. Reese, J. Adamczewski-Musch, N. Kurz, S. Linev, S. Minami, I. Rusanov
GSI/EEL
18.01.2024

Introduction

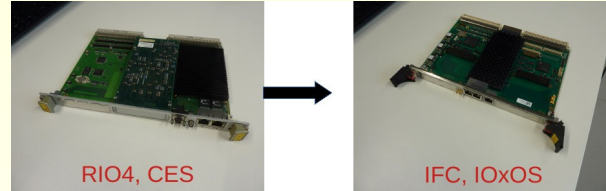
- Overview of MBS V7.0 release, highlight new features
 - New VME platforms: IFC and MVLC
 - White Rabbit timing receiver updates
 - New GSI mass storage interface FSQ
- Ongoing developments on PCIe platform
- About me:
 - PhD in Nuclear Spectroscopy (TU-Darmstadt / GSI)
 - 2016-2023 FAIR Control system: FAIR Timing Receiver (White Rabbit) Hardware and Software
 - Since September 2023: Member of GSI EEL, MBS knowledge transfer, MBS/DAQ support for experiments

MBS history and status

- MBS is the general purpose DAQ system from GSI for 30 years
- Recent releases: V4.3 (2004), V5.1 (2010), [V6.3 \(2017\)](#)
- Mostly used readout systems:
 - VMEbus + modules (GSI and other)
 - PCIe + optical fibre (gossip) + GSI front-end boards
- [See overview talk from 2019 by Nik Kurz \(daq.gsi.de\)](#)
- New release V7.0 ready in January 2023 - [Release notes](#)
 - Installed at GSI MBS cluster at /mbs/v70
 - Default MBS version (mbslogin prod), V6.3 is still available (mbslogin old)

New platform: IFC PPC linux for VME

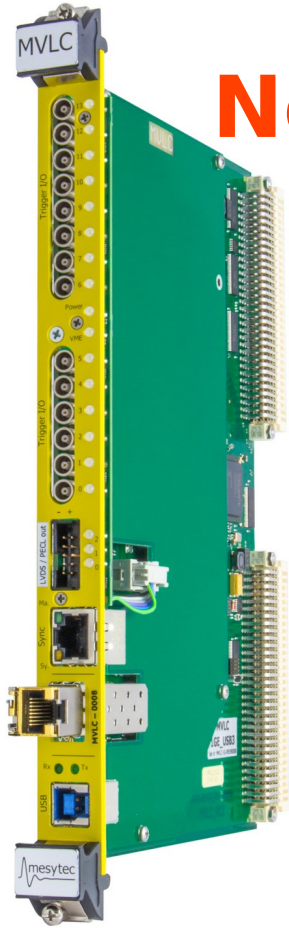
- **CES/RIO4 processor discontinued in 2021**
(in use with >80 modules)!
- **IOxOS IFC:**
 - Kernel modules and drivers software for "jethro" Linux
 - TRIVA trigger module
 - VETAR white rabbit timing receiver + etherbone libs
 - MBS pipe shared memory
 - Example readouts and tests for all known VME modules



(J. Adamczewski-Musch, N.Kurz)

New platform: mesytec MVLC

- MVLC hardware:
 - VME controller/sequencer with **low (no) latency**
 - Controlled via **USB 3** by any commodity MBS PC (X86L-*)
 - Reduced readout flexibility compared to VME processor
- MVME software:
 - **prepare initialization and readout sequences**



MVME software

Object Info

- MVLCTriggerI/O
 - DAQ Start
 - Events
 - event_0_catch_triva_trigger_type Trigger=IRQ4
 - Modules Init
 - triva7_master Type=Triva Master, Address=0x02000000
 - Module Reset
 - Module Init
 - VETAR Type=UserModule_01, Address=0x50000000
 - Readout Loop
 - Cycle Start
 - triva7_master
 - VETAR
 - Cycle End
 - Multicast DAQ Start/Stop
 - event_1_start_acquisition Trigger=IRQ14
 - event_2_stop_acquisition Trigger=IRQ15
 - event_3_hardware_trigger_type_1 Trigger=IRQ8
 - Modules Init
 - V830
 - V785
 - UserModule_03, Address=0x09100000
 - UserModule_03, Address=0x09200000
 - UserModule_01, Address=0x00020000
 - UserModule_01, Address=0x00040000
 - VFTX
 - VFTX
 - triva7_trigger_reset Type=Triva Trigger Reset, Address=0x02000000
 - V1742 Type=UserModule_06, Address=0x00070000
 - Readout Loop
 - Multicast DAQ Start/Stop
 - event_4_hardware_trigger_type_2 Trigger=IRQ9
 - event_5_hardware_trigger_type_3 Trigger=IRQ10
 - event_6_hardware_trigger_type_4 Trigger=IRQ11
 - DAQ Stop
 - Manual

Module Init for module triva7_master (on X86L-114)

```
1 TRIVA registers:
2 # - Status Register      02000000 (hex)
3 # - Control Register    02000004
4 # - FCATIME             02000008
5 # - CTIME               0200000C
6
7 # Triva module initialisation as MASTER
8 # Write Control Register (0x02000004)
9 write A32 D32 0x0004 0x00000004 # MASTER - has to come first!
10 write A32 D32 0x0004 0x00001000 # disable trigger bus
11 write A32 D32 0x0004 0x00000010 # HALT
12 write A32 D32 0x0004 0x00000040 # CLEAR
13 write A32 D32 0x0008 $(65535 - 20) # Fast Clear
14 #write A32 D32 0x000C $(65535 - 1700) # Ctime
15 write A32 D32 0x000C $(65535 - 1300) # Ctime
16
17 ##### MVLCTriggerI/O setup for TRIVA
18 # Optional signal aliases: map
19 # If no alias is set the trigger is directly taken as the IRQ number.
20 # Otherwise if one of the aliases is used all pairs will be active too.
21 # Mapping a trigger value to 0 makes the MVLCTriggerI/O ignore the trigger.
22
23 writeabs A32 D16 0xFFFF7000 14 # TRIVA Trigger 14 (Start)
24 writeabs A32 D16 0xFFFF7002 14 # mapped to MVLCTriggerI/O 14
25
26 writeabs A32 D16 0xFFFF7004 15 # TRIVA Trigger 15 (Stop)
27 writeabs A32 D16 0xFFFF7006 15 # mapped to MVLCTriggerI/O 15
28
29 writeabs A32 D16 0xFFFF7008 1 # TRIVA Trigger
30 writeabs A32 D16 0xFFFF700A 8 # mapped to MVLCTriggerI/O (data_event)
31
32 writeabs A32 D16 0xFFFF700C 2 # TRIVA Trigger
33 writeabs A32 D16 0xFFFF700E 9 # mapped to MVLCTriggerI/O (scaler_event)
34
35 writeabs A32 D16 0xFFFF7010 3 # Map trigger 3 (bit combination of 1&2) to the
36 writeabs A32 D16 0xFFFF7012 10 # data event too. This avoids the readout getting
37 # stuck when both the data and scaler triggers are
38 # active at the same time.
39 writeabs A32 D16 0xFFFF7014 4
40 writeabs A32 D16 0xFFFF7016 11
41
42 /*
```

Integration of MVLC to MBS

- **Required developments:**
 - **Control of TRIVA with MBS commands** (set trigmod, start acquisition,...)
 - Different readout code for different TRIVA trigger types (mvme + mesytec upgrades!)
 - **MVLC USB data is received by new MBS readout process** (regular MBS subevents to pipe buffer)
 - Readout of **VETAR White Rabbit timing receiver** (produce MBS time stamp format)
 - Adjust readout for all types of VME modules used at GSI (mvme editor)
- **It is now possible to combine multiple branches with mvlc and non-mvlc readout!**

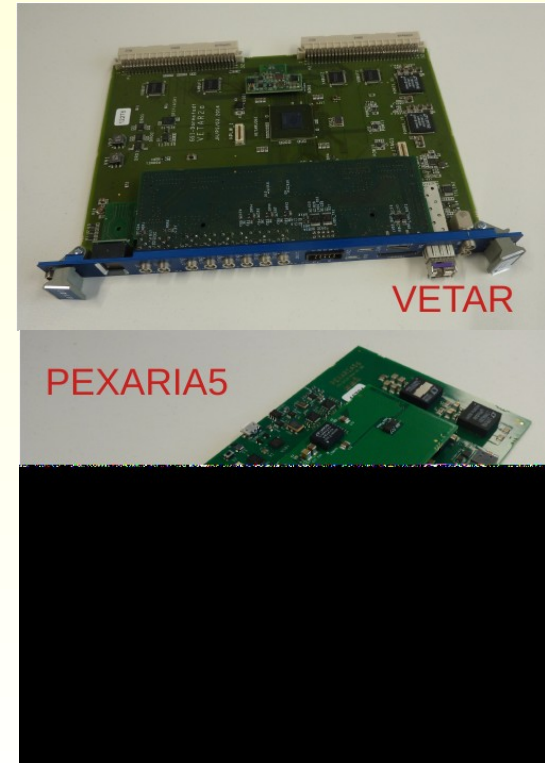
Integration of MVLC to MBS

- All VME modules used at GSI proved to work
- Performance improvement of **factor 3** compared with RIO4 readout for a typical FRS setup
- No other hardware changes required when replacing RIO4 with MVLC
- **Was used in beam end of 2023**

White Rabbit timing receivers

Use of **campus wide timestamp distribution via White Rabbit network** of the FAIR control system

- MBS accepted trigger is time stamped and integrated into event data (8 ns/1 ns resolution)
- Time stamp can be used for synchronization/time sorting of “free running” (aka streaming readout) branches
- Works with VME (also MVLC) and PCIe systems



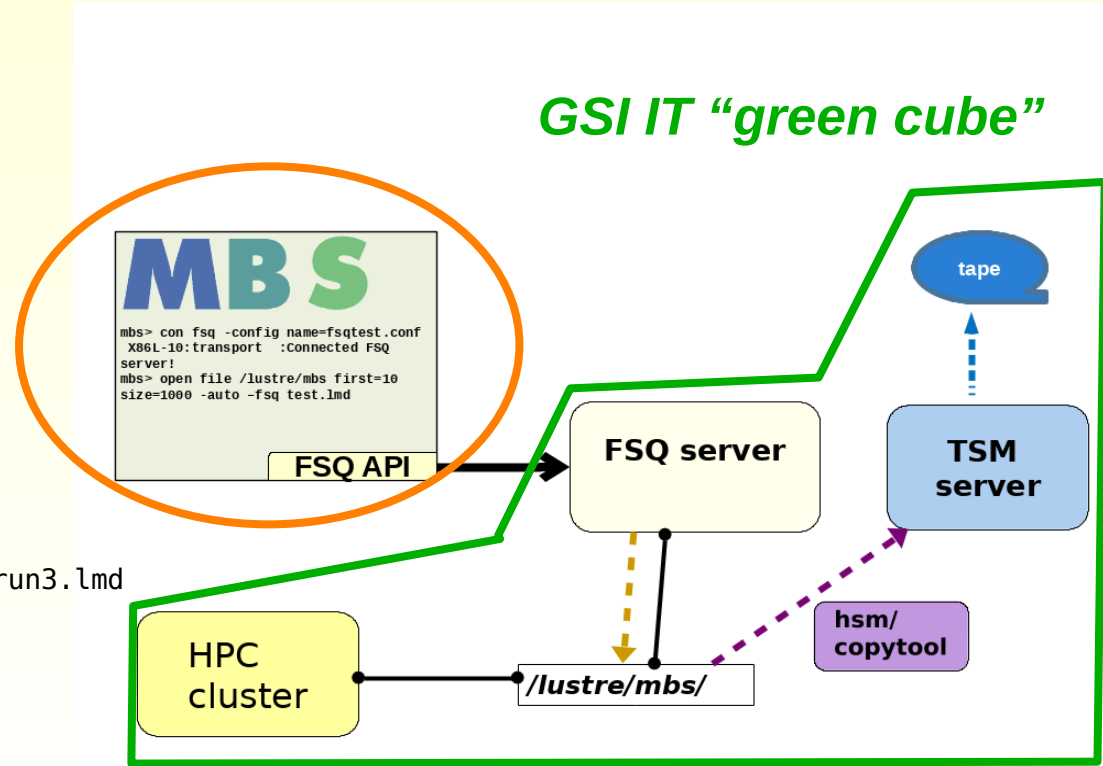
New mass storage interface FSQ

- “**F**ile **S**torage **Q**ueue” (Thomas Stibor, GSI IT)
- **FSQ server** with access to */lustre* and TSM
- DAQ writes data with **lightweight FSQ protocol**, no IBM libs required
- Files are copied to **/lustre first, then archived to tape**
- Developed and **tested with HADES at beamtime FEB22**
- Will be **common GSI storage interface for experiments**

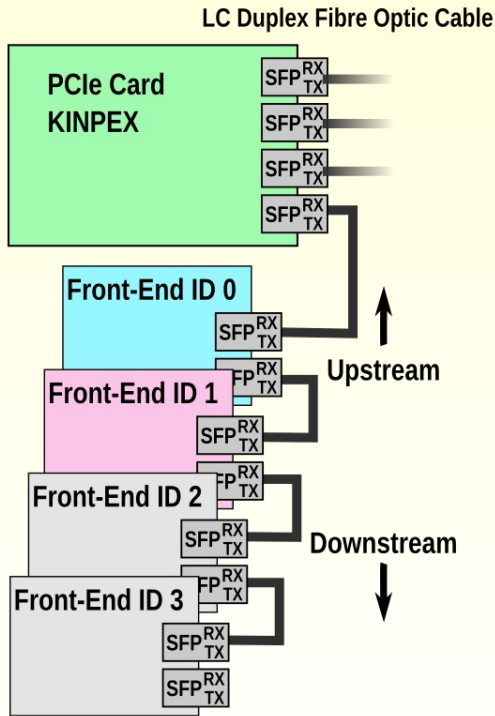
MBS transport to FSQ server

MBS event builder

```
MBS> connect fsq -config myconfig.conf  
MBS> open file -fsq -auto /lustre/exp/mar24/run3.lmd
```



MBS PCIe platform: GOSIP



- Gigabit Optical Serial Interface Protocol
 - Data transfer protocol from FEEs to a PC.
 - 2Gbps (PCIe 4x 2.5 Gbps)
- Address mode
 - Read/write access to 32-bit registers
 - Address space 24bits per FEE
- Block mode (token passing)
 - Fast data transfer to PCIe interface
 - Double buffered block data transfer from all chained FEEs

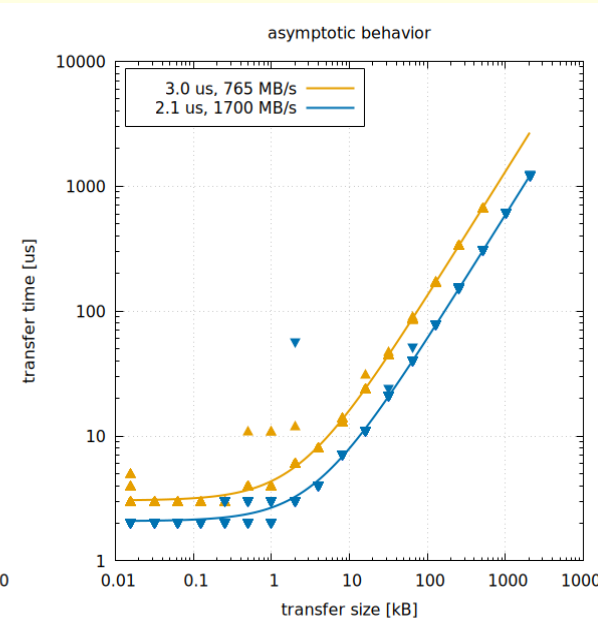
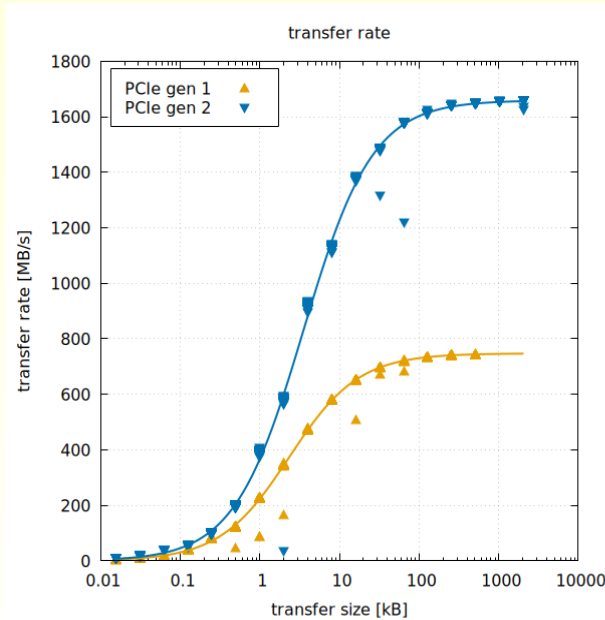
Ongoing development: GOSIP

- **Upgrade to GOSIP with 5 Gbps line rate** (Kintex-7 FPGAs):
 - Clock TDCs (**CLKTDC128, KILOM, MPPC_ROB**)
 - 100 MSPS ADC (**FEBEX4**)
- **Upgrade to GOSIP with 2.5 Gbps line rate** (ECP3 FPGAs):
 - High Resolution TDC (**TAMEXs**)
 - 50 MSPS ADC (**FEBEX3**)
- **Allow dynamic change of line rate** for KINPEX PCIe card: 5 Gbps, 3.125 Gbps, 2.5 Gbps and 2 Gbps.

(S. Minami)

Ongoing development: PCIe

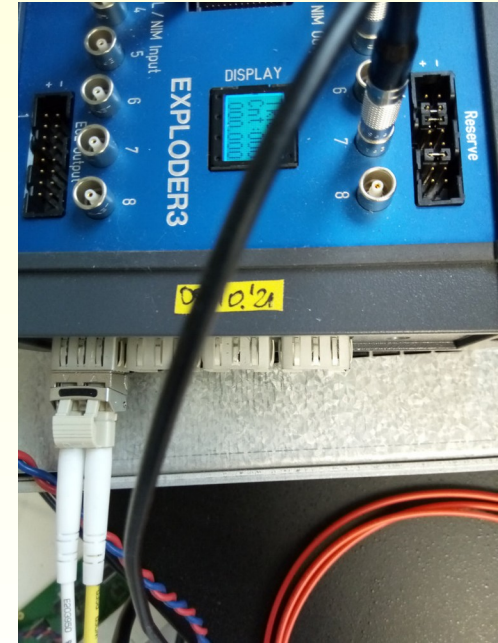
- KINPEX: PCIe gen 1 → gen 2
765 MB/s → 1700 MB/s
- **64-bit DMA addresses** to relax MBS pipe layout constraints



New optical connection between TRIXOR and EXPLODER

- TRIXORFP_FIBER2 replaces ribbon cable with optical link
- No copper between MBS PC and FE-crate
→ no ground loops

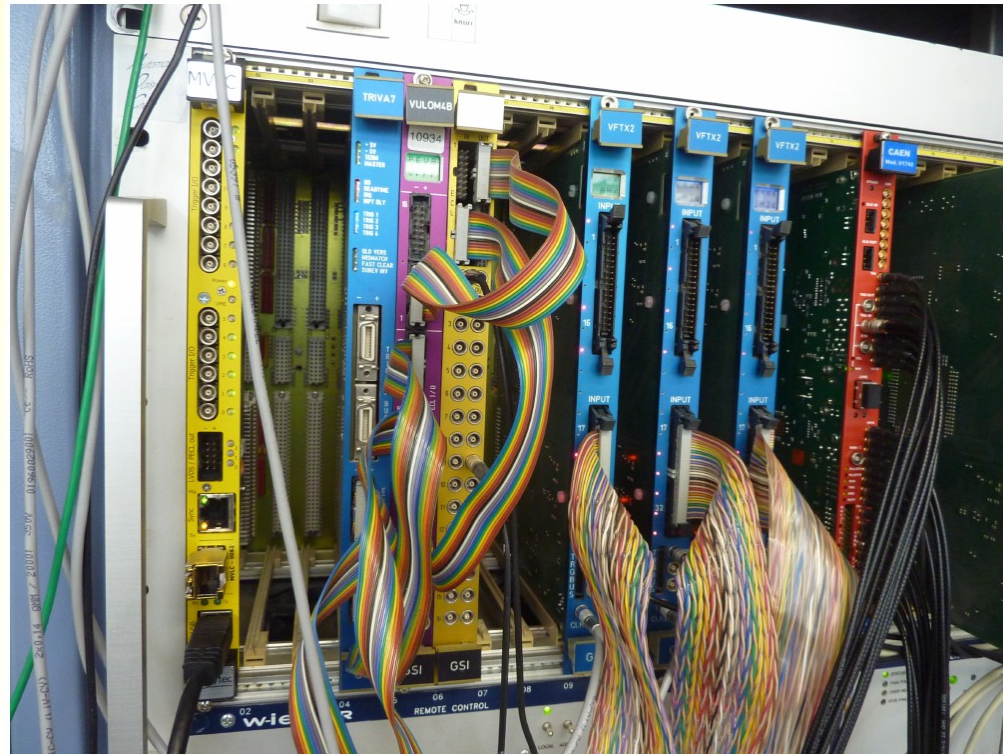
(I. Rusanov, H. Heggen)



Summary

- MBS release V7.0 provides many developments since 2016
- PCIe X86L- hosts have been upgraded to Debian 11
- New VME platforms: IFC, MVLC
- White Rabbit trigger time latch unit supported (VETAR, PEXARIA)
- New GSI storage protocol FSQ integrated to MBS transport (for all MBS Linux platforms, even RIO4)
- Development is ongoing

Thank you!



18.01.2024

M.Reese

17