

MBS release V 7.0

EEL meeting 07.03.2023

Jörn Adamczewski-Musch, Nikolaus Kurz, GSI/EEL
07.March 2023

MBS history and status

- MBS is general purpose DAQ system from GSI for 30 years
- Recent releases: V4.3 (2004), V5.1 (2010), V6.3 (2017) [5]
- Mostly used readout systems:
 - VMEbus + modules (GSI and other)
 - PCIe + optical fibre (gossip) + GSI front-end boards
- See Niks overview talk of 2019 (at daq.gsi.de)
- **New release 7.0 ready in January 2023 – Release notes**

Release 7.0

- Installed at GSI MBS cluster at `/mbs/v70`
- Default MBS version (*mbslogin prod*):
 please rebuild user readouts (*make clean; make;*)!!
- V6.3 is still available as *mbslogin old*
- Code repository now at git.gsi.de (restricted access)

New features

- New platforms:
 - Debian 11 “bullseye” (X86 PCs)
 - QorIQ “jethro” Linux (IFC VME processor)
- Integration of mesytec MVLC VME controller
- White Rabbit timing receiver updates
- New GSI mass storage interface FSQ
- Miscellaneous

New platform: Debian 11 on x86 PC

- Kernel modules and driver software:
 - kinpex/trixor PCIe readout (libmbspex, Qt control GUIs)
 - pexaria White Rabbit timing receiver
- DABC installation (MBS webgui, rgoc command server)
- Go4 installation (. /analysis/bin/go4login)

Please upgrade your machine!

New platform: IFC PPC linux for VME

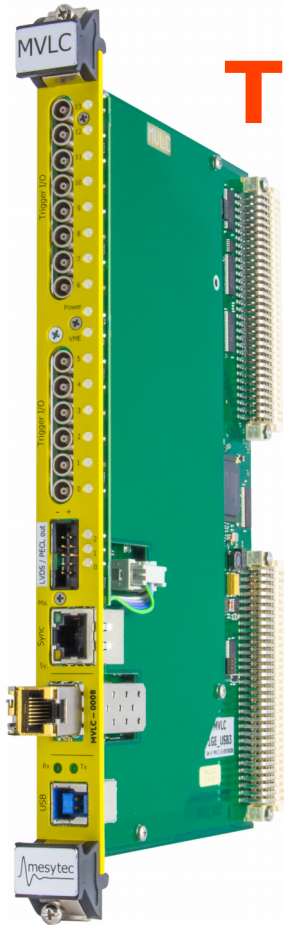
- **CES/RIO4 processor discontinued in 2021** (in use with >80 modules)!
- First try: IoxOS IPV processor
 - Ready integrated to MBS V 7.0!
 - But IoxOS also ceased support end of 2021!
- New hope: IoxOS IFC [\[1\]](#)
 - Freescale PPC P2020
 - QorIQ "jethro" Linux



New platform: IFC PPC linux for VME

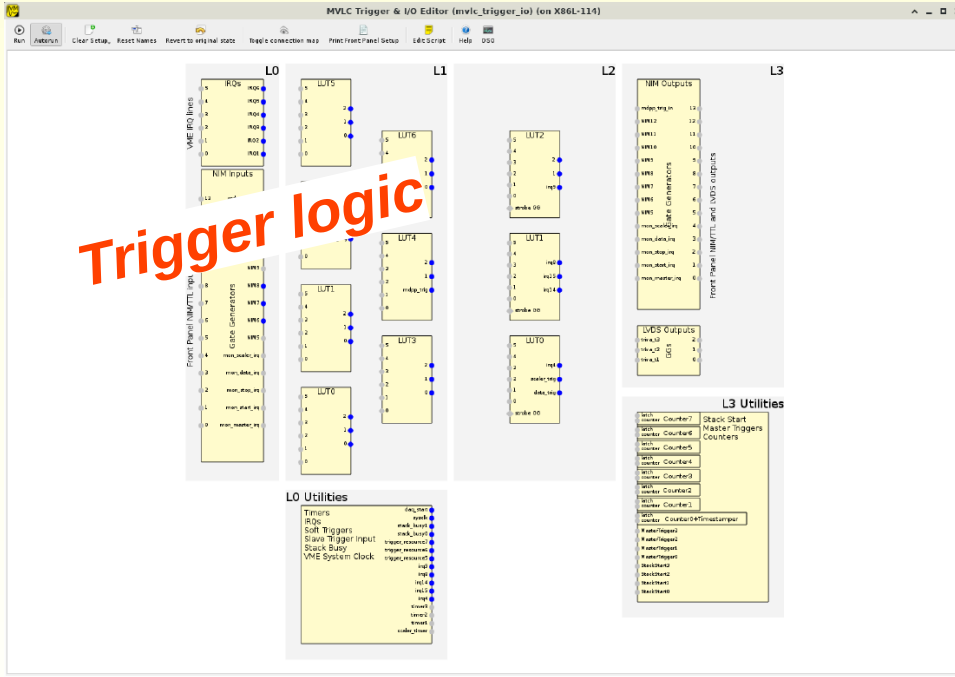
- Kernel modules and driver software
 - TRIVA trigger module
 - VETAR white rabbit timing receiver + etherbone libs
 - MBS pipe shared memory
- Fast direct mapping of VME addresses
- Example readouts and tests for all known VME modules

The mesytec VME controller



- MVLC:
 - VME controller/sequencer with **low (no) latency**
 - Reduced readout methods compared to VME processor
 - **Control and data transfer via USB 3 or ethernet (GSI: USB)**
 - **Host can be “any” commodity MBS PC (X86L-*)**
 - Free programmable I/O
 - Powerful trigger logic (not mandatorily required at GSI)
- MVME:
 - Data acquisition and “analysis” software (C++)
 - **Tool to prepare initialization and readout sequences**
 - Graphical tools to program **trigger logic**

MVME software



all_modules - [DAQ mode] - mvme (on X86L-114)

Start

Histo data: Keep Clear

Run duration: unlimited

VME Controller: Disconnected (MVLIC_USB)

Settings

Reconnect Force Reset

DAQ State: Idle

Analysis State: Idle

Listfile Output:

Write Listfile Format: ZIP fast compression

Run Settings Workspace Settings Run

Current Filename:

Current Size: 0.00 B

Free Space: 5.72 TB

New Open Save Save As Edit Variables More

Triva7_direct_irq_readout.vme

Object Info

- MVLIC Trigger/I/O
- DAQ Start
- Events
 - event_0_catch_triva_trigger_type Trigger=IRQ4
 - event_1_start_acquisition Trigger=IRQ14
 - event_2_stop_acquisition Trigger=IRQ15
 - event_3_hardware_trigger_type_1 Trigger=IRQ8
 - 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

Running time: 00:00:00

Buffers read: 0

Bytes read: 0.00 MB

Data rates: 0.00 buffers/s, 0.00 MB/s

Main window

MVME software

Object Info

- MVLIC Trigger/IO
 - 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 Type=...
 - V785 Type=...
 - MADC_GSI ..._us, Address=0x09100000
 - MADC_GSI ..._UserModule_03, Address=0x09200000
 - V1290 Type=UserModule_01, Address=0x00020000
 - V775 Type=UserModule_01, Address=0x00040000
 - VFTX Type=UserModule_05, Address=0x07000000
 - VFTX Type=UserModule_05, Address=0x08000000
 - 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_trigepr_type_4 Trigger=IRQ11
 - DAQ Stop
 - Manual

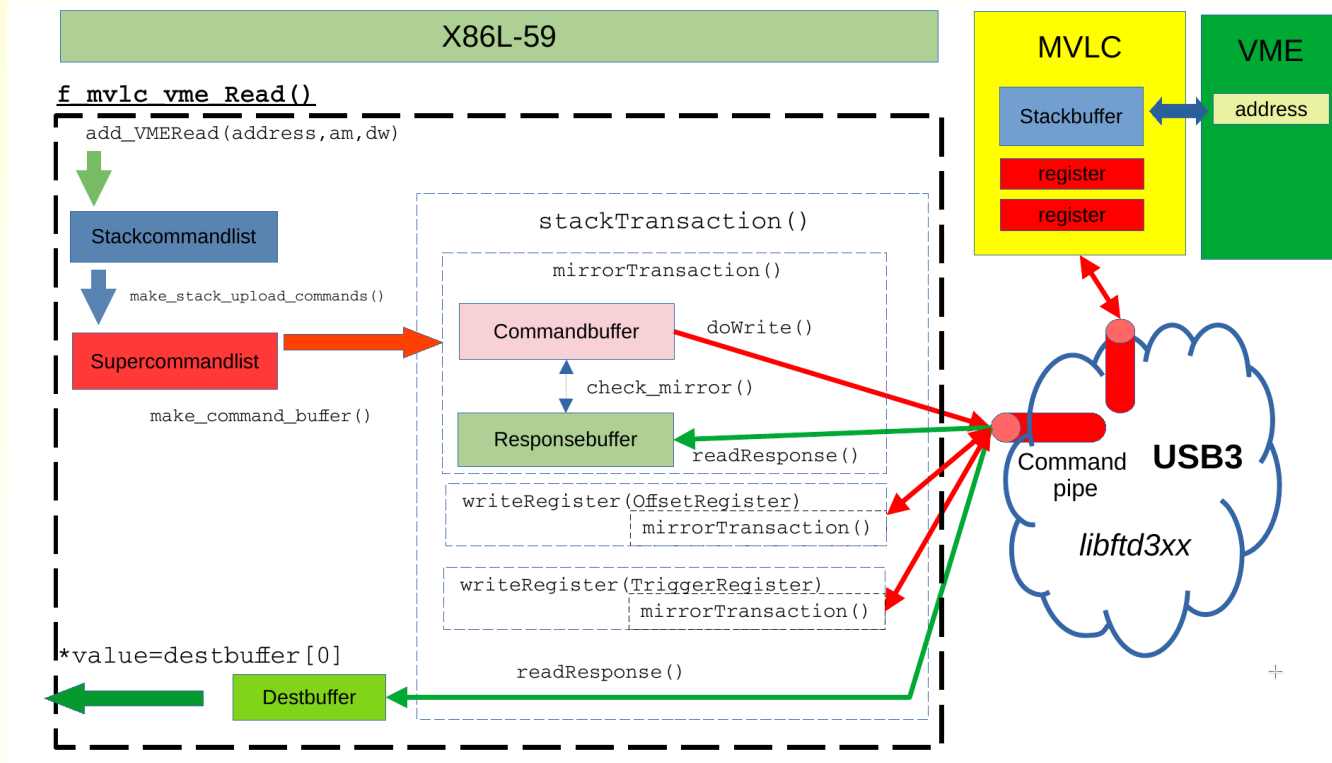
Module Init for module triva7_master (on X86L-114)

```
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) #
15 write A32 D32 0x000C $(65535 - 1300)
16
17 ##### MVLIC setup for TRIVA
18 # Optional signal aliases: map trigger values to MVLIC IRQs (1..15).
19 # If no alias is set the trigger number is directly taken as the IRQ number.
20 # Otherwise if one of the alias register pairs is used all pairs will be active too.
21 # Mapping a trigger value to 0 makes the MVLIC ignore the trigger.
22
23 writeabs A32 D16 0xFFFF7000 14 # TRIVA Trigger 14 (Start)
24 writeabs A32 D16 0xFFFF7002 14 # mapped to MVLIC IRQ 14
25
26 writeabs A32 D16 0xFFFF7004 15 # TRIVA Trigger 15 (Stop)
27 writeabs A32 D16 0xFFFF7006 15 # mapped to MVLIC IRQ 15
28
29 writeabs A32 D16 0xFFFF7008 1 # TRIVA Trigger
30 writeabs A32 D16 0xFFFF700A 8 # mapped to MVLIC IRQ (data_event)
31
32 writeabs A32 D16 0xFFFF700C 2 # TRIVA Trigger
33 writeabs A32 D16 0xFFFF700E 9 # mapped to MVLIC IRQ (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
39 writeabs A32 D16 0xFFFF7014 4
40 writeabs A32 D16 0xFFFF7016 11
41
42 /*
```

Integration of MVLC to MBS

- **Goal: event building of multiple branches with mvlc and non-mvlc readout!**
- To do:
 - *Control of TRIVA with MBS commands* (set trigmod, start acquisition,...)
 - Different readout code for different TRIVA trigger types (mvme + mesytec upgrades! **OK**)
 - *MVLC USB data is received by MBS 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, **OK**)

MVLC software: VME read transaction



MVLC software integration

- Re-implemented mvme C++ functions into **MBS C library** (*lib_mvme.a*)
- **New MBS readout process** *m_read_mvme*
 - Used instead common readout *m_read_meb*
 - Access to **TRIVA registers** via USB/mvme
 - Reads **mvlc data stream** from USB
 - **Formatting of mbs subevents** for pipe
 - Optionally formatting of **VETAR White Rabbit timestamps**
 - !No user defined readout function – **user readout to be defined with mvme tools**
- New **linux kernel module** *mbsmem.ko* for host PC (shared memory pipe)
- New **command line tool** *mec* (pendant to *goc*): VME memory access via USB/mvlc

MVLC with MBS “how to”

1. Configure MVLC with MVME GUI tool.

- It is available as alias “mvme” on all MBS PC hosts for v7.0.
- Example readout stacks for all VME modules are provided by EEL/DAQ group

2. Exit MVME tool. DAQ setup will remain in MVLC FPGA until power down.

3. Startup MBS with settings:

- In setup.usf: MASTER=19
- in startup.scom: m_read_mvme instead m_read_meb.
- Command “START MVLC” to start acquisition.

Summary MVLC with MBS

- All VME modules used at GSI proved to work
- Performance improvement of factor 3 compared with RIO4 readout for a typical FRS setup
- Performance improvement of factor 5 compared with RIO4 for 2eSST readout of CAEN v1742 (150 MB/sec)
- Switching from pure MVME to MBS DAQ in a few seconds
(Use *mvme* to load MVLC sequences for MBS)
- No other hardware changes required when replacing RIO4 with MVLC
- MBS with MVLC controller works as standalone, master or slave TRIVA trigger MBS sub(system)
- No multi-event readout required for performance reasons

White Rabbit timing receivers

Read **FAIR time stamp** (<4 ns resolution) latched at the moment of **accepted trigger** into event data

Used for **synchronization / time sorting** between “free running” (aka streaming readout) branches



Updates for White Rabbit with MBS

- Readout of VETAR: **for IPV, IFC, MVLC** (see next slide!)
- Support of **FAIR timing release *fallout*** [4]:
 - Software installations on MBS X86_64 Linux (etherbone, saftlib) for debian9, debian 11
 - Firmware deployed and tested on PEXARIA and VETAR
- Support of **new time latch unit ECA_TLU**:
 - Old TLU (GSI TM LATCH V2) deprecated; has bugs in latest fallout firmware
 - Direct access readout of ECA_TLU has been demonstrated; fixes problems
 - **Requires initial configuration with higher level saftlib!** (automatically at boot time)

VETAR readout with MVLC

- **MVLC has no OS** => no wishbone.ko, vetar.ko kernel modules! No etherbone libs.
- Must use direct TLU register access
- Configuration of VETAR implemented with stack code in MVME tool => **“kernel driverless” mode (also useful for other platforms...)**
- Readout of VETAR timestamp by default with TRIVA stack frame
- Formatting of “MBS timestamps” from TLU words done in *m_read_mvme* (no *f_user.c* with common VETAR readout code!)

New mass storage interface FSQ

- “**File Storage Queue**” (Thomas Stibor, GSI IT [\[6\]](#))
- **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

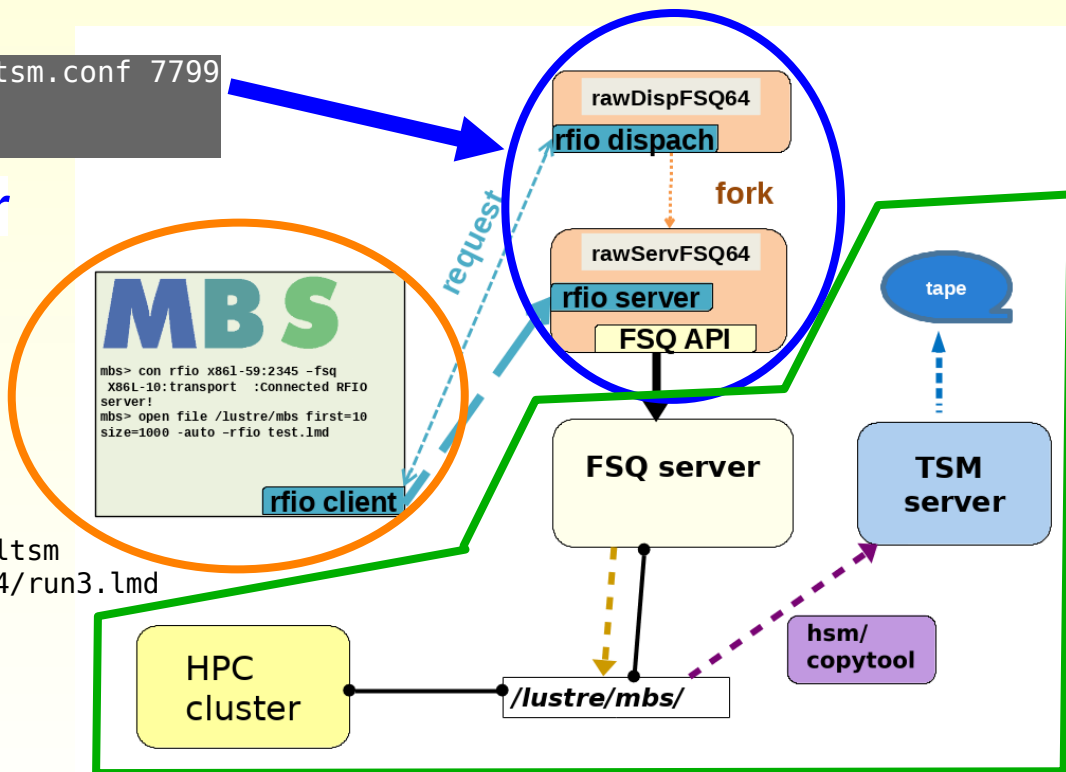
MBS with RFI02FSQ gateway

```
adamczew@x86L-59: rawDispFSQ64 -c ./myltsm.conf 7799
```

experiment's server

MBS event builder

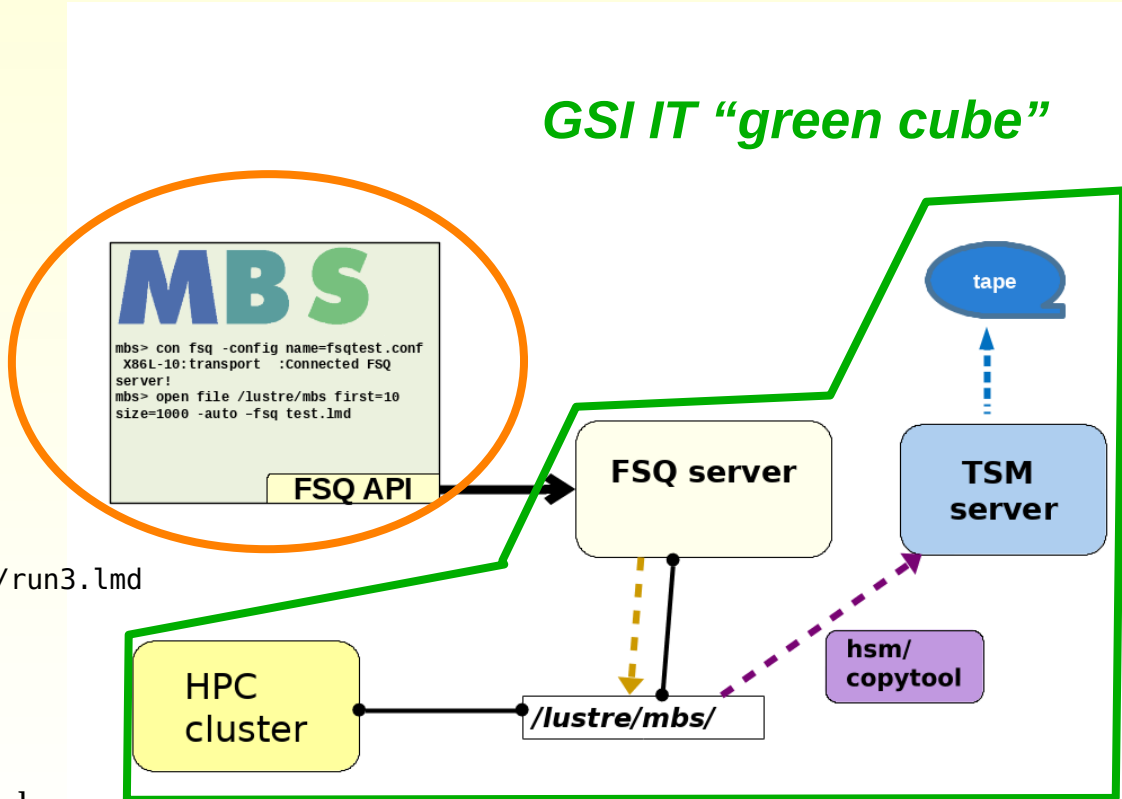
```
MBS> connect rfio <servername >:<port > -ltsm  
MBS> open file -rfio -auto /lustre/exp/mar24/run3.lmd
```



MBS transport to FSQ server

MBS event builder

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



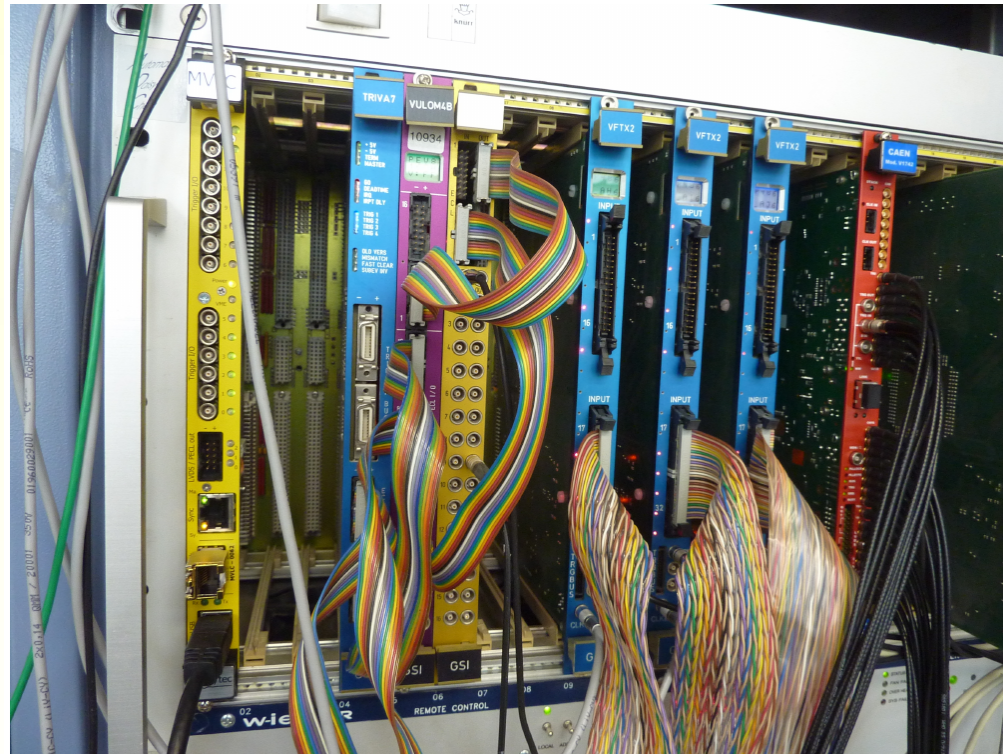
Miscellaneous

- Support of file sizes up to 128GiB
- Improved exit handlers, *f_user_exit()* API
- Fix of *resl* command (failed for long pids on Debian11)
- Several bugfixes (?)

Summary

- **MBS release v7.0 provides many developments since 2016**
- **PCIe X86L- hosts ready for upgrade 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)**

Thank you!



07.03.2023

J.Adamczewski-Musch, N. Kurz

24

References

- [1] IOxOS Technologies, mvme - IFC1211 documentation,
Available at https://www.ioxos.ch/produit/ifc_1211

- [2] mesytec GmbH Co. KG , mvme - mesytec VME Data Acquisition documentation,
Available at <https://www.mesytec.com/products/nuclear-physics/MVLC.html>

- [3] J.Adamczewski-Musch, N.Kurz, S.Linev, Status and developments for DAQ system MBS v6.3,
in: GSI Scientific Report 2014, Available at <http://dx.doi.org//10.15120/GR-2015-1-FG-CS-05>

- [4] D.Beck, A.Hahn, et.al Timing Release – Fallout-v6.2.1,
Available at https://github.com/GSI-CS-CO/bel_projects/releases

- [5] J.Adamczewski-Musch, N.Kurz, S.Linev, GSI Data Acquisition System MBS Release Notes V6.3,
Available at https://www.gsi.de/fileadmin/EE/MBS/gm_mbs_rel_63.pdf

- [6] T.Stibor, Lustre TSM File Storage Queue,
Available at <https://github.com/tstibor/ltsm>

Bonus slides

Command line tool mec

```
X86L-158 adamczew > mec -h
```

```
*****
```

```
m_test_mvme for mesytec mvlc
```

```
v0.16 23-July-2021 by JAM (j.adamczewski@gsi.de)
```

```
*****
```

```
usage: m_test_mvme [-h][-x][-m][-d][-r[-t][-n]]-w[-t][-n]-b|-f
```

```
Options:
```

```
-h                : display this help
-t                : TRIVA mode (address is offset to TRIVA base) (0)
-x                : hex mode (read and display as hex numbers) (0)
-m am            : specify address modifier (0x9)
-d dw            : specify VME datawidth (D32)
-a address       : specify VME address (TRIVA offset if option -t)
-r               : read VME register at address (TRIVA offset if option -t).
-b words         : VME block read of specified words from given address.
                  Default am(0xb) may be changed with -m
-w value         : write value to VME address
-n repeats       : Read or Write repeats (1) number of words consecutively,
                  starting from given address
-f filename.mec  : write values to VME addresses as specified in ASCII
                  configuration file (line format: address value am dw)
```

New MBS commands for MVLC

- START MVLC (replaces START ACQUISITION)
- STOP MVLC (replaces STOP ACQUISITION)
- SHOW MVLC (replaces SHOW TRIG MOD)
- SET TRIG MOD (reimplemented)
- ENABLE/DISABLE TRIG MOD (reimplemented)
- ENABLE/DISABLE IRQ (reimplemented)
- READ MVLC (read from any VME bus address via MVLC controller)
- WRITE MVLC (write to any VME bus address via MVLC controller)

Mass storage file interface

- Old protocol RFIO(gstore) deprecated at GSI
 - => **new system LTSM established in 2019** (Thomas Stibor, GSI IT [\[6\]](#))
- LTSM client requires IBM TSM libraries! Not available for all MBS platforms.
- **Gateway application rawServLTSM** was developed for **MBS v6.3**
 - runs on X86_64 PC with TSM libs only; RFIO as interface from MBS
- Direct connection between MBS and storage system?
- Faster availability of files on /lustre for near online analysis?