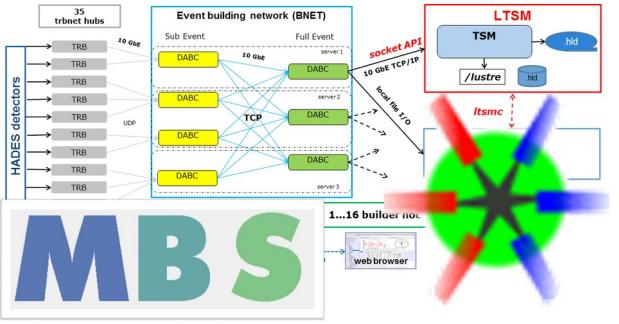
Streaming readout for multi-purpose DAQ system MBS



***Jörn Adamczewski-Musch*, Nikolaus Kurz**

Experiment ELectronics department (EEL),
GSI Helmholtzzentrum f. Schwerionenforschung,
Darmstadt, Germany

BNL streaming readout workshop VII, 16-18 November 2020

Some Facts about MBS

- **General purpose DAQ**
- System scalable:
  from single-crate systems up to
  hierarchically structured **Multi-Branch System (MBS)**
- > **70 systems** installed and running at GSI - **"standard" DAQ** at GSI
- > **150 systems** installed and running outside of GSI
- Entirely written in C; **started in 1993; continuously developed**
- Support for **Linux** and LynxOS
- Support for **VME**, VXI, CAMAC, FASTBUS, PCI and **PCI Express** based systems
- Supported processor platforms: **IPV, RIO4**, RIO3, RIO2, E7 (VME)
  CVC (CAMAC)
  PC (PCI Express, PCI)
- Data Transport via all **address-mapped buses** and **TCP/IP** (10 Mb/s up to 10 Gb/s)

Streaming readout for GSI “standard” DAQ?

See full presentation of Nikolaus Kurz, EEL, GSI at **https://www.gsi.de/fileadmin/EE/MBS/mbs_overview.pdf**

Time Stamp Synchronization of GSI MBS (Multi Branch System) DAQ with White Rabbit:

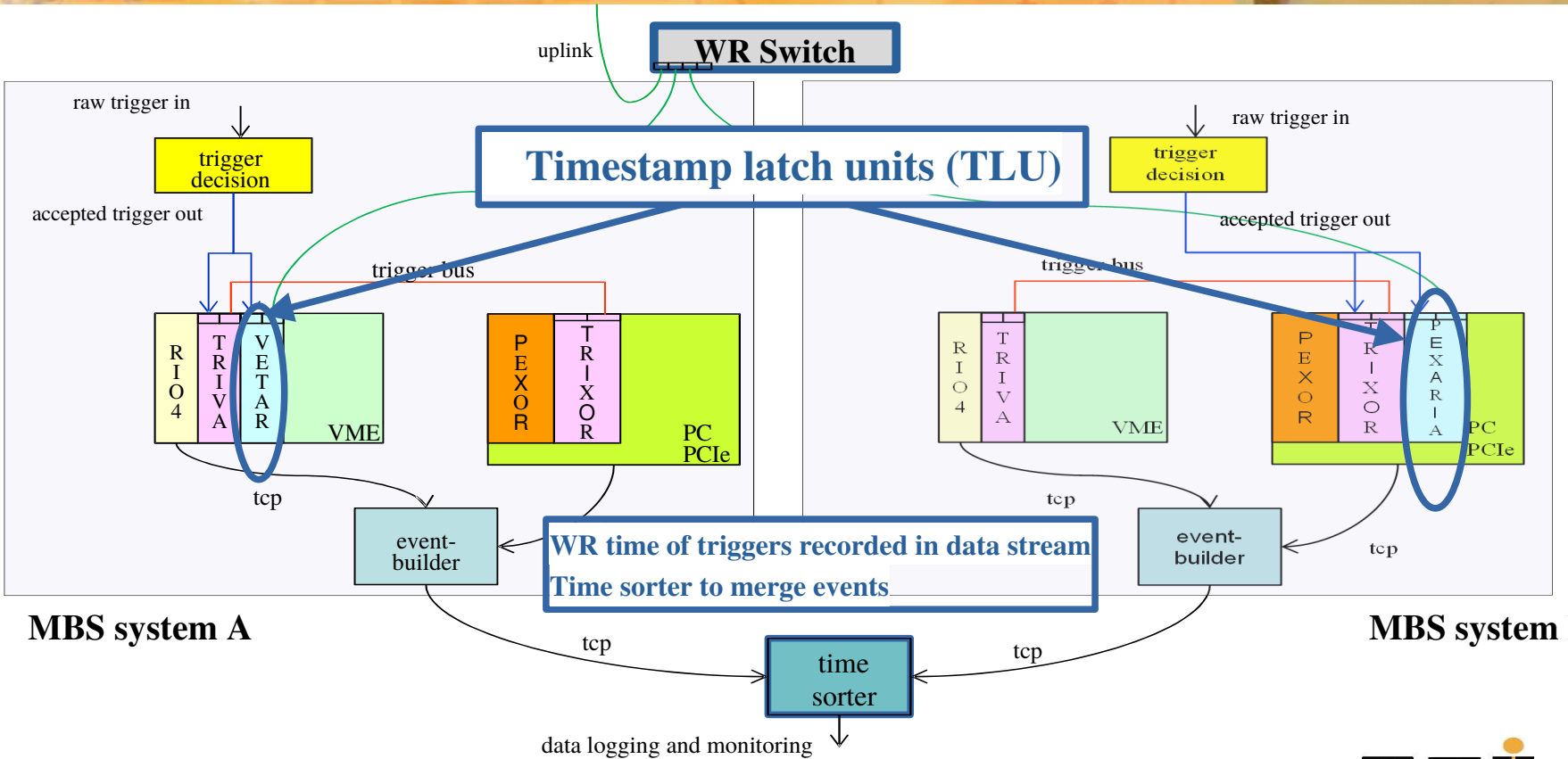
Synchronization of **several globally triggered** MBS systems

Synchronization of **locally triggered (free running) DAQ systems** with globally triggered MBS DAQ

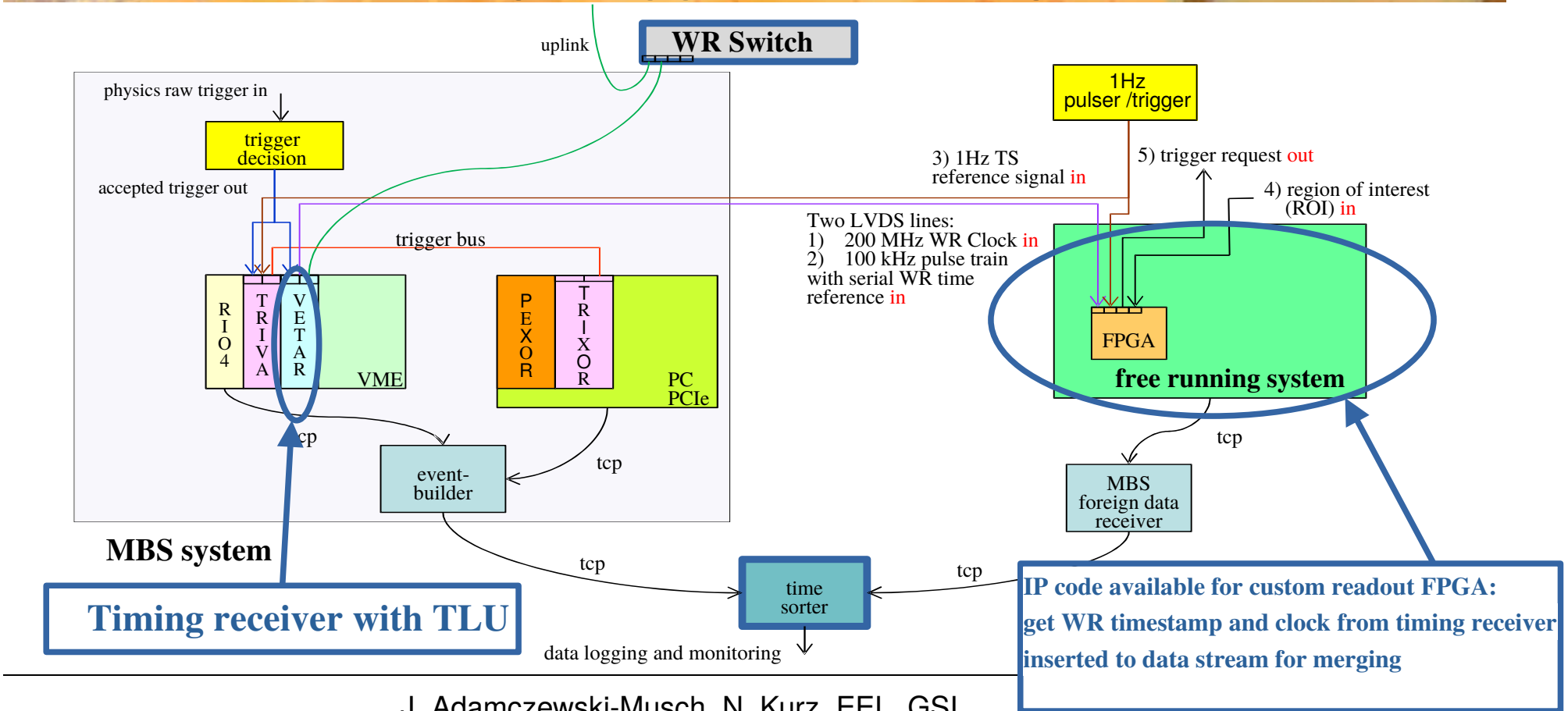
Intended FAIR (NUSTAR) experiments:

R3B Silicon Tracker, AIDA, HYDE, MONSTER, ??

Two Globally Triggered, White Rabbit Synchronized MBS Systems



Synchronization of “Free” Running System with Globally Triggered MBS System



Summary “streaming readout” for MBS DAQ

- **WhiteRabbit** is **general machine timing system for FAIR**, available in all caves
- WhiteRabbit **timing receivers** with TLU **ready for MBS DAQ systems**
- **MBS event time sorting software** is well established
- Allows to combine **independent triggered readout branches**
- Allows to combine **triggered with streaming readout branches**
- Data is always build at least as **“readout trigger” event packets**
- Users may add **own software for physics event selection** later

Thank you!

Bonus slides:

Selected Features of White Rabbit Timing Receivers (WTR)

- 64 bit time stamps with 1 ns units, starting from 1970
- FIFOs for 256 WR time stamps (allows for multi-event readout for globally triggered MBS systems)
- time stamp latch units. Latches actual time stamp with leading edge of TTL input signal. Input signal shall be accepted trigger signal
- 125 and 200 MHz WR controlled high precision LVDS clock outputs
- 100 KHz T0 LVDS pulse train
- WR time transmission between T0 pulses via serial protocol

PEXARIA: 1 ns time stamp granularity
VETAR2: 8 ns (125 MHz native network clock)

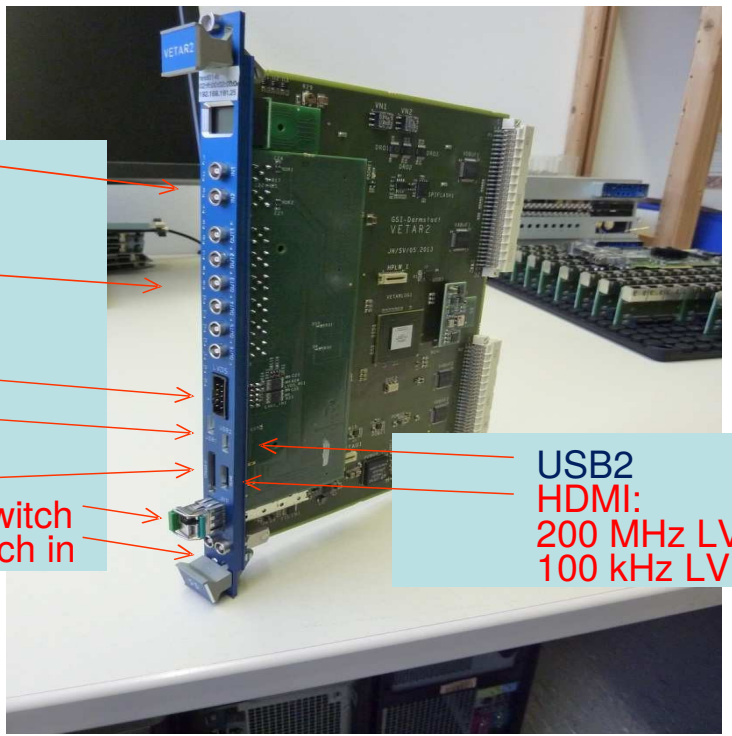
White Rabbit VME Timing Receiver VETAR2

2 general TTL IN

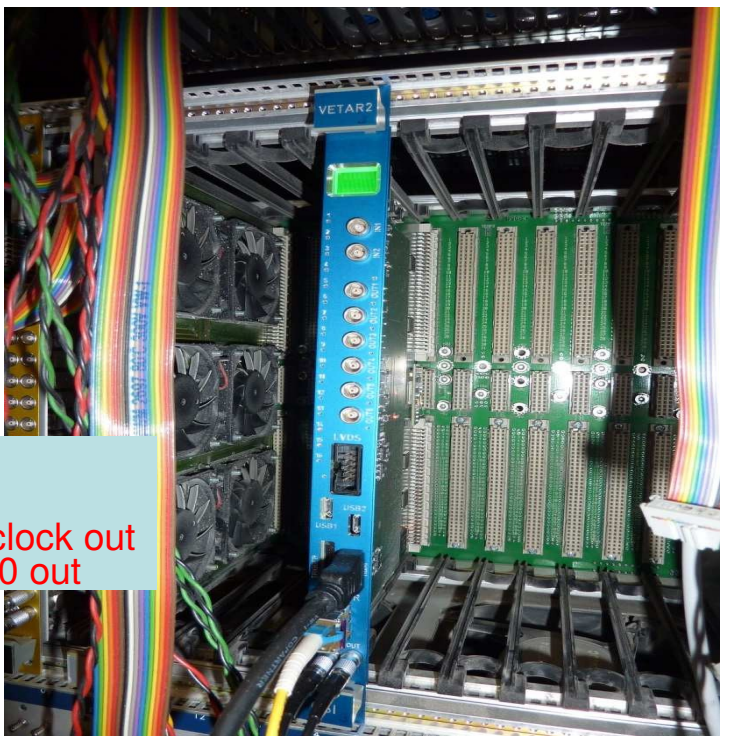
5 general TTL out

LVDS out
USB1

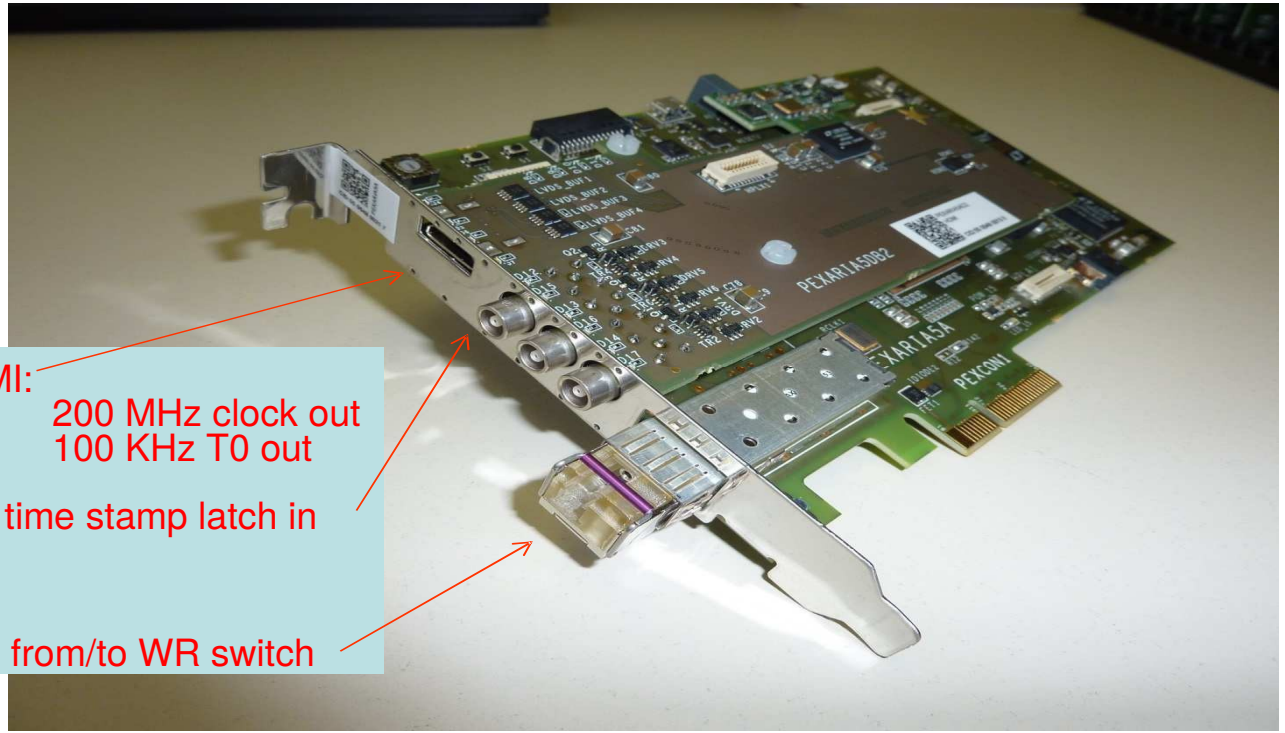
Trigger bus
SFP to/from WR switch
TTL time stamp latch in



USB2
HDMI:
200 MHz LVDS clock out
100 kHz LVDS T0 out



White Rabbit PCI Express Timing Receiver PEXARIA5



Synchronization Inputs/Outputs for "Free" Running Systems

1) 200 MHz: input, mandatory
   - disciplined by WR
   - could be other frequency (50 MHz?) on request
   - basic precision 5 ns (with 200 MHz WR clock). Could be improved with fast FPGA in "free" running system

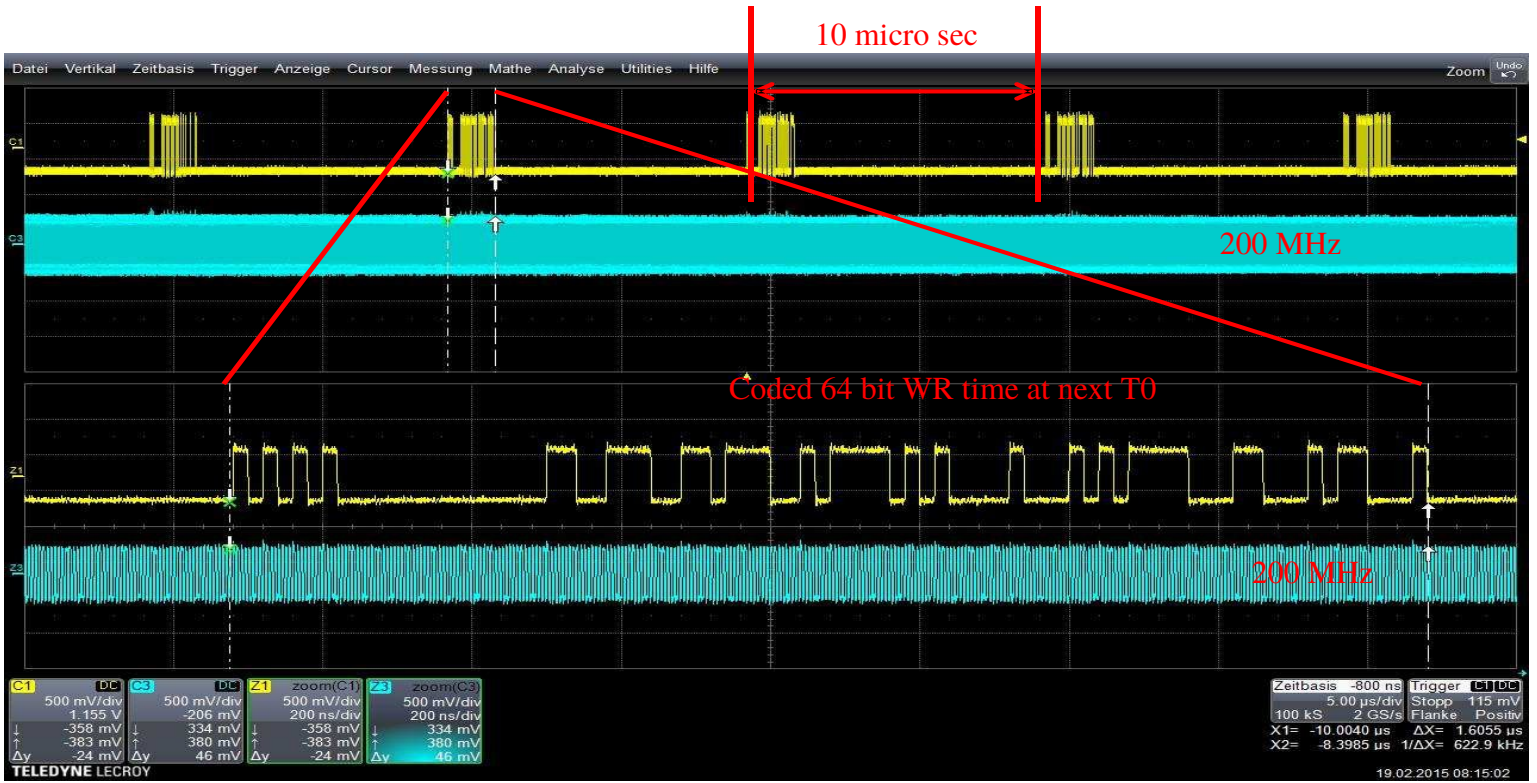
2) 100 KHz T0 + serial time forwarding: input, mandatory
   - decoder and encoder FPGA code available
   - phase stable with 200 MHz WR clock
   - no reset necessary, if new systems couple to a running system

3) 1 Hz reference hit: input, mandatory
   - allows permanent testing of WR time stamp status
   - allows for continuous data flow. important for time sorter

4) ROI (region of interest): input, (important)
   - allows to cut out data outside a programmable time window from free running system
   - ROI could be accepted trigger signal from globally triggered system
   - important, if data rate from unwanted hits (noise, delta electrons) becomes too high

5) TR (trigger request): output, nice to have (depends)
   - to be used by globally triggered systems as input for trigger processing

White Rabbit 100 KHz T0 and - Serial Time Distribution
White Rabbit 200 MHz



Streaming readout for multi-purpose DAQ system MBS

data packets from free running systems:

- free running data acquisition systems sends formatted **MBS sub-events** of hits
- each sub-event is **headed by White Rabbit Time Stamp (WRTS) - 1 ns units, starting from 1970**
- each sub-event contains **data from "many" hits** (MBS container)
- **each hit has TS** of variable size, but significantly smaller than WRTS
- each hit TS must have a **sufficient correlation to full WRTS** header
- **hit data format** inside MBS container has no dependency for time sorting and **can be chosen freely by each detector/sub-system**

MBS subevent header (big endian representation):

```
typedef struct{

          int       l_dlen;             /* Data length +2 in words */

          short     i_subtype;          /* Subtype */

          short     i_type;             /* Type number */

          char      h_control;          /* Processor type code */

          char      h_subcrate;         /* Subcrate number */

          short     i_procid;           /* Processor ID [from setup] */

} s_veshe;
```

White Rabbit full timestamp header (WRTS):

```
          int       sub-system id       /* (32 bits, multiples of 0x100) */

          short     0x03E1              /* (16 bit fixed code)*/

          short     WRTS_L16            /*  WRTS bits 00-15 */

          short     0x04E1              /* (16 bit fixed code)*/

          short     WRTS_M16            /*  WRTS bits 16-31 */

          short     0x05E1              /* (16 bit fixed code)*/

          short     WRTS_H16            /*  WRTS bits 32-47 */

          short     0x06E1              /* (16 bit fixed code)*/

          short     WRTS_X16            /*  WRTS bits 48-61 */
```

Custom hit messages with "local" timestamps (TS, relative to WRTS header)

.....