

VMM3a related firmware activities in 2020 and 2021

ESS work on RD51 SRS components

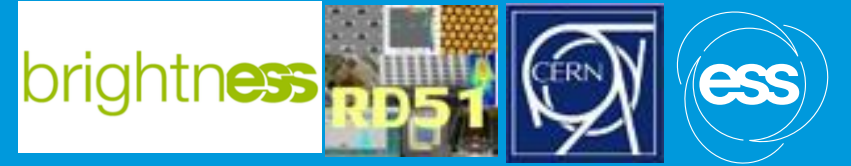
DOROTHEA PFEIFFER (ESS)
STEVEN ALCOCK (ESS)
SERGEI ODINTSOV (UNIVERSITY OF TALIN)

15/02/2021

Overview



1. Tallinn University Inkind project: Cleanup, refactoring and documentation of VMM3a hybrid firmware
2. Integration of RD51 VMM3a hybrid into ESS readout
3. SRS FEC firmware and slow control changes



Tallinn University: Refactoring firmware for RD51 VMM3a hybrid

15/02/2021

Inkind project for NMX

Tallinn University: Refactoring firmware for RD51 VMM3a hybrid

brightness



- Project between the electronics department of Tallinn University and ESS NMX, signed just before Xmas in 2019
- Aim of the project was to clean up the hybrid firmware, refactor and document it
- Person chosen: Sergei Odintsov, Electronics Engineer, FPGA expert, at the moment finishing his PhD
- Original plan was, Sergei comes to CERN for a few weeks to get familiar with the SRS, and works together with Marek
- But then came Corona...
- We did not have a spare SRS system to send to Tallinn, therefore we set up a PC so that Sergei could have remote access to an SRS system via Teamviewer

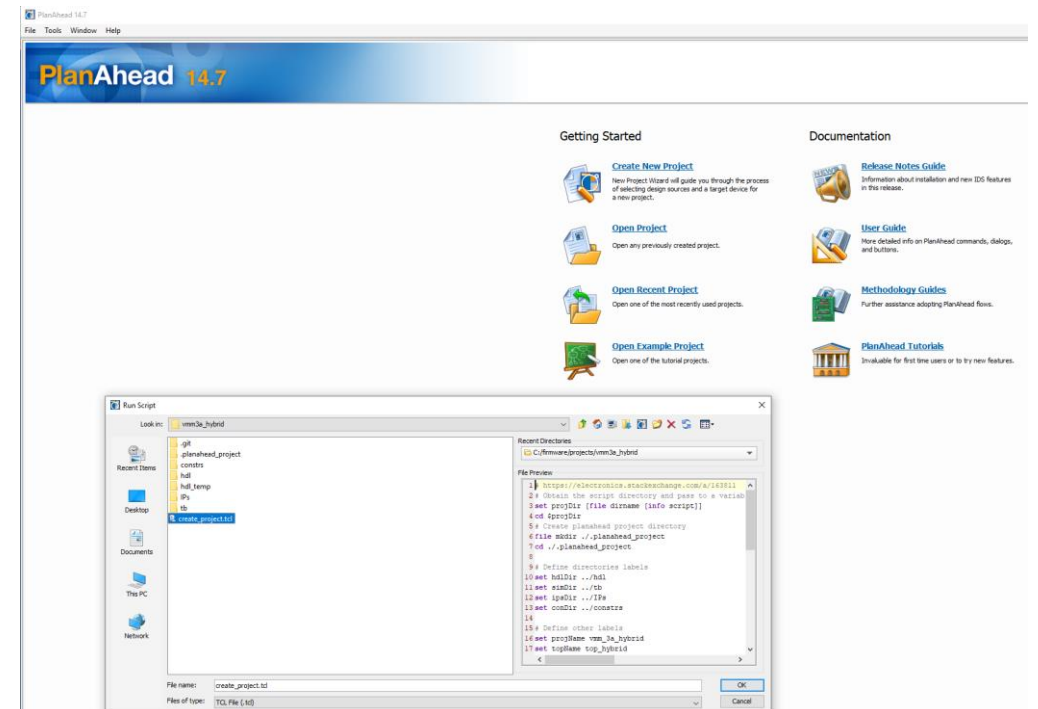
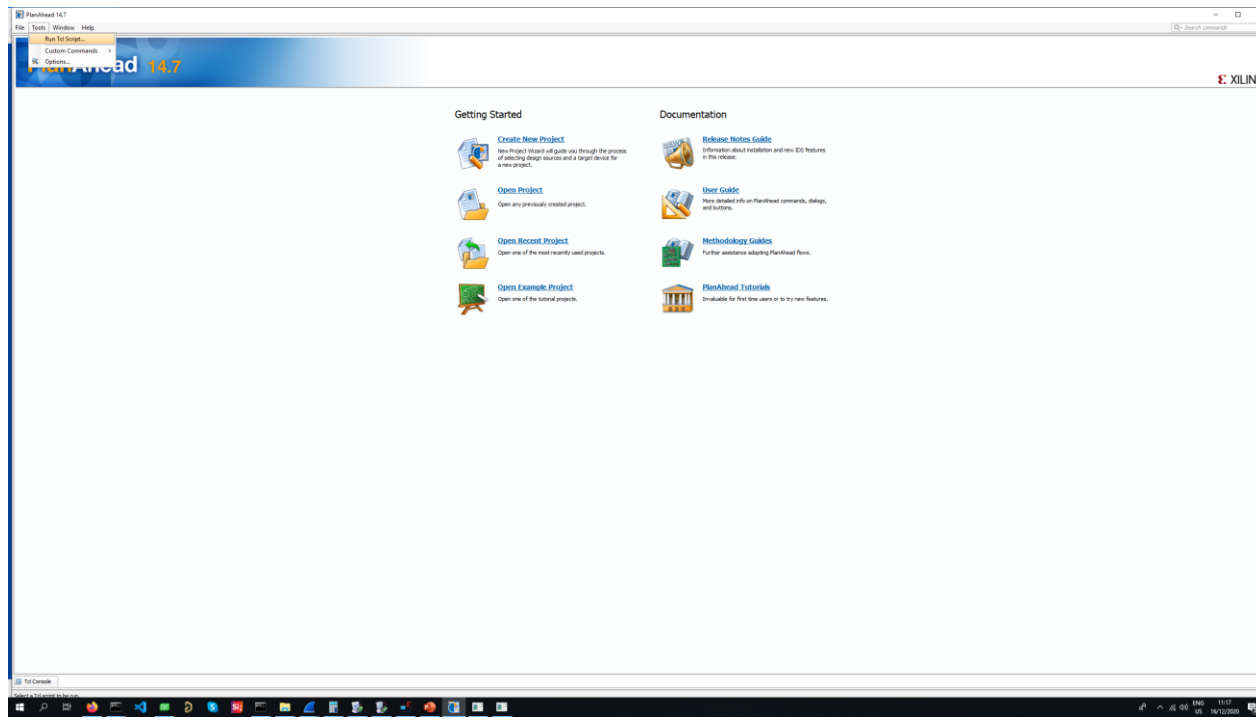
RD51 VMM3a hybrid firmware

brightness

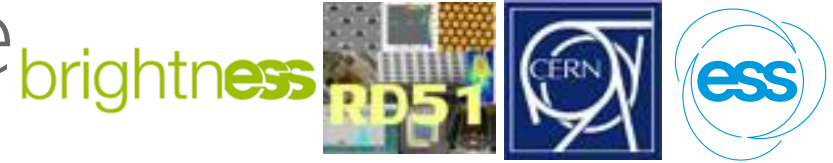


Change of Xilinx tool

- Xilinx Vivado (tool for series 7), the successor of the much hated ISE, was build using PlanAhead technology
- PlanAhead can be used instead of ISE for Virtex 6 and Spartan 6
- Repo does not store numerous ISE project files anymore, or even PlanAhead project, just tcl script (fun fact: Marek, Sergei and Steven came up in parallel with the same solution)
- TCL script is executed in PlanAhead and creates the project



RD51 VMM3a hybrid firmware



Change of repository logic

- create_project.tcl script is very simple, describes which directories and files are added to the project
- Strategies for synthesis and implementation can be added, e.g. set_property strategy TimingWithIOBPacking [get_runs synth_1]
- Since project is generated and not stored in repo, no more problems and errors due to temporary ISE project files (my computer was driving me insane, claiming insufficient rights all the time)
- Repo contains only vhdl, verilog files and xci files for IP cores
- Repo structure follows “quasi” standard: vhdl/Verilog files are in hdl, xci files for IP cores in IPs, ucf constraints files in constrs, and test benches in tb
- Sub folders used in hdl to create logical structure of the source code

European Spallation Source / Detector Group TTU In-Kind / vmm3a_hybrid

vmm3a_hybrid

Sonar for Bitbucket failed
Access denied. You must have write or admin access.

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to your repository.

master Files Filter files

Name	Size	Last commit
/		
IPs		2 days ago
constrs		6 hours ago
hdl		7 hours ago
tb		yesterday
.gitignore	131 B	2020-07-13
README.md	565 B	2020-07-03
create_project.tcl	1.34 KB	2020-11-09

European Spallation Source / Detector Group TTU In-Kind / vmm3a_hybrid

create_project.tcl

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to your repository.

Source master 1f946bb Full commit

```
vmm3a_hybrid / create_project.tcl
1 # https://electronics.stackexchange.com/a/163811
2 # Obtain the script directory and pass to a variable
3 set projDir [file dirname [info script]]
4 cd $projDir
5 # Create planAhead project directory
6 file mkdir ../planAhead_project
7 cd ../planAhead_project
8
9 # Define directories labels
10 set hdlDir ../hdl
11 set simDir ../tb
12 set ipDir ../IPs
13 set conDir ../constrs
14
15 # Define other labels
16 set projName vmm_3a_hybrid
17 set topName top_hybrid
18 set device xc6slx16-2csq225
19
20 # Create project in prj directory
21 create_project -force $projName prj -part $device
22
23 # Add simulation files from sim directory and disable synthesis property
24 # Do it before adding any other source!!
25 add_files -fileset sim_1 $simDir
26 set_property used_in_synthesis false [ get_files -of_objects [get_filesets sim_1] -filter {FILE_TYPE == VHDL || FILE_TYPE == Verilog} ]
27
28 # Add HDL sources
29 add_files $hdlDir
30
31 # Add IP cores
32 add_files $ipDir/bram_configVMM3reg/bram_configVMM3reg.xci
33 add_files $ipDir/dfifo40/dfifo40.xci
34
35 # Add constraints
36 add_files -fileset constrs_1 $conDir
37
38 # Set VHDL libraries
39 #set_property library awesome_lib [get_files $hdlDir/awesome_library/*]
40
41 # Set top module
42 set_property top $topName [current_fileset]
43
44 make_wrapper -files [get_files *.xmp] -top -fileset [get_filesets sources_1] -import
45 set_property -name {steps.bitgen.args.More Options} -value {-g INIT_9K:Yes} -objects [get_runs impl_1]
```

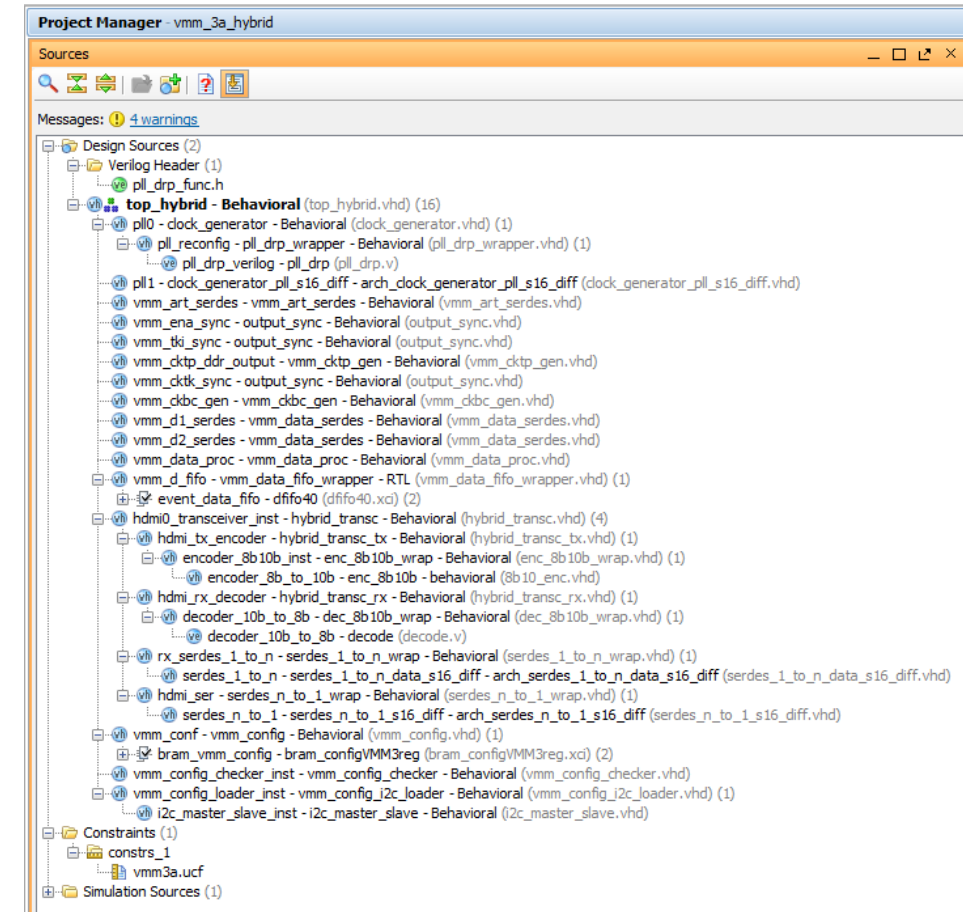
RD51 VMM3a hybrid firmware

brightness



Change of unit names and cleaning of code

- Unused modules removed
- Unused code in modules removed
- Units were renamed so that names are consistent
- Code structured in three major parts, PLLs, HDMI transceivers and VMM



RD51 VMM3a hybrid firmware

brightness



Coding standard

- Naming of signals according to following coding standard
 - Constants: capital letters starting with C_
 - Wires connected to output, i_ or o_ depending on direction
 - Types: t_
 - Internal wires: w_
 - Registers: f_
 - Outputs of combinatorial logic: k_
- Components are changed to entities
- Use of generics and constants instead of numbers
- Reformatting of code
- Correction of sensitivity lists of processes, replacing clk'event by rising_edge
- Explanatory comments added
- Structural changes to simplify code

```
67 architecture Behavioral of hybrid_transc is
68   -- C_.* = constant
69   constant C_RST_SYNC_SHIFT_REG_LEN      : natural := 2;
70   constant C_VMM_DATA_FIFO_CLEAR_TTC_POS : natural := 3;
71   constant C_VMM_BCR_TTC_POS             : natural := 4;
72   constant C_VMM_CKTP_GENERATOR_EN_TTC_POS : natural := 0;
73   -- o_.*_i = wire connected to output
74   signal o_hdmi_tx_i                      : std_logic_vector(G_VMM_NUM - 1 downto 0);
75   signal o_hdmi_rx_config_data_i         : std_logic_vector(work.hybrid_transc_rx_pkg.C_DATA_WORD_LEN - 1 downto 0);
76   signal o_hdmi_rx_config_addr_i        : std_logic_vector(work.hybrid_transc_rx_pkg.C_DATA_WORD_LEN - 1 downto 0);
77   signal o_hdmi_rx_config_data_valid_i  : std_logic;
78   signal o_vmm_data_fifo_rd_en_i        : std_logic_vector(G_VMM_NUM - 1 downto 0);
79   signal o_vmm_data_fifo_clear_i        : std_logic;
80   signal o_hdmi_link_state_acq_i        : std_logic;
81   signal o_vmm_tki_i                    : std_logic_vector(G_VMM_NUM - 1 downto 0);
82   signal o_vmm_bcr_i                    : std_logic;
83   signal o_vmm_cktp_generator_en_i      : std_logic;
84   signal o_hdmi_8b10b_decoding_error_i  : std_logic;
85   -- t_.* = type
86   type t_8b10b_encoder_output_array is array (natural range <>) of std_logic_vector(C_8B10B_ENCODER_OUTPUT_LEN - 1 downto 0);
87   -- w_.* = wire
88   signal w_rst_sync                      : std_logic;
89   signal w_tx_data_10b                   : t_8b10b_encoder_output_array(1 downto 0);
90   signal w_reset_phy                     : std_logic;
91   signal w_rx_data_bitslip               : std_logic;
92   signal w_ttc                           : std_logic_vector(work.hybrid_transc_rx_pkg.C_DATA_WORD_LEN - 1 downto 0);
93   signal w_ttc_valid                     : std_logic;
94   signal w_rx_data_10b                   : std_logic_vector(C_8B10B_DECODER_INPUT_LEN - 1 downto 0);
95   signal w_8b10b_code_err                : std_logic;
96   signal w_8b10b_disp_err                : std_logic;
97   signal w_rx_link_state_init            : std_logic;
98   signal w_rx_link_state_link            : std_logic;
99   signal w_rx_link_state_idle            : std_logic;
100  signal w_rx_link_state_acq              : std_logic;
101  signal w_rx_serdes_init_done            : std_logic;
102  -- f_.* = register
103  signal f_link_state_acq                 : std_logic;
104  signal f_f_link_state_acq               : std_logic;
105  signal f_vmm_tki                        : std_logic_vector(G_VMM_NUM - 1 downto 0);
106  signal f_rx_serdes_toggle               : std_logic;
107  signal f_rst_sync                       : std_logic_vector(C_RST_SYNC_SHIFT_REG_LEN - 1 downto 0);
108  attribute ASYNC_REG                     : string;
109  attribute ASYNC_REG of f_rst_sync : signal is "TRUE";
110  -- k_.* = combinational logic output
111  signal k_reset_serdes                   : std_logic;
```

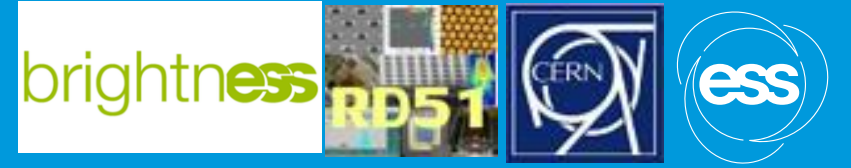

RD51 VMM3a hybrid firmware

brightness



Status summary and future

- Refactoring finished, documentation and writing of test benches still ongoing
- Private bitbucket.com project
https://bitbucket.org/europeanspallationsource/vmm3a_hybrid
- Access can be granted to anybody, not limited to people with CERN or ESS email address
- Project based on latest commit 5c5eb939 from Marek to gitlab.com (02.09.2020 - firmware flashed to new hybrids)
- Firmware will now be ported to Spartan7
- After porting, ESS firmware will evolve in different direction : No ART, different BC clock and data clock (ESS facility clock 88 MHz, hence BCCKL 44 MHz, SERDES for HDMI 440 MHz, CKDT 176 MHz DDR)

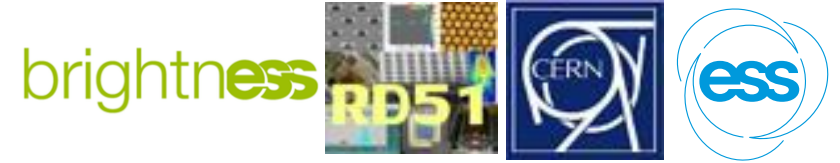


Integration of RD51 VMM3a hybrid into ESS Readout

15/02/2021

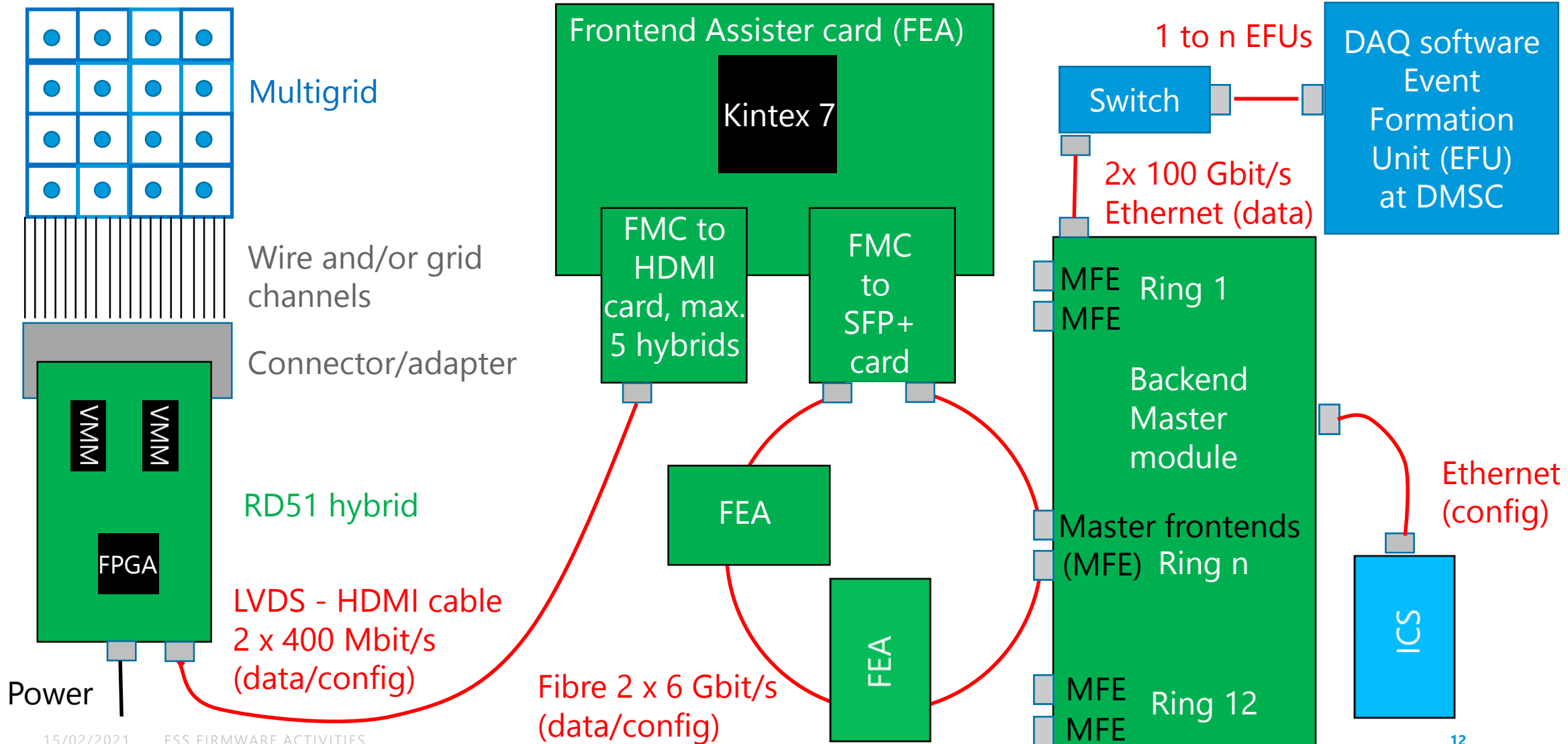
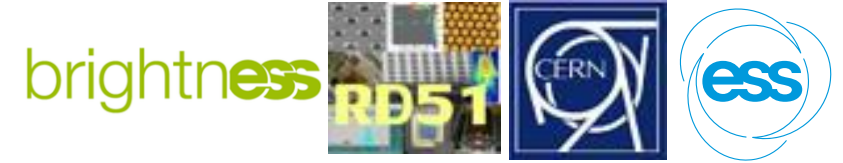
RD51 VMM3a hybrid at ESS

Integration into the ESS readout



- Since 2018 ESS has shown that VMM3a can also be used for non-MPGD gaseous detectors like Multigrid and Multiblade (analog and digital data taken)
- Multigrid is basically grids of single wires, Multiblade is a cathode strip chamber
- VMM3a is the electronics choice to read out the NMX (GEM), CSPEC (Multigrid), ESTIA, TREX and FREIA (all Multiblade) detectors
- Successful Multigrid detector review in August, reviewers agree with the choice of RD51 VMM3a hybrid as front end
- Only worry: Limitation of ≤ 4 Mhits/s per channel might be too low if ESS is operated at full power, in cases of very high instantaneous rates on grid, since a grid has a surface area of several cm^2

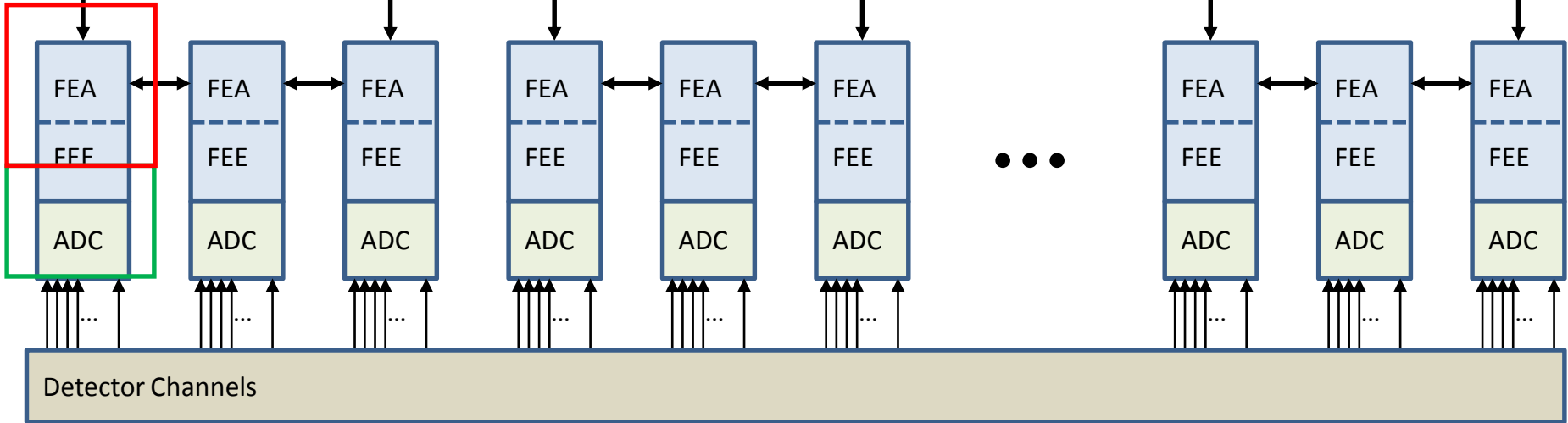
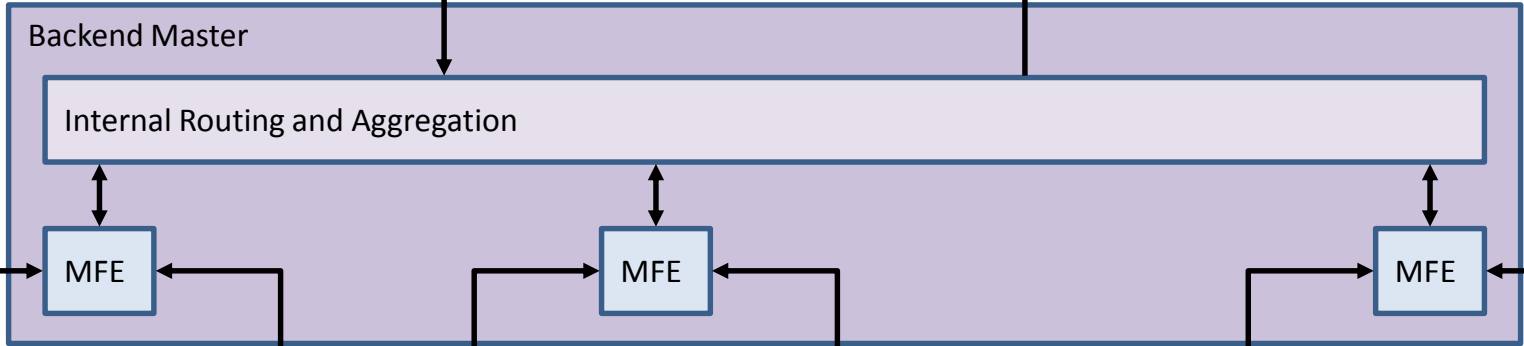
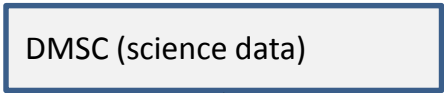
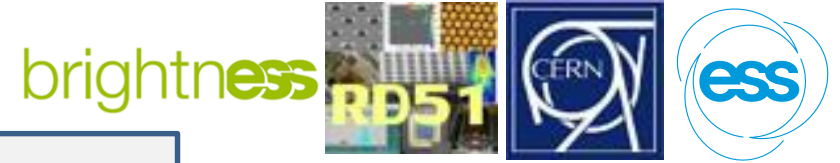
Readout chain Multigrid



ESS Readout

Schematic drawing

Steven Alcock, Detector Group, 2nd July 2019



Assister and adapter

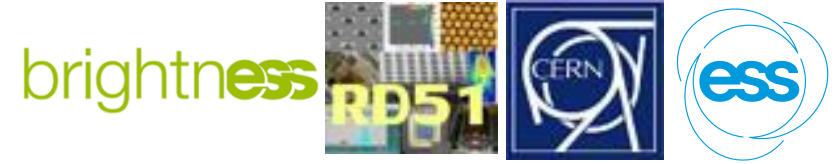
RD51 hybrid

Readout Architecture is described in BrightnESS Deliverable D4.1:
<https://dx.doi.org/10.17199/BRIGHTNESS.D4.1>

FEA for RD51 hybrid

Integration of RD51 hybrid into ESS readout

- Adapter card connects two RD51 hybrids to the FMC connector of the Kintex KC705 evaluation board
- Upgraded version will contain 5 HDMI ports
- RD51 SRS FEC Virtex6 firmware ported to FEA Kintex 7
- Substantial changes: E.g. new ethernet MAC since IP could not be ported
- No problems with FIFO IP
- Care needed with Xilinx components like SERDES and DELAY



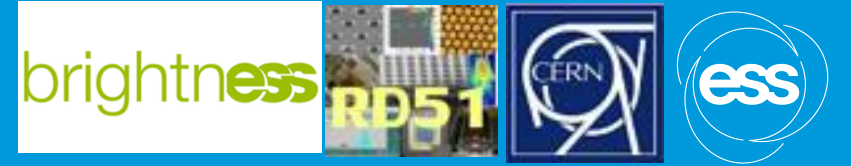
FEA for RD51 hybrid

Integration of RD51 hybrid into ESS readout



- Data transmission via UDP and configuration of VMMs operational
- Xilinx KCU 705 has now all functionalities of the RD51 SRS FEC that are compatible with the hardware of the evaluation board, and can be used with the slow control and ESS DAQ
- Now changing the data format to the ESS data format, and transfer data to the well established assister firmware so that the RD51 hybrid can be used with the full ESS readout (rings, backend)

The screenshot displays the VMM Slow Control software interface. On the left, there are control panels for 'FEC 1' (General and Advanced settings), 'ACCQ' (Global ACCQ, Pulser, External, ACC On/Off), and 'FEC Response' (version, FEC IP, DAQ destination IP). The main window shows a network traffic capture for 'srsvmm' on the 'Ethernet' interface. The capture table lists packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Below the table, a detailed view of a packet shows hit information: '> Hit: 49, offset: 18, vmmID: 0, ch: 31, bcid: 101, tdc: 92, adc: 200, over thr: 1'. The status bar at the bottom indicates 'SRSVMM Protocol (srsvmm), 8,932 bytes' and 'Packets: 100 - Displayed: 100 (100.0%) - Dropped: 0 (0.0%) - Profile: Default'. The Windows taskbar at the bottom shows the date as 15/09/2020 and time as 15:54.

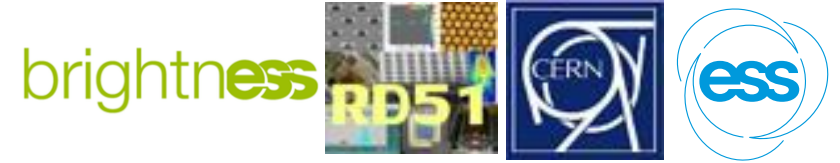


SRS FEC firmware and slow control changes

15/02/2021

Slow control

Cleaning up and changes



- Lucian created very nice new repo <https://gitlab.cern.ch/rd51-slow-control/vmm3c>
- Major clean-up, restructuring and adding of new features
- Hard reset for all VMMs, copy of channel settings to all VMMs, new calibration features
- For fitting of data (linear and non-linear regression), integration of alglib (cross-platform numerical analysis and data processing library) <https://www.alglib.net/>
- In ESS branch work in progress, implementation of new features, in master branch more stable version

The screenshot shows the 'VMM3 - SRS DCS new' software interface. It has a top menu bar with 'VMM Slow Control', 'Calibration', 'Logging', and 'Testing'. The main area is divided into several panels: 'FEC 1' with IP address settings; 'Hybrids' with checkboxes for 1-8; 'Acquisition/Test pulse' with various timing and format options; 'ACQ for all FECs' with 'ACQ On' and 'ACQ Off' buttons; 'Config file' with 'Load' and 'Save' buttons; 'Hybrid 1' and 'Hybrid 2' sections with 'VMM' checkboxes and 'Position' dropdowns; 'VMM 1' and 'VMM 2' sections with 'General Settings' and 'Advanced Settings' tabs; 'ADC' settings including 'ADCs on/off' and '8-bit Conv. Mode'; 'Dual Clock' settings; 'Threshold DAC' and 'Test Pulse DAC' settings; and 'All VMMs: Settings and reset' with 'Set global settings', 'Set channel settings', 'Hard reset VMM', and 'Hard reset all' buttons. On the right, there is a 'Channel Settings' table with columns for SC, SL, ST, STH, SM, SD, SZ2010b, SZ208b, and SZ206b, and rows for channels 0-33.

FEC firmware

Repo, project and timing errors



- Change to Planahead project like hybrid firmware
- Removal of unused modules, cleaning up code, cleaning up constraints file
- Code not yet refactored like hybrid firmware
- Aim of changes: reliable and fast continuous mode without timing errors
- Main sources of timing errors (timing score of around 40000):
 - Triggered mode (not correctly working anyway, compared non-gray encoded FEC trigger counter with gray encoded VMM BCID)
 - FIFO logic in vmm3unit.vhd
 - Due to use of DDR3 memory, logic that determines the next VMM FIFO to read is run at 200 MHz, combinatorial logic too slow to meet timing requirements

Name	From	To	Total Delay	Logic Delay	Net %	Stages	Source Clock
TS_0320 = PERIOD TNSOP "TNS_0320" 3 ns HSPD 50%							
Path 1	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.186	3.197	48.3	5	0.000
Path 2	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.158	3.197	48.1	5	0.000
Path 3	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.134	3.197	47.9	5	0.000
Path 4	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.086	3.192	47.6	5	0.000
Path 5	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.035	3.197	47.5	5	0.000
Path 6	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.021	3.197	46.9	5	0.000
Path 7	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	6.001	3.192	46.8	5	0.000
Path 8	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.995	3.192	46.8	5	0.000
Path 9	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.990	3.192	46.7	5	0.000
Path 10	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.986	3.192	46.7	5	0.000
Path 11	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.999	3.197	46.4	5	0.000
Path 12	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.906	3.197	45.9	5	0.000
Path 13	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.900	3.192	45.8	5	0.000
Path 14	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.815	3.169	45.5	5	0.000
Path 15	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.782	3.194	44.8	5	0.000
Path 16	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.781	3.194	44.8	5	0.000
Path 17	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.762	3.188	44.7	5	0.000
Path 18	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.744	3.175	44.7	5	0.000
Path 19	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.730	3.197	44.4	5	0.000
Path 20	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.757	3.172	44.9	5	0.000
Path 21	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.755	3.172	44.9	5	0.000
Path 22	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.700	3.175	44.3	5	0.000
Path 23	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.686	3.194	43.8	5	0.000
Path 24	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.678	3.194	43.7	5	0.000
Path 25	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.674	3.192	43.7	5	0.000
Path 26	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.661	3.172	44.0	5	0.000
Path 27	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.640	3.194	43.4	5	0.000
Path 28	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.639	3.172	43.8	5	0.000
Path 29	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.653	3.188	43.4	5	0.000
Path 30	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	5.632	3.175	43.6	5	0.000
Path 31	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.449	0.435	70.0	2	0.000
Path 32	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.448	0.437	69.8	2	0.000
Path 33	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.430	0.428	70.5	2	0.000
Path 34	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.398	0.352	74.8	2	0.000
Path 35	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.385	0.382	72.4	2	0.000
Path 36	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.381	0.437	68.4	2	0.000
Path 37	..._f60A00net_ff_..._d.cb.pamb000	..._f60A00net_ff_..._d.cb.pamb000	1.372	0.437	68.1	2	0.000

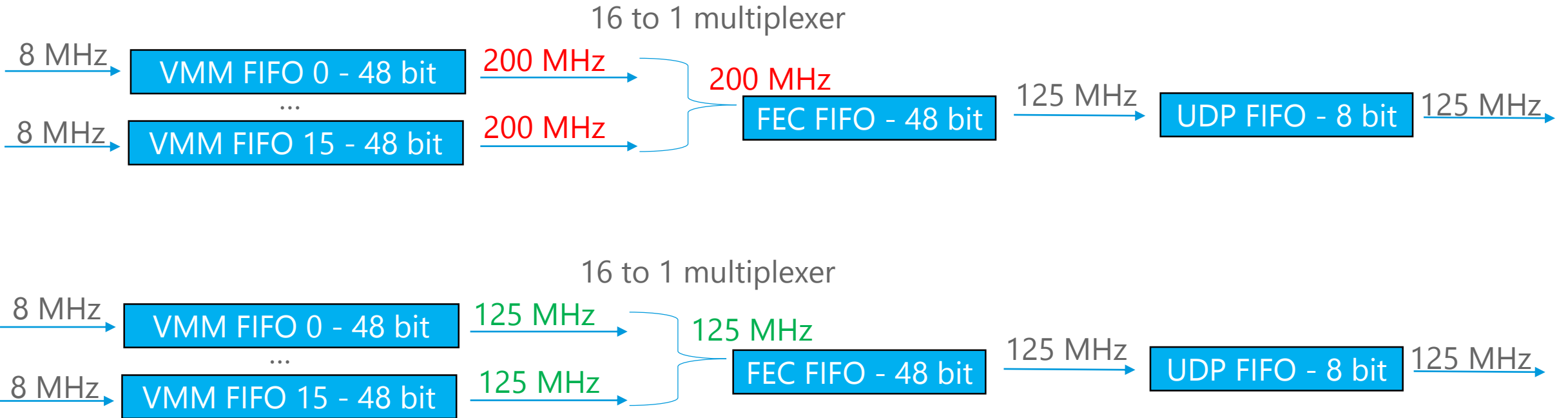
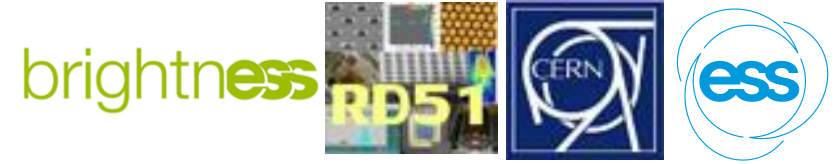
```

831 -- check if the pointed vmm_fifo has data and the connected fifo is full or not
832 vmm3_rden_gen : FOR i IN 0 TO 15 GENERATE
833   PROCESS (nextvmmpointer, fec_fifo_pgfull, vmm_fifo_emptyn)
834     BEGIN
835       IF nextvmmpointer = i THEN
836         IF vmm_fifo_emptyn(i) = '1' THEN
837           IF fec_fifo_pgfull = '0' THEN
838             vmm_fifo_rden(i) <= '1';
839           ELSE
840             vmm_fifo_rden(i) <= '0';
841           END IF;
842         ELSE
843           vmm_fifo_rden(i) <= '0';
844         END IF;
845       ELSE
846         vmm_fifo_rden(i) <= '0';
847       END IF;
848     END PROCESS;
849 END GENERATE;

850
851 -- look for a nonempty vmm_fifo
852 vmm_fifo_emptyn <= cfg_chmask(15 DOWNT0 0) AND (NOT vmm_fifo_empty);
853
854 PROCESS (clk200)
855   VARIABLE var1 : INTEGER RANGE 0 TO 15 := 0;
856   VARIABLE var2 : INTEGER RANGE 0 TO 15 := 0;
857 BEGIN
858   var1 := 0;
859   var2 := 0;
860   IF rising_edge(clk200) THEN
861     FOR i IN 0 TO 15 LOOP
862       IF vmm_fifo_emptyn(15-i) = '1' THEN
863         var1 := 15 - i;
864         IF 15 - i > nextvmmpointer THEN
865           var2 := 15 - i;
866         END IF;
867       END IF;
868     END LOOP;
869     IF var2 > var1 THEN
870       nextvmmpointer <= var2;
871     ELSE
872       nextvmmpointer <= var1;
873     END IF;
874   END IF;
875 END PROCESS;
    
```

FEC firmware

VMM3unit.vhd – situation **before** and **after** changes

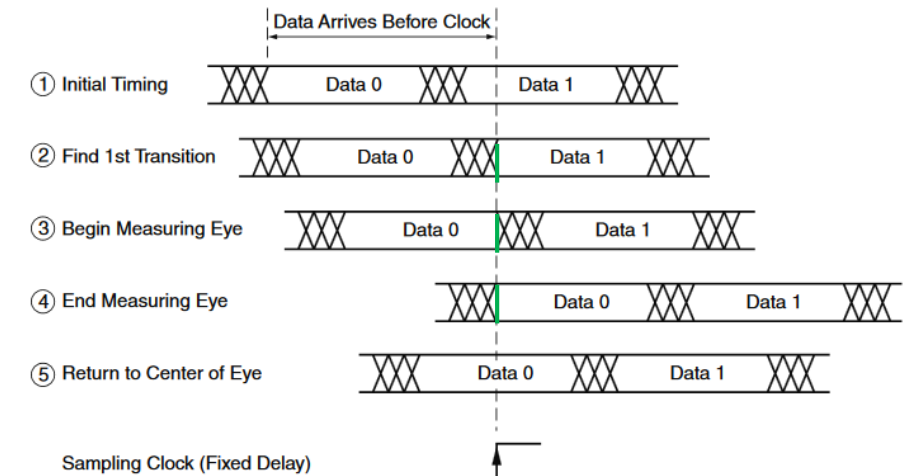


- Bottleneck is UDP, 8bit send with 125 MHz (1 Gbits/s ethernet)
- Reading of the 48 bit VMM FIFOs with 200 MHz does not make sense, choosing VMM FIFO to read and write to FEC FIFO does not work at 200 MHz (logic too slow)
- Performance stays the same, if VMM FIFOs are read with 125 MHz, and FEC FIFO is written with 125 MHz, timing is met

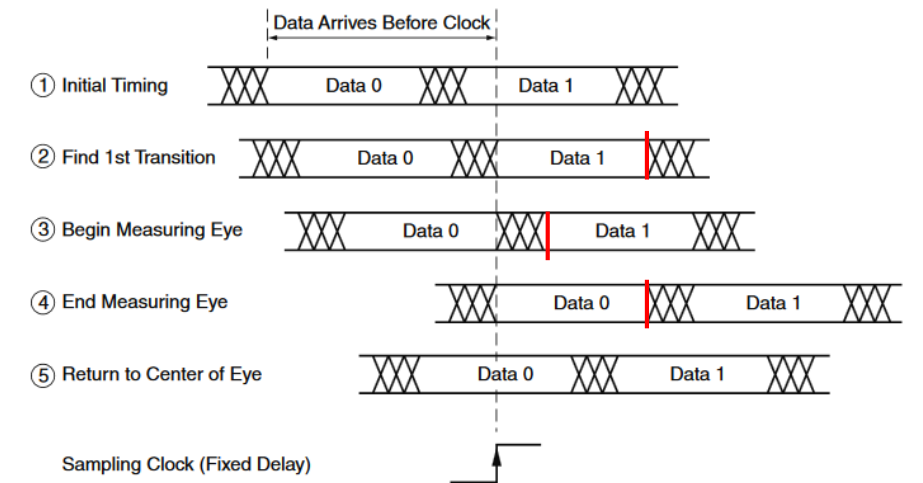
FEC firmware

Link status

- HDMI SERDES use master and slave ISERDES_NODELAY together with IODELAY
- IODELAY has 64 taps of 78 ps that can shift the data (tap 64 is identical to tap 0 again)
- ISERDES can do bitslip to rearrange the 10 bits (after 10 bitslips one arrives again at the original word)
- Aim of alignment: To be in the center of the eye and receive correct 8b/10b control word from Spartan 6 on hybrid (link control word)
- Originally FEC was using bit align machine of Xilinx
https://www.xilinx.com/support/documentation/application_notes/xapp855.pdf
- Problem: If after finding the first transition, the second transition that is found is not on the other side of the window, but just the end of the same transition as the first, the center of the eye is not found
- Therefore Xilinx released update
<https://www.xilinx.com/support/answers/38672.html>
- Now user has to set a parameter that specifies the minimum width of the eye, to avoid the identification of a transition as eye
- With new DVM card, XAPP855 was unstable, thus successfully implemented update in spring 2020



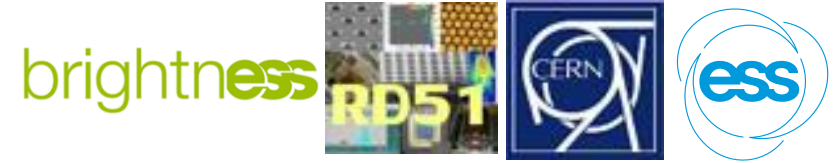
Good: Correct transitions identified



Bad: Second transition not start of eye

FEC firmware

Link status



- When using the CTF card, we still had problems to obtain a stable link status 4 (probably the minimum eye width parameter would have to be adapted per channel and would be different for use with CTF and without)
- I implemented thus my own bit align code, based on firmware I wrote for the Kintex 7
- On Kintex 7, the Xilinx bit align machine does not work anymore at all (no surprise, it was for Virtex 5)
- IODELAY has only 32 taps of 78 ps, so that one cannot scan through the whole eye when running the SERDES with 400 MHz (delays between 0 and 2.418 ns possible), 400 MHz is slow for Series 7
- New code measures all combinations of bit slip and tap and chooses setting in the center of the largest stable region
- A tab /bit slip combination is stable if the correct link word has been received 128 times in a row
- If e.g. at the same bit slip, taps 10-30 are stable, we have a stable region of 21 taps for this bit slip. The center of the window is then tap 20
- After changes a very stable link status 4 is obtained with or without CTF on all HDMI ports

FEC and Slow control

NIM trigger and multiple test pulses



- For trigger in and trigger out now the “polarity” of the digital signal can be set via the slow control
- Trigger input can be given out on trigger output
- Alternatively, trigger output can be configured to occur at any time in the cycle
- Michael’s “Registering trigger timestamp feature” unchanged, just checkbox removed, as soon as the trigger input is activated in the slow control, timestamp is sent out
- Instead of starting the acquisition directly after clicking “ACQ ON” in the slow control, feature to only start the acquisition with the arrival of the first NIM trigger
- It is now possible to have several internal test pulses at defined times

The screenshot shows the VMM3 - SRS DCS new software interface. Key sections include:

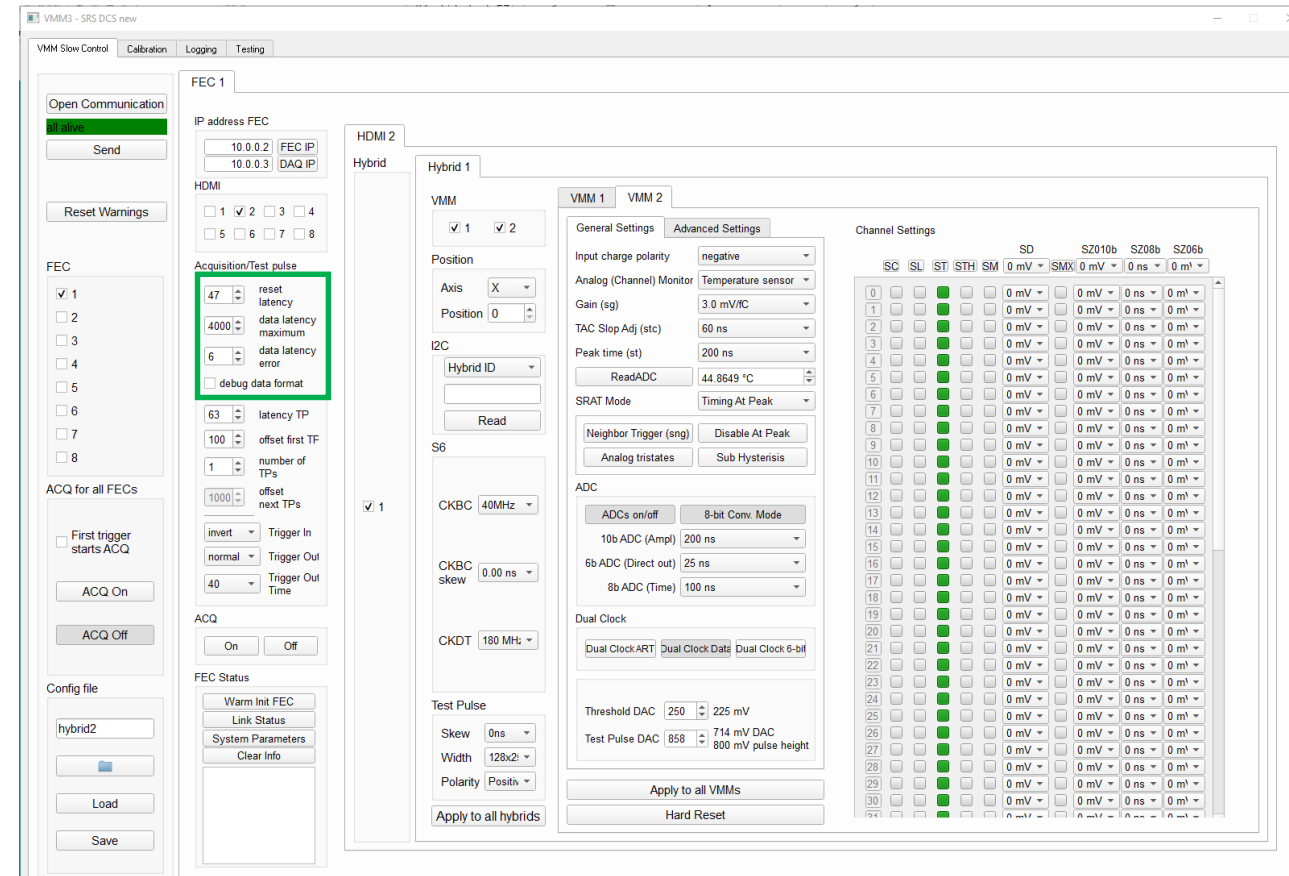
- FEC 1:** IP address FEC (10.0.0.2 FEC IP, 10.0.0.3 DAQ IP), HDMI (checkboxes 1-8), Acquisition/Test pulse (latency TP: 63, offset first TF: 100, number of TPs: 1, offset next TPs: 1000), ACQ for all FECs (checkboxes 1-8), ACQ (On/Off), and Config file (hybrid2).
- HDMI 2 Hybrid:** Hybrid 1 (checkboxes 1, 2), VMM (checkboxes 1, 2), Position (Axis X, Position 0), I2C (Hybrid ID, Read), S6 (checkboxes 1-4), CKBC (40MHz), CKBC skew (0.00 ns), CKDT (180 MHz), Test Pulse (Skew: 0ns, Width: 128x2, Polarity: Positiv), and Apply to all hybrids.
- VMM 1 VMM 2:** General Settings (Input charge polarity: negative, Analog (Channel) Monitor: Temperature sensor, Gain (sg): 3.0 mV/IC, TAC Slop Adj (stc): 60 ns, Peak time (st): 200 ns, ReadADC: 44.8649 °C, SRAT Mode: Timing At Peak), Advanced Settings (Neighbor Trigger (sng), Disable At Peak, Analog tristates, Sub Hysteresis), ADC (ADCs on/off, 8-bit Conv. Mode, 10b ADC (Amp): 200 ns, 6b ADC (Direct out): 25 ns, 8b ADC (Time): 100 ns), Dual Clock (Dual Clock ART, Dual Clock Data, Dual Clock 6-bit), Threshold DAC (250, 225 mV), Test Pulse DAC (858, 714 mV DAC, 800 mV pulse height), and Apply to all VMMs.
- Channel Settings:** A table with columns SC, SL, ST, STH, SM, SD, S2010b, S208b, S206b, and rows 0-30.

FEC and Slow control

Continuous data taking without window

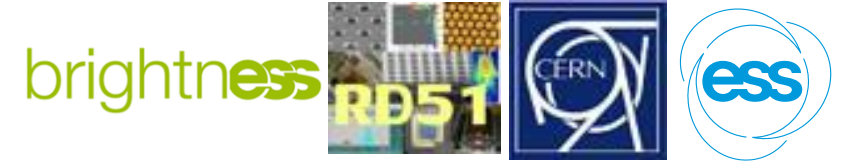


- For the use at ESS (neutron time of flight measurements), we have to be able to take data all the time without regularly occurring dead time
- BC clock at ESS will be derived from facility clock and have fixed relation with proton pulse
- Problem so far has been the overflow of the BCID on the VMM
- As a 12 bit value, the BCID on the VMM only covers times between 0 and 102.4 us (at 40 MHz)
- Due to limited bandwidth, we do not add a higher order timestamp to the hits on the Spartan6, since then the hit size would considerably increase from 40 bits to ≥ 100 bits
- Higher order 42 bit timestamp with 25 ns resolution and a BCID overflow counter (offset) is added upon arrival of the hits on the FEC



FEC and Slow control

Continuous data taking without window



- Challenge: Since the data is read out out from the VMM via the Spartan6 and the HDMI in a serial fashion, there is a large difference between hits arriving with minimum and maximum latency
- One solution to ensure data integrity is to use a acceptance window as before
- New solution:
 - The FEC has a clock counter (TRG) that counts for each BC clock frequency from 0 to 4095
 - At the beginning of the acquisition, the BCID on the VMMs is reset (soft reset), and the S6 FIFO is emptied
 - The moment, at which the reset is sent, can be determined with the field "reset latency"
 - In the new debug data mode, the value of the FEC clock counter, at which the hit arrived, is sent out in the data instead of the ADC and TDC
- Two steps to set things up:
 - The "reset latency" has to be set in such a way, that the BCID and the TRG in Wireshark show the same value (please pulse only one channel)
 - If TRG and BCID show the same value, but the BCID too large or too small (e.g. 105 instead of 100), then the "TP latency" has to be increased or decreased

```
▼ SRS Header
  Frame Counter: 26028 (-85510)
  Data Id: VMM3a Data
  FEC ID: 2
  UDP Timestamp: 86917232 (-27161085500)
  Offset overflow last frame: 12
  > Hit: 1, offset: 4, vmmID: 2, ch: 0, bcid: 100, trg: 99, latency: -1
  > Hit: 2, offset: 5, vmmID: 2, ch: 0, bcid: 100, trg: 99, latency: -1
  > Hit: 3, offset: 6, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
  > Hit: 4, offset: 7, vmmID: 2, ch: 0, bcid: 100, trg: 101, latency: 1
  > Hit: 5, offset: 8, vmmID: 2, ch: 0, bcid: 100, trg: 101, latency: 1
  > Hit: 6, offset: 9, vmmID: 2, ch: 0, bcid: 100, trg: 99, latency: -1
  > Hit: 7, offset: 10, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
  > Hit: 8, offset: 11, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
  > Hit: 9, offset: 12, vmmID: 2, ch: 0, bcid: 100, trg: 101, latency: 1
  > Hit: 10, offset: 13, vmmID: 2, ch: 0, bcid: 100, trg: 99, latency: -1
  > Hit: 11, offset: 14, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
  > Hit: 12, offset: 15, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
  > Marker: 1, VMM ID 2, SRS timestamp: 81530880
  > Marker: 2, VMM ID 3, SRS timestamp: 81530880
```

FEC clock counter at which hit arrives

↓

FEC and Slow control

Continuous data taking without window



- The maximum latency setting now depends on the CKDT setting
- With 180 MHz DDR one hits arrives every 5th 40 MHz clock cycles (yeah!! Thanks to Patrick!!)
- If the 4 hit deep FIFO exists, the maximum latency is $64 \times 4 \times 5$ clock cycles
- If all 64 channels are pulsed 4 times in quick succession, the maximum latency is thus 1280 clock cycles
- Last setting needed is "latency error". We can see that there is a bit of jitter, sometimes a hit that normally arrives at 0 latency will arrive at -1 or +1. This might be due to the internal test pulses (let's do measurements with external pulsers)
- Algorithm for higher order time stamp:
 - $TRG \geq BCID$ and $TRG - BCID \leq$ maximum latency: current offset
 - $TRG < BCID$ and $BCID - TRG \leq$ latency error: current offset
 - $TRG < BCID$ and $BCID - TRG \leq 4096 -$ maximum latency: previous offset
 - Other cases: invalid

▼ SRS Header

Frame Counter: 32717 (-78821)
 Data Id: VMM3a Data
 FEC ID: 2
 UDP Timestamp: 140136765 (-25830597175)
 Offset overflow last frame: 12

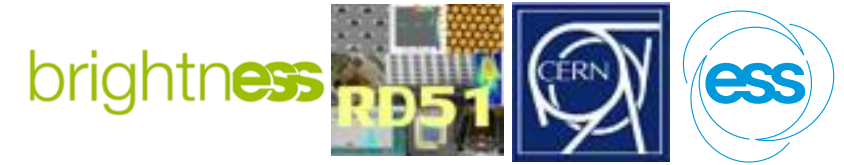
> Hit: 1, offset: 9, vmmID: 2, ch: 0, bcid: 100, trg: 100, latency: 0
> Hit: 2, offset: 9, vmmID: 3, ch: 0, bcid: 101, trg: 100, latency: -1
> Hit: 3, offset: 9, vmmID: 3, ch: 1, bcid: 101, trg: 105, latency: 4
> Hit: 4, offset: 9, vmmID: 2, ch: 1, bcid: 100, trg: 105, latency: 5
> Hit: 5, offset: 9, vmmID: 3, ch: 2, bcid: 101, trg: 110, latency: 9
> Hit: 6, offset: 9, vmmID: 3, ch: 3, bcid: 101, trg: 115, latency: 14
> Hit: 7, offset: 9, vmmID: 2, ch: 2, bcid: 100, trg: 110, latency: 10
> Hit: 8, offset: 9, vmmID: 3, ch: 4, bcid: 101, trg: 120, latency: 19
> Hit: 9, offset: 9, vmmID: 2, ch: 3, bcid: 100, trg: 115, latency: 15
> Hit: 10, offset: 9, vmmID: 3, ch: 5, bcid: 101, trg: 125, latency: 24
> Hit: 11, offset: 9, vmmID: 2, ch: 4, bcid: 100, trg: 120, latency: 20
> Hit: 12, offset: 9, vmmID: 2, ch: 5, bcid: 100, trg: 125, latency: 25
> Hit: 13, offset: 9, vmmID: 3, ch: 6, bcid: 101, trg: 130, latency: 29
> Hit: 14, offset: 9, vmmID: 2, ch: 6, bcid: 101, trg: 130, latency: 29
> Hit: 15, offset: 9, vmmID: 3, ch: 7, bcid: 101, trg: 135, latency: 34
> Hit: 16, offset: 9, vmmID: 2, ch: 7, bcid: 101, trg: 135, latency: 34
> Hit: 17, offset: 9, vmmID: 2, ch: 8, bcid: 100, trg: 140, latency: 40
> Hit: 18, offset: 9, vmmID: 3, ch: 8, bcid: 101, trg: 140, latency: 39
> Hit: 19, offset: 9, vmmID: 2, ch: 9, bcid: 100, trg: 145, latency: 45
> Hit: 20, offset: 9, vmmID: 3, ch: 9, bcid: 101, trg: 145, latency: 44
> Hit: 21, offset: 9, vmmID: 2, ch: 10, bcid: 101, trg: 150, latency: 49
> Hit: 22, offset: 9, vmmID: 3, ch: 10, bcid: 100, trg: 150, latency: 50
> Hit: 23, offset: 9, vmmID: 2, ch: 11, bcid: 101, trg: 155, latency: 54
> Hit: 24, offset: 9, vmmID: 3, ch: 11, bcid: 101, trg: 155, latency: 54
> Hit: 25, offset: 9, vmmID: 2, ch: 12, bcid: 100, trg: 160, latency: 60
> Hit: 26, offset: 9, vmmID: 3, ch: 12, bcid: 101, trg: 160, latency: 59
> Hit: 27, offset: 9, vmmID: 2, ch: 13, bcid: 101, trg: 165, latency: 64
> Hit: 28, offset: 9, vmmID: 3, ch: 13, bcid: 101, trg: 165, latency: 64
> Hit: 29, offset: 9, vmmID: 2, ch: 14, bcid: 101, trg: 170, latency: 69
> Hit: 30, offset: 9, vmmID: 3, ch: 14, bcid: 101, trg: 170, latency: 69
> Hit: 31, offset: 9, vmmID: 2, ch: 15, bcid: 101, trg: 175, latency: 74
> Hit: 32, offset: 9, vmmID: 3, ch: 15, bcid: 101, trg: 175, latency: 74
> Hit: 33, offset: 9, vmmID: 2, ch: 16, bcid: 101, trg: 180, latency: 79
> Hit: 34, offset: 9, vmmID: 3, ch: 16, bcid: 101, trg: 180, latency: 79
> Hit: 35, offset: 9, vmmID: 2, ch: 17, bcid: 101, trg: 185, latency: 84
> Hit: 36, offset: 9, vmmID: 3, ch: 17, bcid: 101, trg: 185, latency: 84
> Hit: 37, offset: 9, vmmID: 2, ch: 18, bcid: 101, trg: 190, latency: 89

How cool, a new hit every 5 cycles !

FEC and Slow control

Continuous data taking without window

- If hits belong to previous offset, offset -1 is added as timestamp
- Only problem arising if offset on the FEC is 0, and a new marker has already been generated
- Then -1 is sent as offset
- This means the 5 bit offset is not a unsigned number any more with valid entries going from 0 to 31, but a signed number with a valid range from -1 to 15
- -16 is used as indicator for a hit that violates the latency conditions, these hits can be discarded by the DAQ (or directly on the FEC if needed)



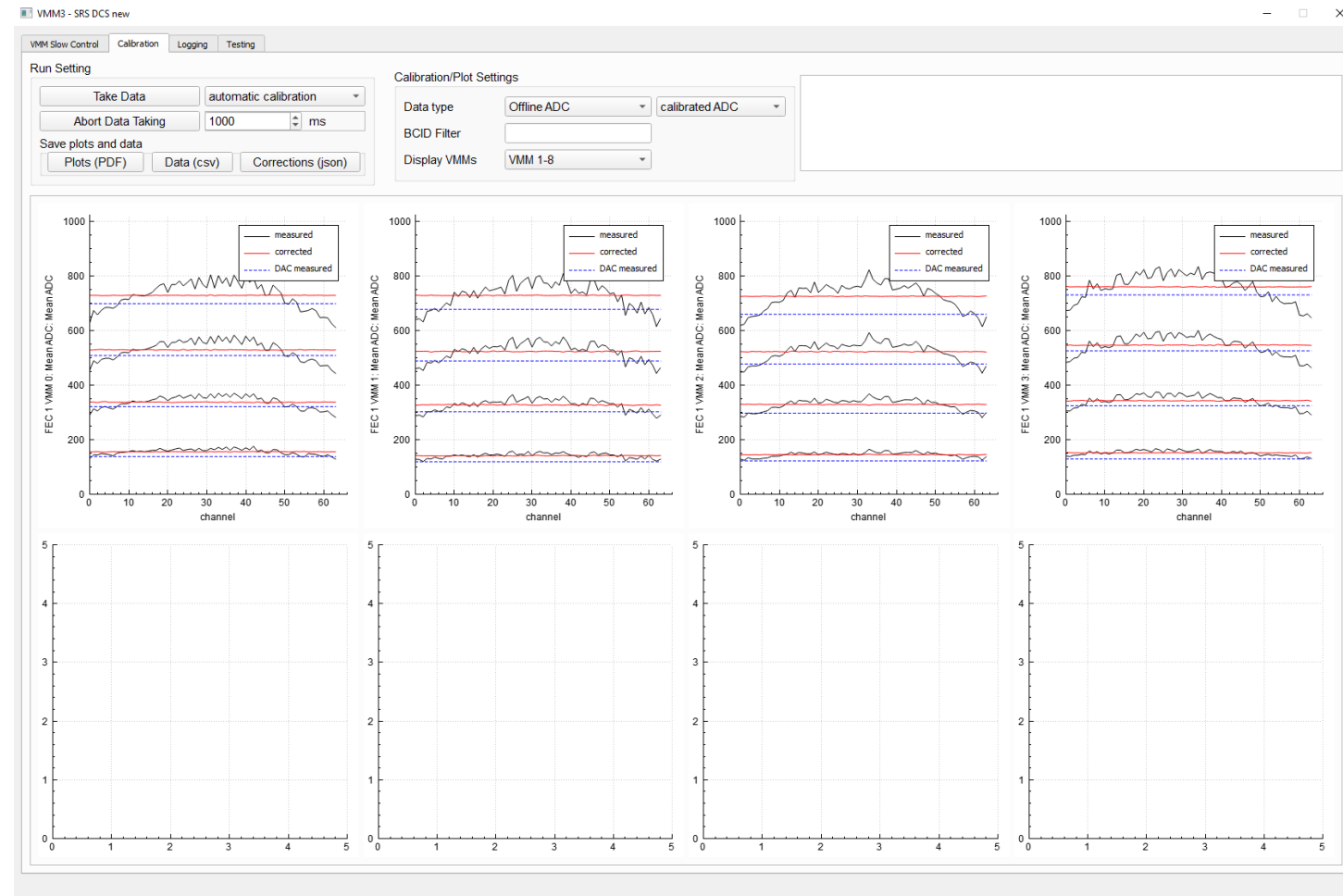
```
> Hit: 436, offset: 15, vmmID: 2, ch: 11, bcid: 4001, trg: 4055, latency: 54
> Hit: 437, offset: 15, vmmID: 3, ch: 11, bcid: 4001, trg: 4055, latency: 54
> Hit: 438, offset: 15, vmmID: 2, ch: 12, bcid: 4000, trg: 4060, latency: 60
> Hit: 439, offset: 15, vmmID: 3, ch: 12, bcid: 4001, trg: 4060, latency: 59
> Hit: 440, offset: 15, vmmID: 2, ch: 13, bcid: 4000, trg: 4065, latency: 65
> Hit: 441, offset: 15, vmmID: 3, ch: 13, bcid: 4001, trg: 4065, latency: 64
> Hit: 442, offset: 15, vmmID: 2, ch: 14, bcid: 4001, trg: 4070, latency: 69
> Hit: 443, offset: 15, vmmID: 3, ch: 14, bcid: 4001, trg: 4070, latency: 69
> Hit: 444, offset: 15, vmmID: 2, ch: 15, bcid: 4000, trg: 4075, latency: 75
> Hit: 445, offset: 15, vmmID: 3, ch: 15, bcid: 4001, trg: 4075, latency: 74
> Hit: 446, offset: 15, vmmID: 2, ch: 16, bcid: 4000, trg: 4080, latency: 80
> Hit: 447, offset: 15, vmmID: 3, ch: 16, bcid: 4001, trg: 4080, latency: 79
> Hit: 448, offset: 15, vmmID: 2, ch: 17, bcid: 4001, trg: 4085, latency: 84
> Hit: 449, offset: 15, vmmID: 3, ch: 17, bcid: 4001, trg: 4085, latency: 84
> Hit: 450, offset: 15, vmmID: 2, ch: 18, bcid: 4001, trg: 4090, latency: 89
> Hit: 451, offset: 15, vmmID: 3, ch: 18, bcid: 4001, trg: 4090, latency: 89
> Hit: 452, offset: 15, vmmID: 2, ch: 19, bcid: 4001, trg: 4095, latency: 94
> Hit: 453, offset: 15, vmmID: 3, ch: 19, bcid: 4001, trg: 4095, latency: 94
> Marker: 1, VMM ID 3, SRS timestamp: 95162368
> Marker: 2, VMM ID 2, SRS timestamp: 95162368
> Hit: 454, offset: -1, vmmID: 2, ch: 20, bcid: 4001, trg: 4, latency: 99
> Hit: 455, offset: -1, vmmID: 3, ch: 20, bcid: 4001, trg: 4, latency: 99
> Hit: 456, offset: -1, vmmID: 2, ch: 21, bcid: 4001, trg: 9, latency: 104
> Hit: 457, offset: -1, vmmID: 3, ch: 21, bcid: 4001, trg: 9, latency: 104
> Hit: 458, offset: -1, vmmID: 2, ch: 22, bcid: 4001, trg: 14, latency: 109
> Hit: 459, offset: -1, vmmID: 3, ch: 22, bcid: 4001, trg: 14, latency: 109
> Hit: 460, offset: -1, vmmID: 2, ch: 23, bcid: 4001, trg: 19, latency: 114
> Hit: 461, offset: -1, vmmID: 3, ch: 23, bcid: 4001, trg: 19, latency: 114
> Hit: 462, offset: -1, vmmID: 2, ch: 24, bcid: 4001, trg: 24, latency: 119
> Hit: 463, offset: -1, vmmID: 3, ch: 24, bcid: 4001, trg: 24, latency: 119
> Hit: 464, offset: -1, vmmID: 2, ch: 25, bcid: 4000, trg: 29, latency: 125
> Hit: 465, offset: -1, vmmID: 3, ch: 25, bcid: 4001, trg: 29, latency: 124
```

Offset now 5bit signed number, to cover cases where hits arrive after generation of new marker

ADC calibration

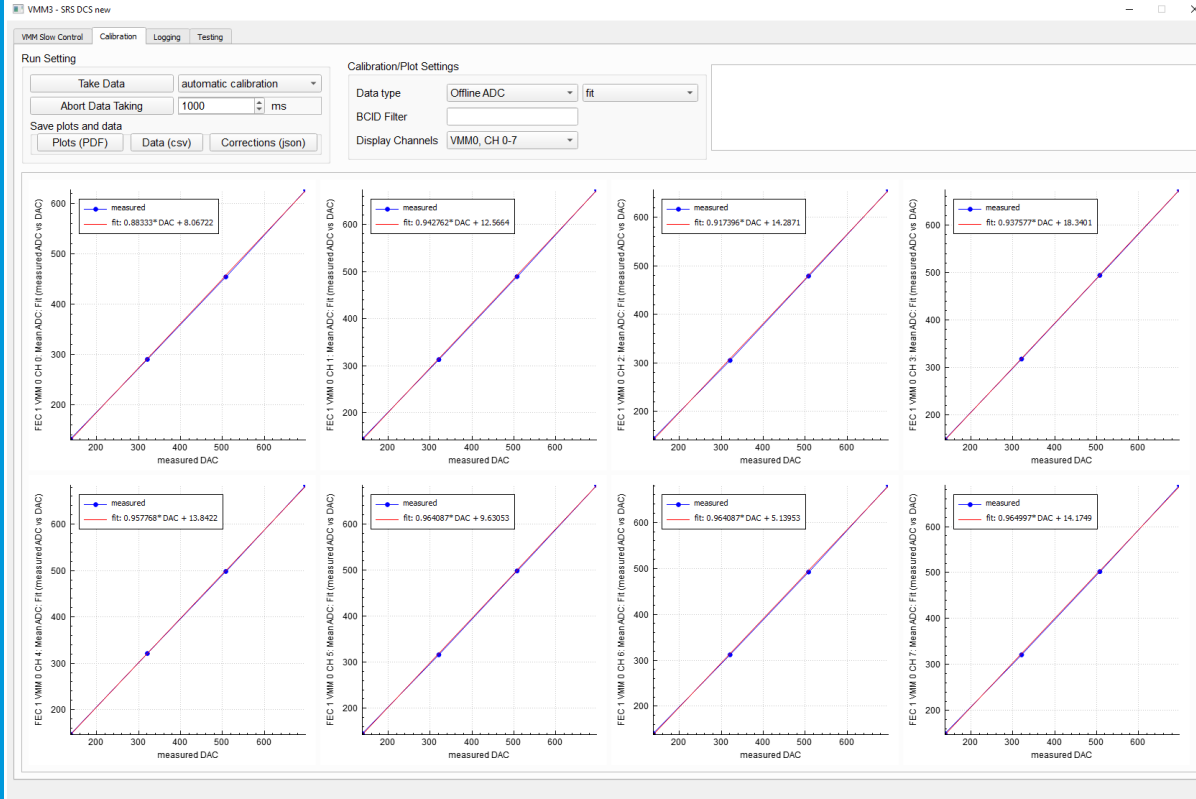
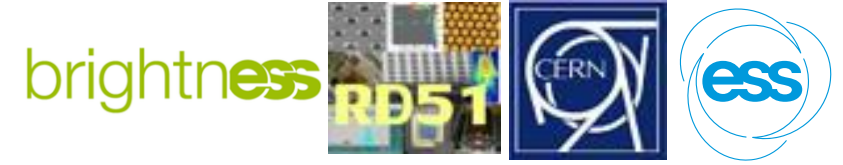
New algorithm

- Pulse all channels with internal test pulses
- For 4 different pulse heights, measure ADC
- Measure the actual level in mV of the pulser DAC (observation: using the same DAC values, different VMMs have quite different pulse heights)
- Fit per channel with pulse height in mV on x-axis, ADC on y-axis and determine slope and offset off the fit
- Calculate corrections per channel

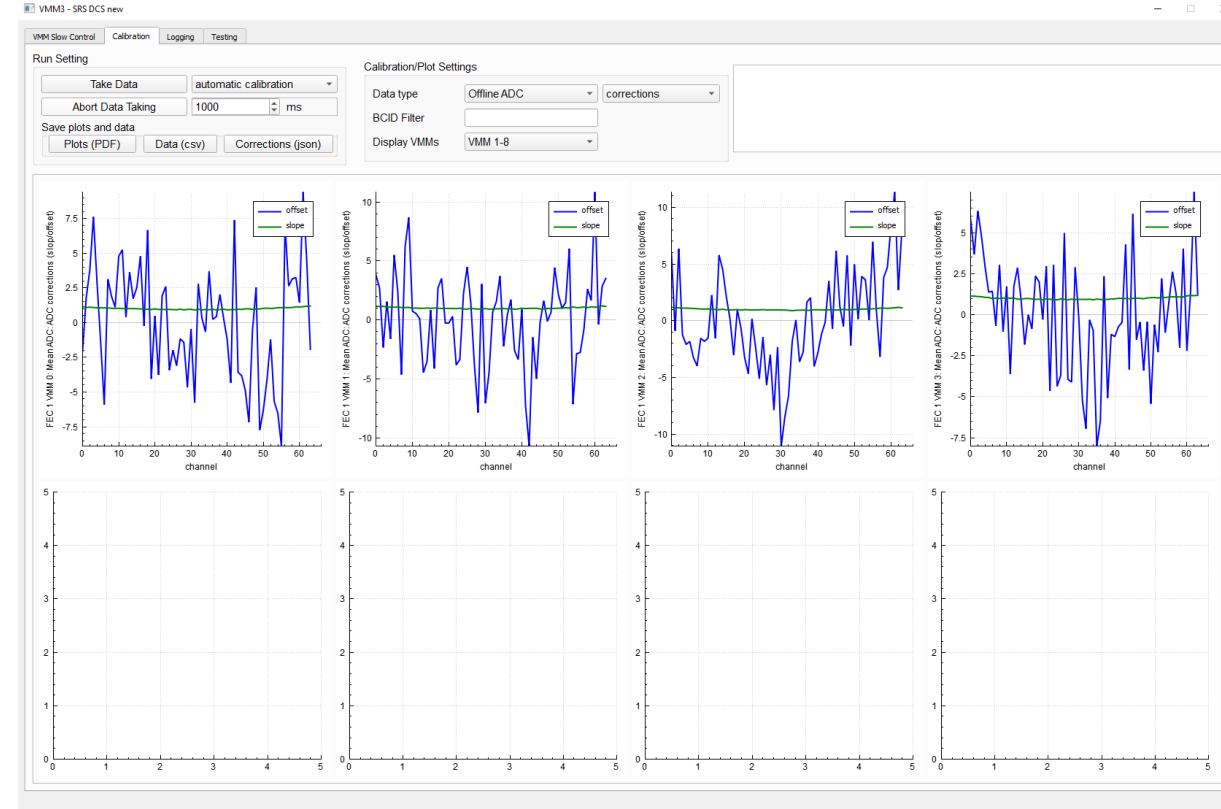


ADC calibration

Fit and corrections



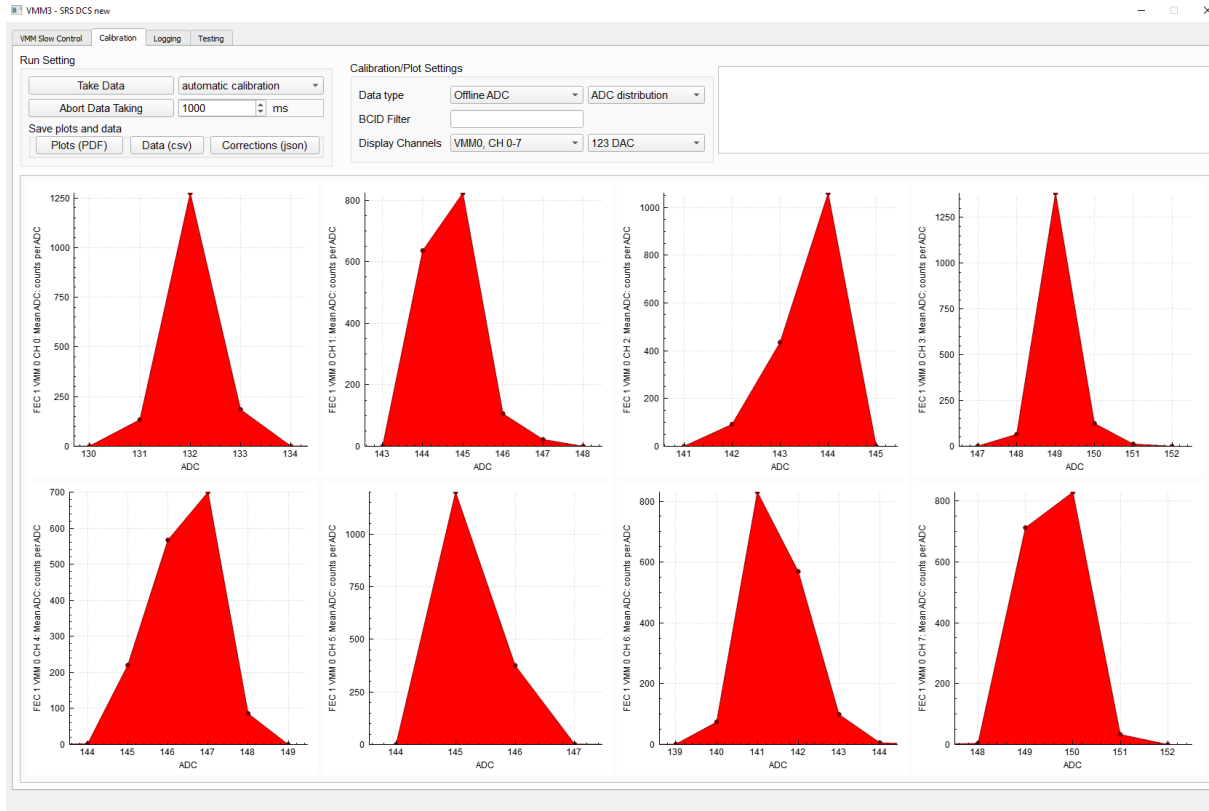
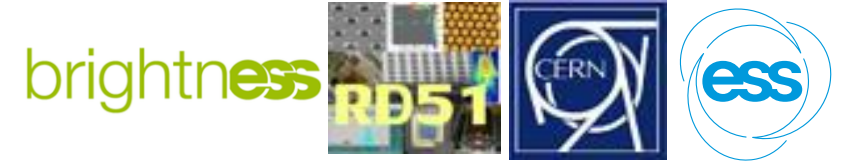
Fit to determine offset and slope corrections
Plot per channel



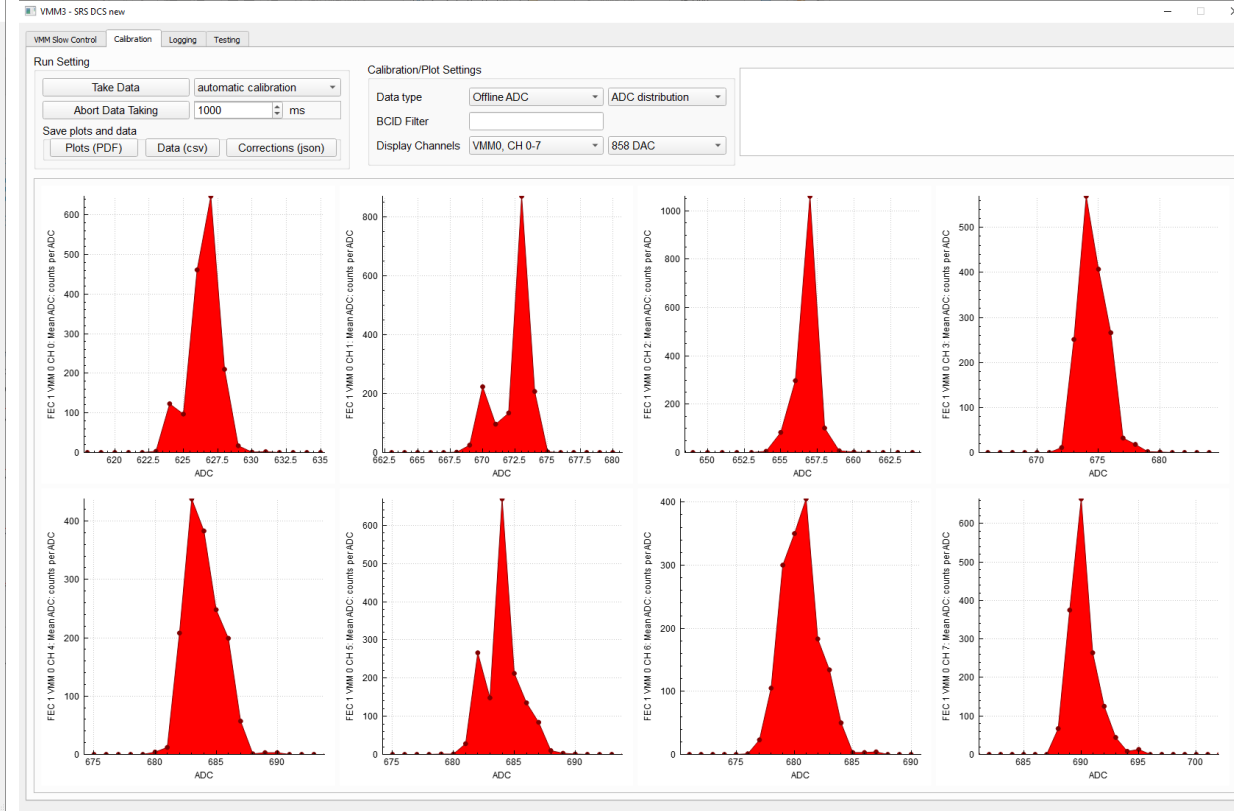
Offset and slope corrections
Plot per VMM

ADC calibration

ADC distribution



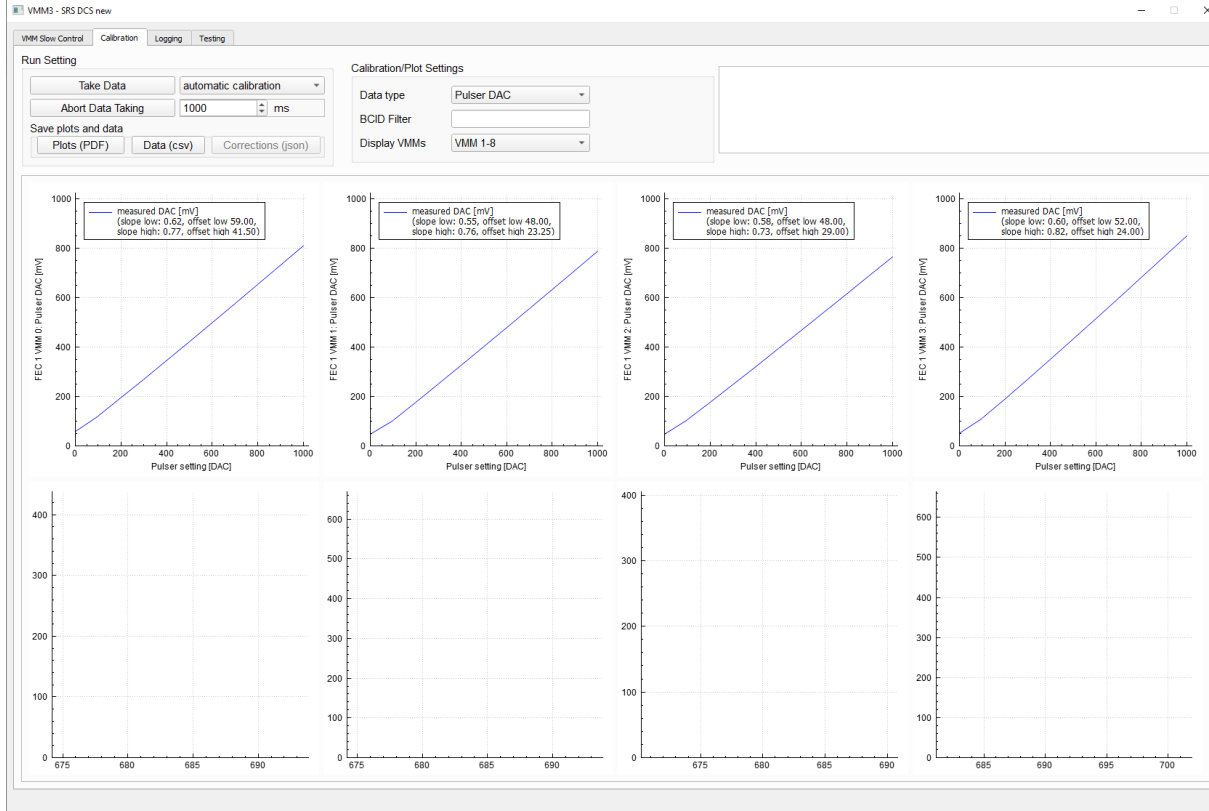
ADC distribution for DAC setting 123
Plot per channel



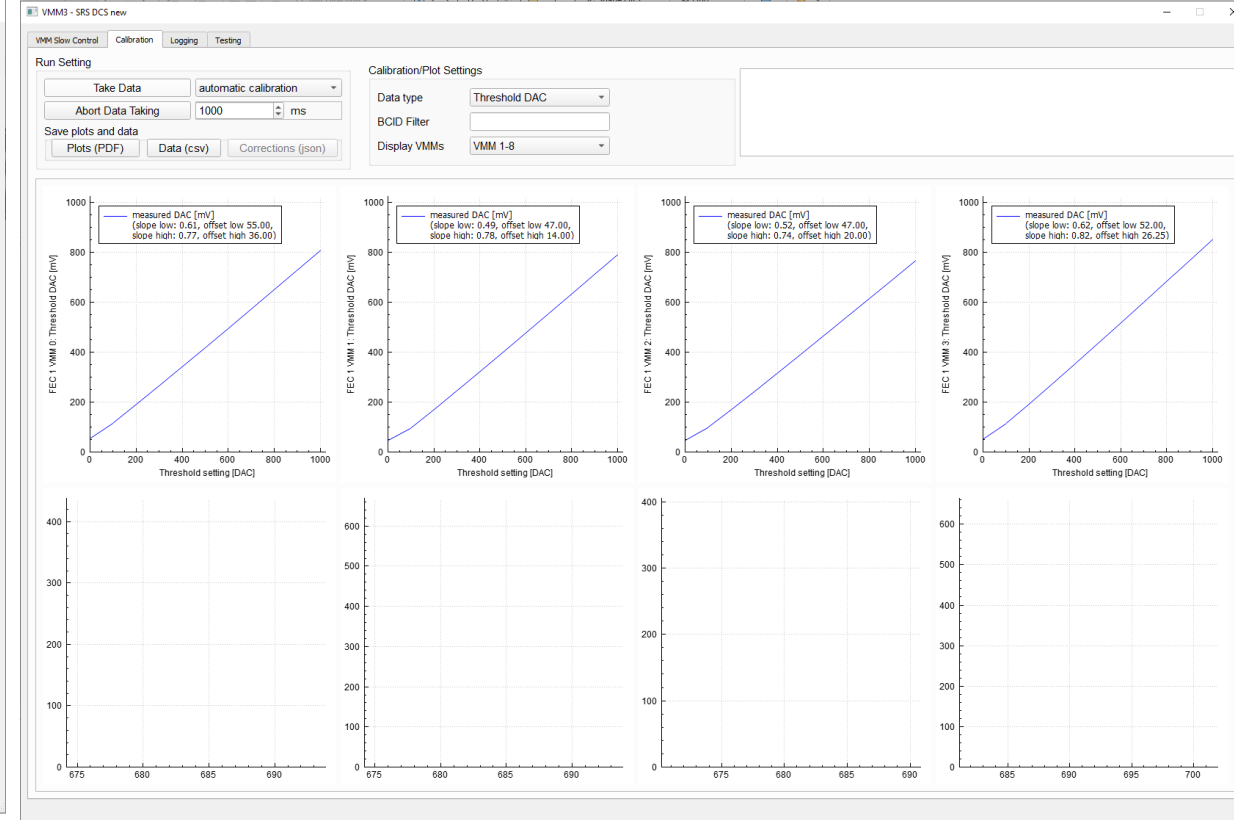
ADC distribution for DAC setting 858
Plot per channel

Pulser and Threshold DAC

Measurement of levels [mV]



Pulser DAC
Plot per VMM



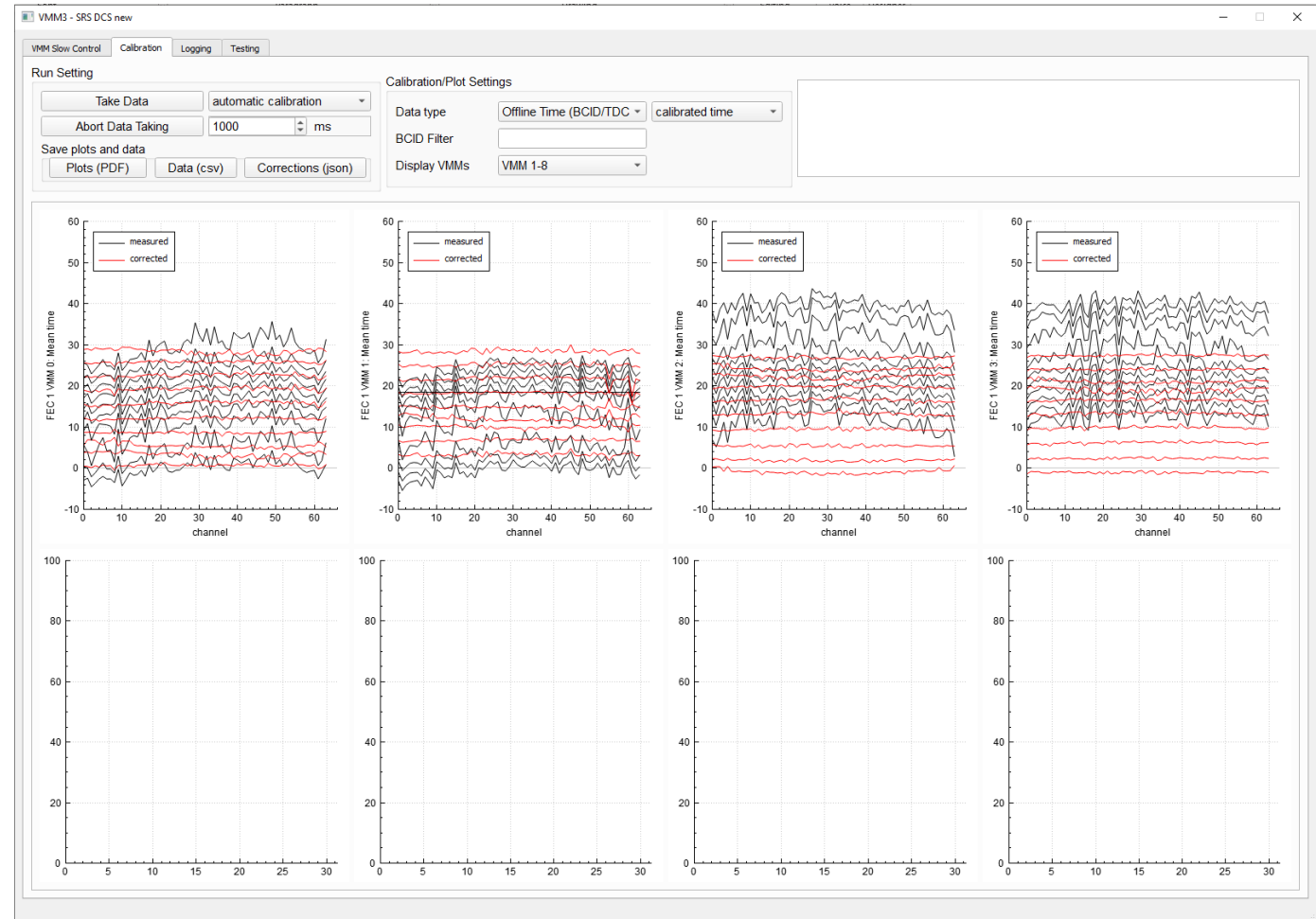
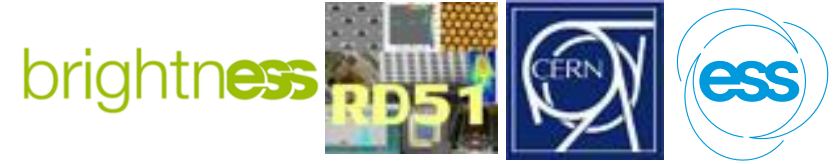
Threshold DAC
Plot per VMM

Both DACs have piece-wise two different slopes!

TDC calibration

New algorithm

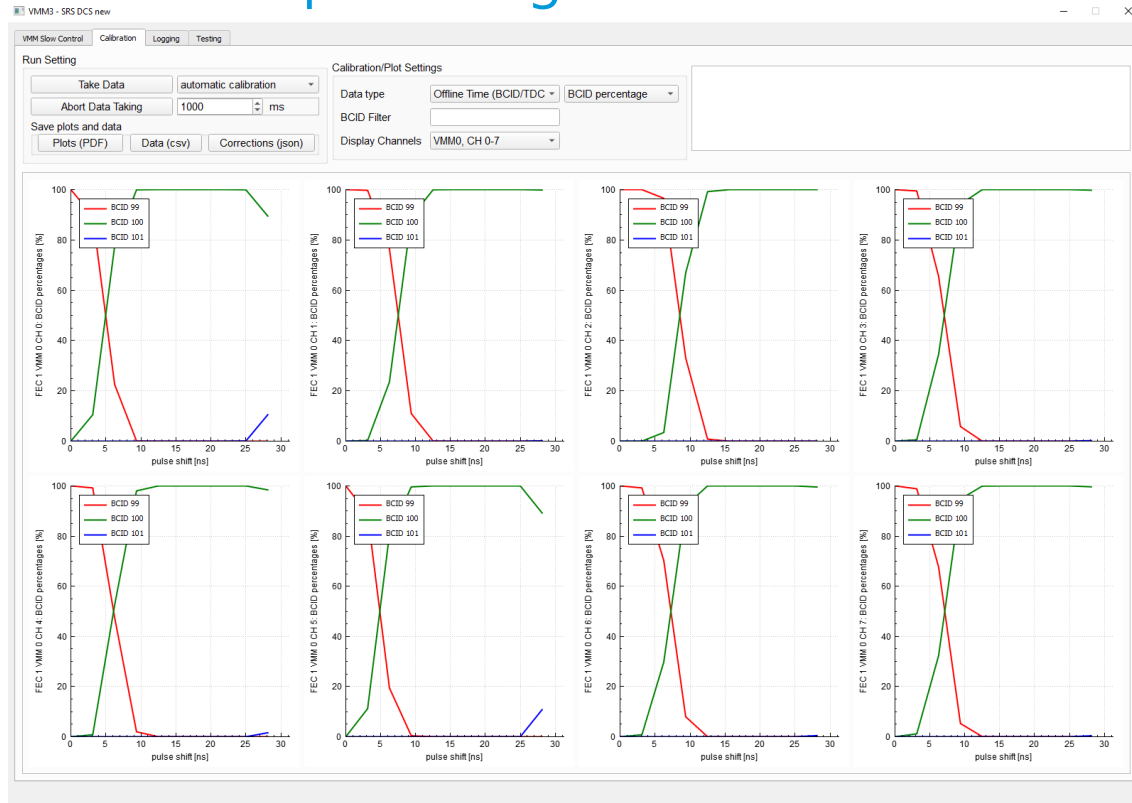
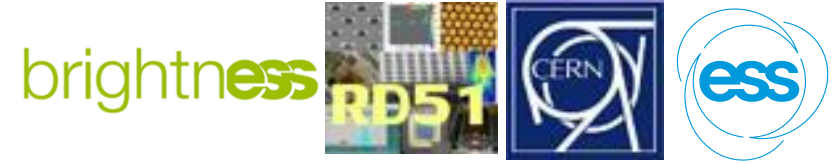
- Pulse all channels with internal test pulses
- Shift test pulse between 0 ns and 31.25 ns
- Measure TDC and the BCID for every channel
- Look for the time shift where 50% of hits have BCID n and 50% have BCID n+1
- At this point one has the largest TDC values for BCID n+1 and the smallest TDC values for BCID n



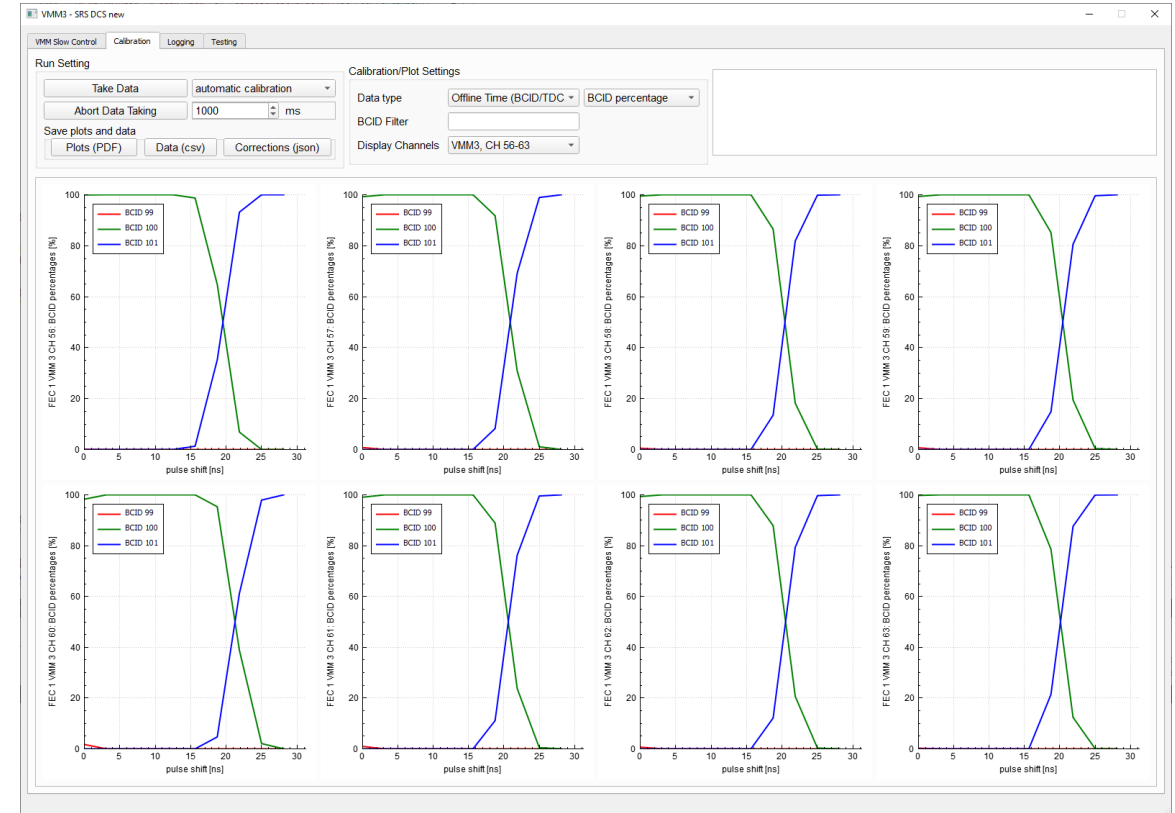
Setting 40 MHz Bc clock, TAC slope 60 ns, shaping time 200 ns

TDC calibration

BCID percentages



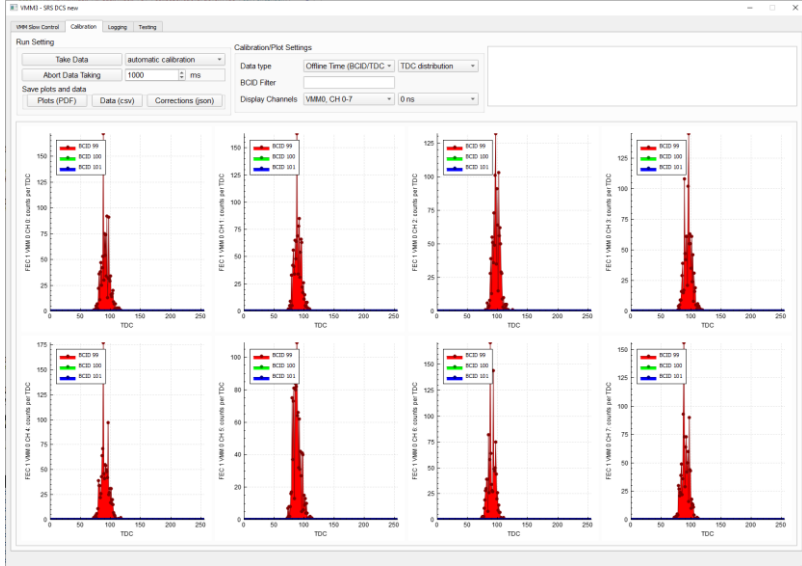
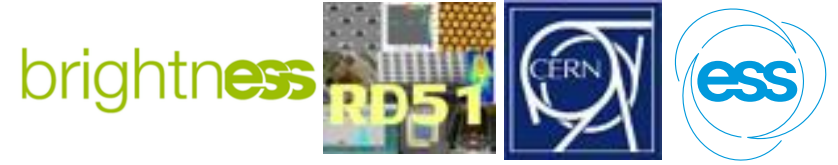
BCID percentage VMM0 (BCID 99-100)



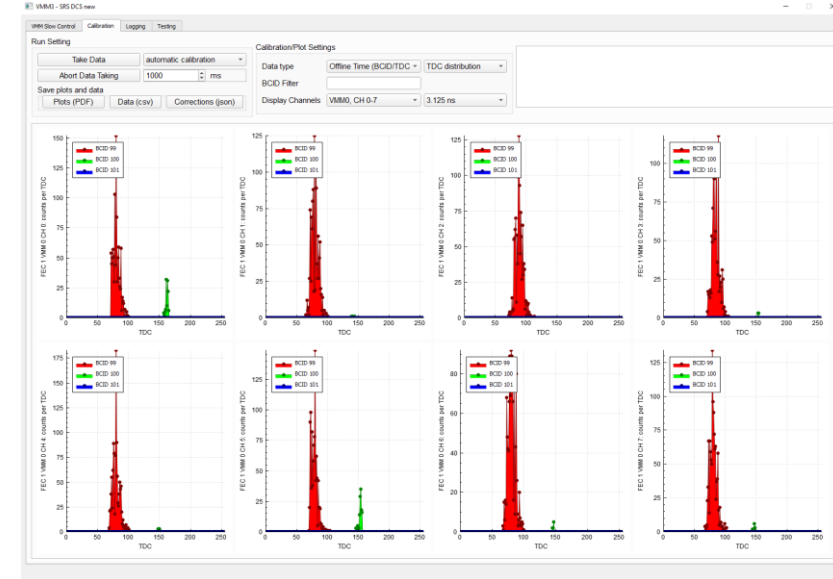
BCID percentage VMM3 (BCID 100-101)

TDC calibration

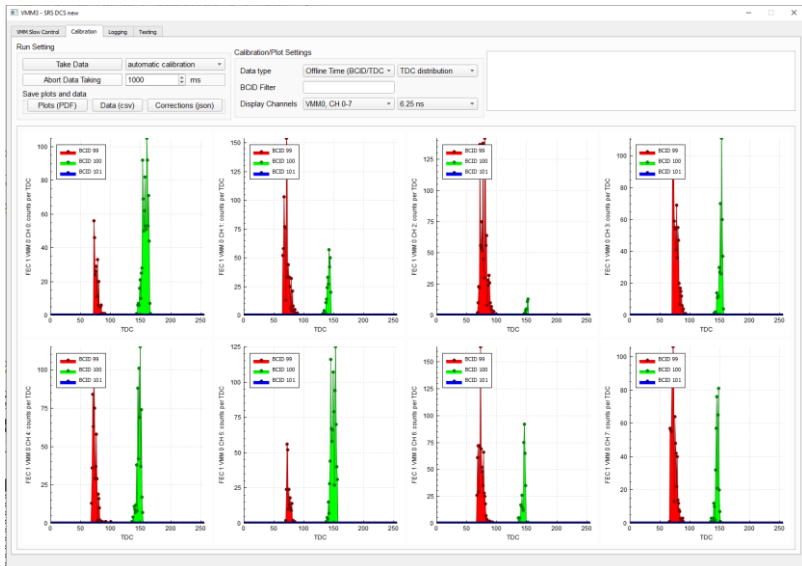
TDC spectrum per BCID



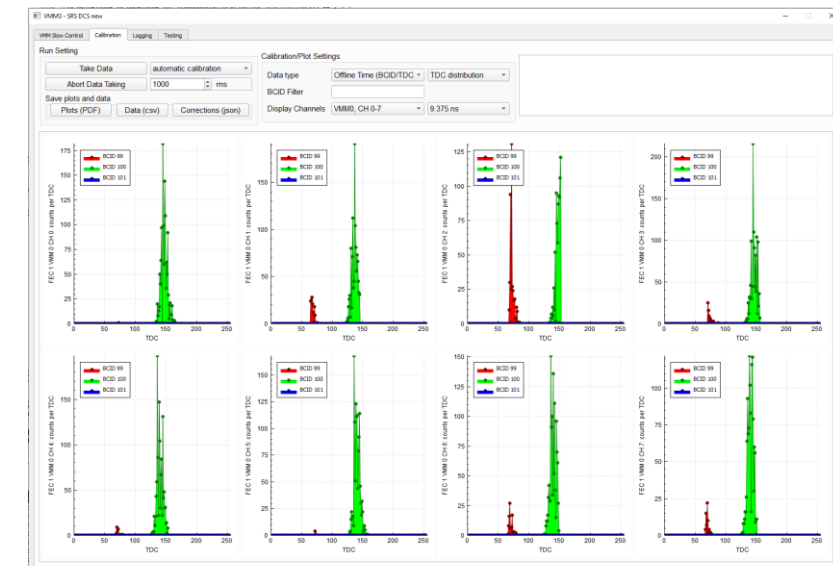
0 ns



3.125 ns



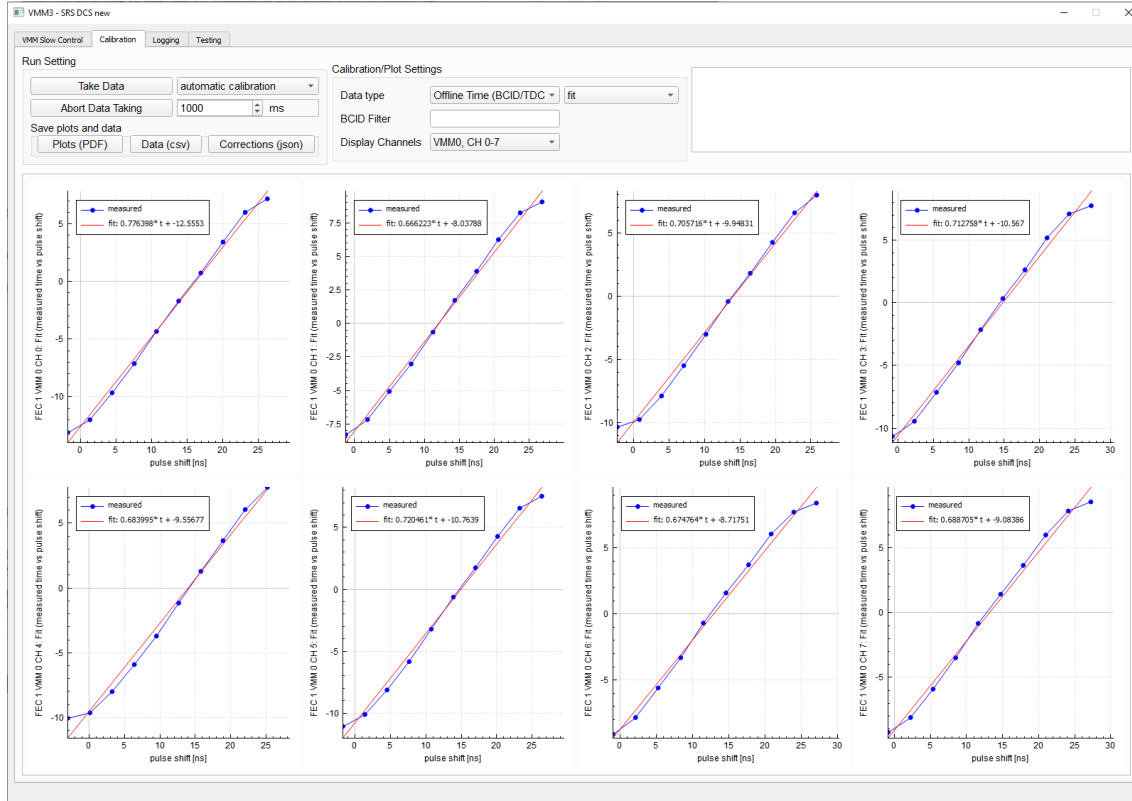
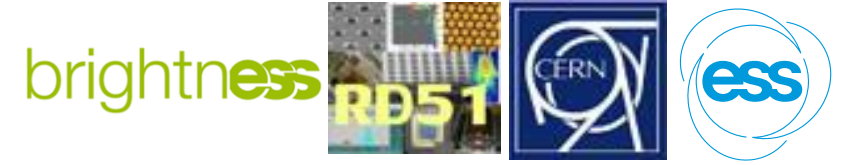
6.125 ns



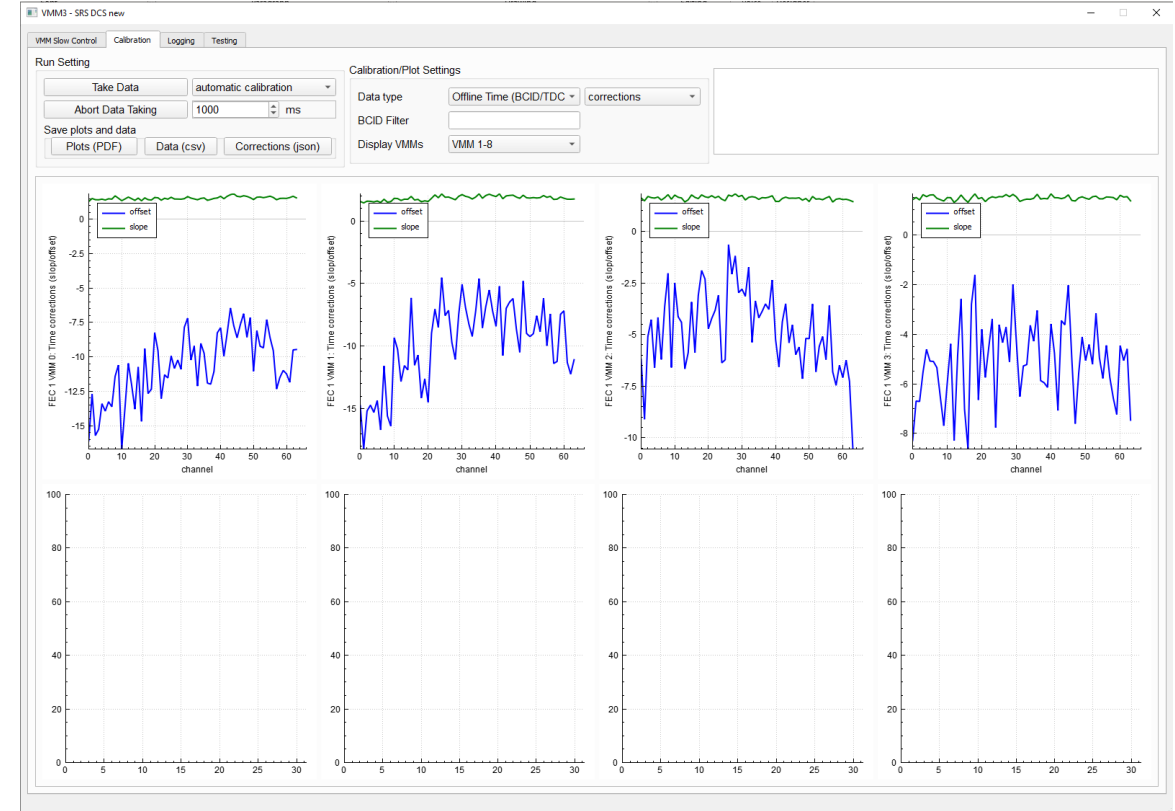
9.375 ns

TDC calibration

Fit and corrections



Fit to determine offset and slope corrections
Plot per channel

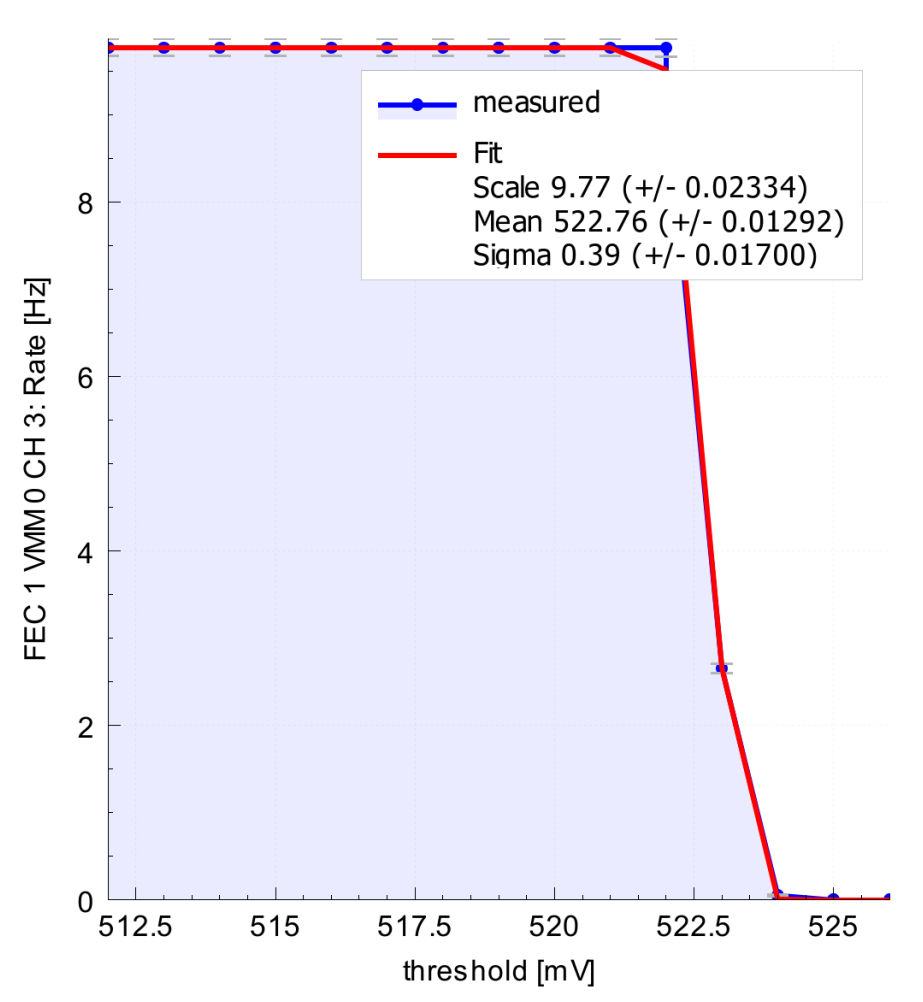


Offset and slope corrections
Plot per VMM

To Do: compare different TAC slopes, shaping times, test whether corrections are stable over time

S-curve

- Pulse all channels with internal test pulses
- Shift the global threshold in steps of one DAC
- Count the number of hits per channel per threshold setting
- Fit complementary error function to data (alglib: nonlinear regression using function values and gradient)
- On Windows, work only for a few channels at a time, otherwise rates do not reach 10 kHz
- On Linux, works for a whole VMM (64 channels)
- Problem: A step of 1 DAC is equivalent to about 0.8 mV, this is very coarse for channels with low noise, fit not always good for steep drop in rates





**EUROPEAN
SPALLATION
SOURCE**