RCE for Pixel Rd53 Chip Readout

Speaker: Zijun Xu (zixu@cern.ch)

Co-authors: Larry Ruckman, Martin Kocian, Matthias Wittgen, Su Dong

49th SLAC Summer Institute

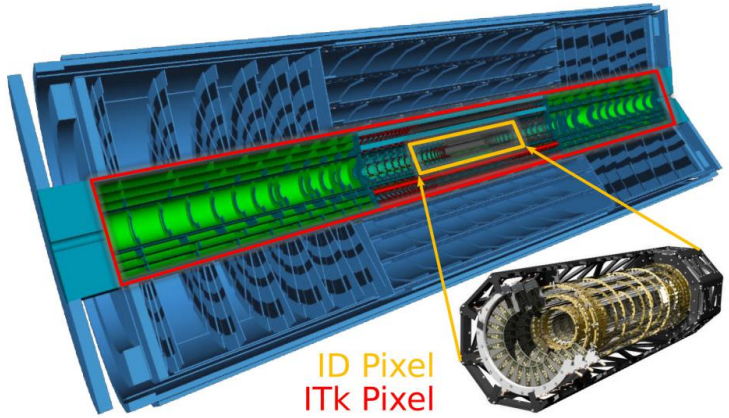
16-27 August 2021

SLAC NATIONAL ACCELERATOR LABORATORY

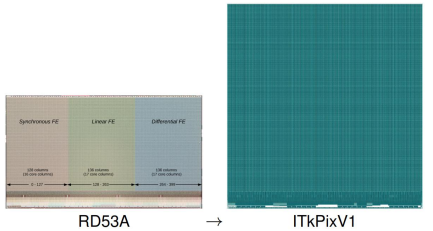
U.S. DEPARTMENT OF ENERGY | Stanford University

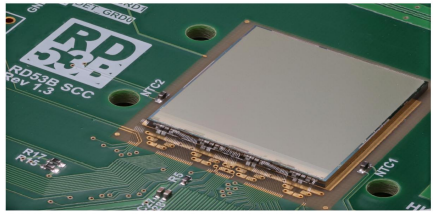
ATLAS Inner Tracker Upgrade and Pixel FE

- ATLAS ITk Upgrade
  - current Inner Detector -> all-silicon ITk
  - pixel: 1.7m^2 -> 13m^2 (5e9 channels)
  - tracking coverage η: 2.5 -> 4.0
  - trig rate: 100 kHz -> 1 MHz
  - pixel data rate increased by a factor of O(100)

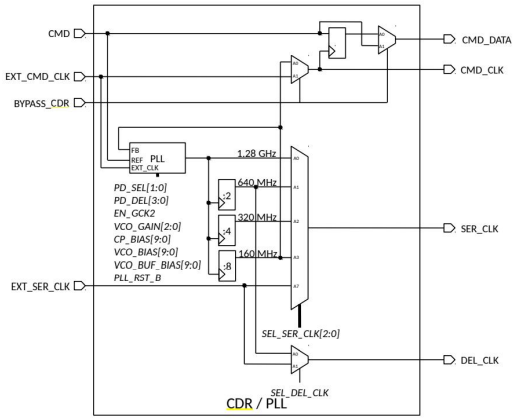


- Rd53 Front-End chip for pixel detector
  - Common design framework for ATLAS and CMS
  - chip is fully controlled with a single serial input stream
    - Lock and Data Recovery circuit (CDR) to extract clock from the 160Mbps Cmd input
  - All data, messages, and configuration read-back are output on high speed serial ports
    - up to **4x 1.28 Gbps** nominal bandwidth





ITkPixV1 chip on Single Chip Card (SCC) [3]



https://twiki.cern.ch/twiki/bin/viewauth/Atlas/RCEGen3SDK

RCE for Pixel RD53a/b readout

- a new generation RCE(Reconfigurable Cluster Element) system
  - Xilinx ZCU102 board with **UltraScale+ ZYNQ** MPSoC XCZU9EG
  - quad-core Arm Cortex-A53, 4GB of DDR4 memory, 64-bit

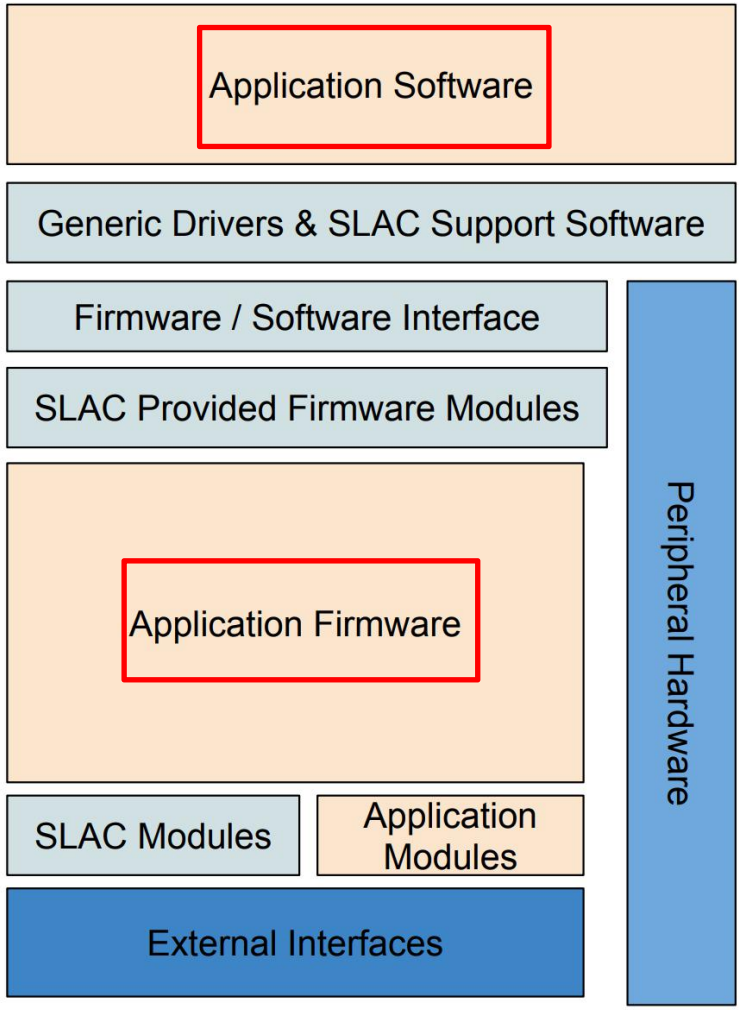
- Custom FMC adaptor to direct run RD53a/b
  - **SelectIO HP** pins for 160Mbps Cmd and 1.28Gbps data
  - Retimer (160MHz clock, PLL, and flip-flop) on the FMC card to reduce Cmd stream random jitter to **2ps**, comparable to lpGBT e-link
  - adjustable Cmd pre-emphasis and data equalizer for lossy data transmission chain
  - FMC pin mapping is compatible with various FPGA boards

RCE DAQ Development

- Based on the common FW&SW library, rapid developments to support various DAQ interface/format/protocols in parallel

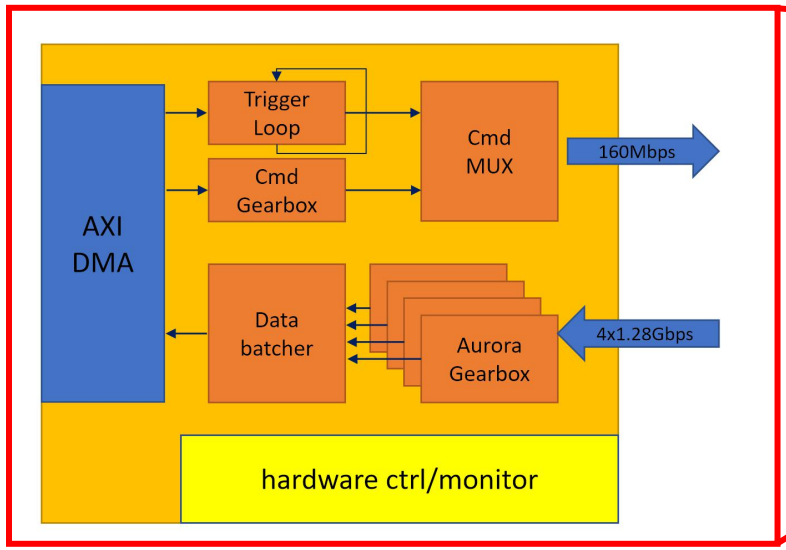
- FW building blocks for a wide range of Xilinx FPGAs/SoCs (RCE like, PCIe cards)
  - https://github.com/slaclab/rce-gen3-fw-lib
  - example: https://github.com/slaclab/rce-gen3-fw-examples
  - Ruckus: Vivado Build System based on Makefiles, https://github.com/slaclab/ruckus

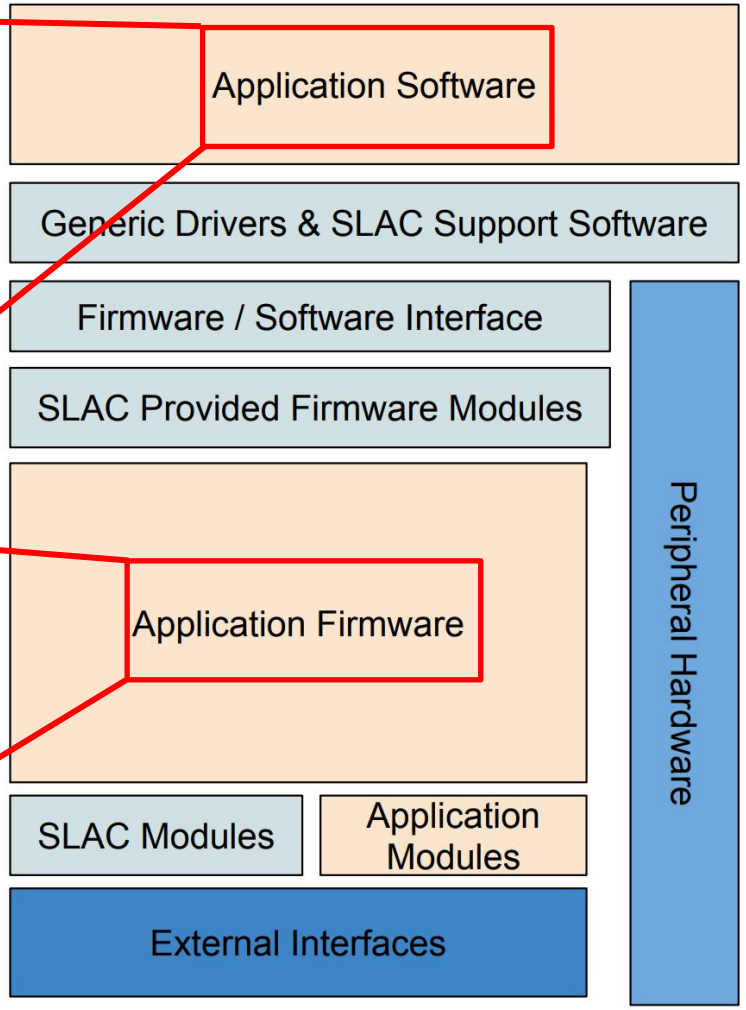
- Hardware Abstraction SW library
  - Rogue: https://github.com/slaclab/rogue
  - Python bindings for rapid DAQ prototyping and GUIs, can be configured for pure C++ use



RCE DAQ Development

- common Rd53 chip calibration software
  - https://gitlab.cern.ch/YARR/
  - aim to support various DAQ platform: YARR, FELIX, RCE, BDAQ ...
- RCE Advantage:
  - SoC CPU embedded in FPGA fabric with direct access to FE I/O ports and firmware resources not confined by PCIe bridge



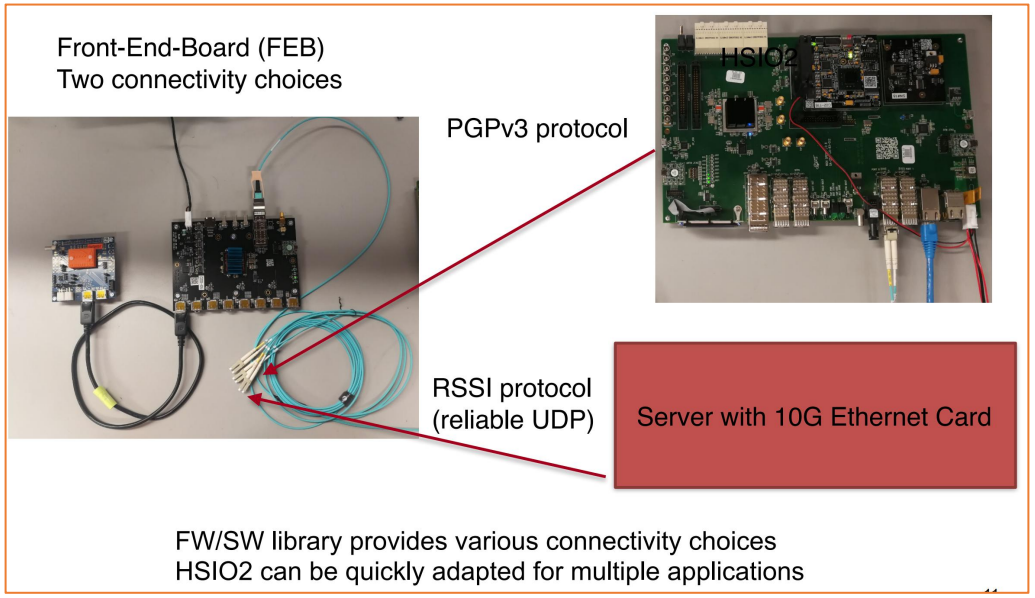


- https://github.com/slaclab/atlas-rd53-fmc-dev
- https://github.com/slaclab/atlas-rd53-fw-lib

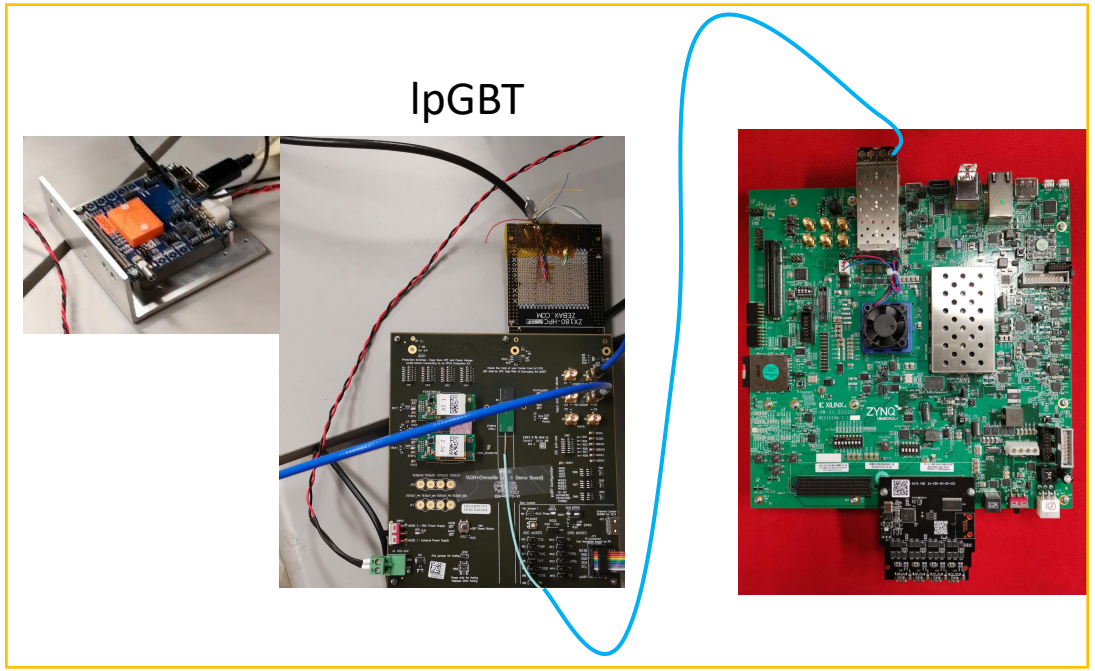
RCE Extension for Pixel RD53 Readout

- Based on the common FW&SW library, enables rapid development to support various data interfaces/formats/protocols

Rd53 + external FPGA + RCE through PGP protocol
Rd53 + external FPGA + PC through 1G/10G-Eth or PCIe



RD53 + FMC + ZCU102
RD53 + lpGBT + ZCU102

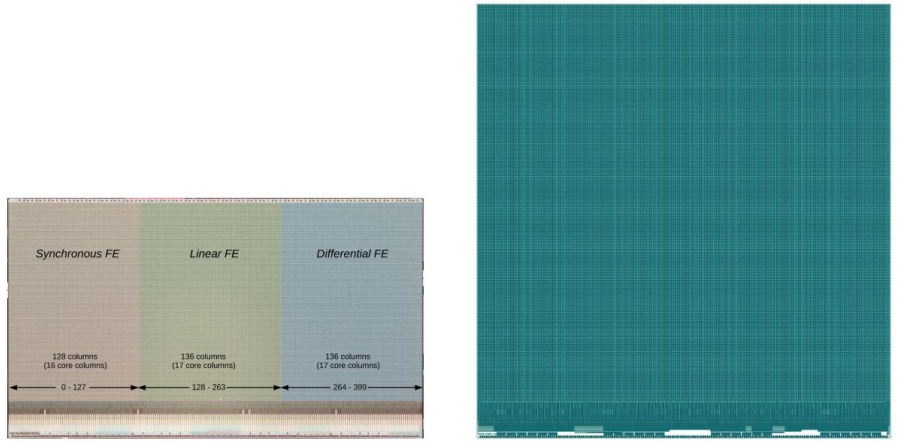


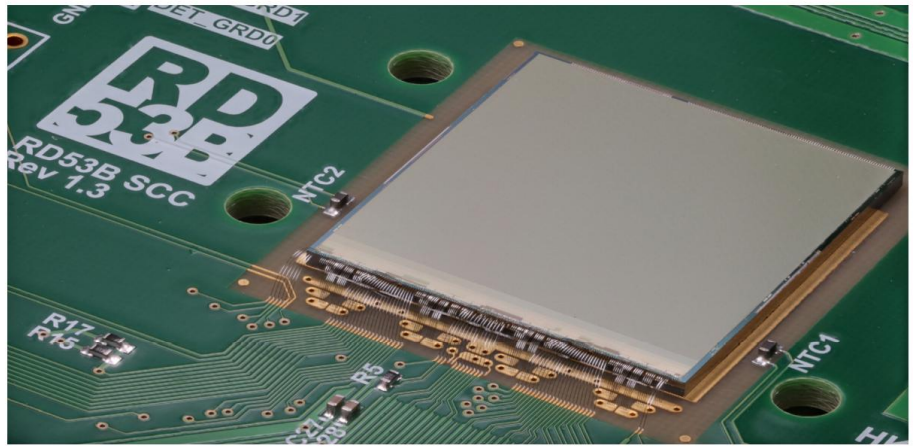
Backup

Rd53 readout chip for Pixel Detector

- Common design framework for ATLAS and CMS
- Rd53a is a prototyping chip
  - 3 analog FEs: Sync, Linear, Diff
- Rd53b
  - all RD53A elements with bug fixes, technical improvements where needed, and additional features
    - ShuntLDO, CDR/PLL, ...
    - Optimal data formatting and compression
  - ATLAS (ITkPix): 400x384 pixels, differential FE
  - CMS (CROC): 432x336 pixels, linear FE

Chip size (ATLAS)	$20 \times 21\ mm^2$
Pixel size	$50 \times 50\ \mu m^2$
High hit rate	$3\ GHz/cm^2$ (vs. FEI4: $400\ MHz/cm^2$)
High trigger rate	1 MHz with 12.8 μs latency
High data rate	up to 5.12 Gbit/s per chip
Low threshold	$\sim$ 600 e
Radiation tolerance	500 Mrad
Low power	$\sim$ 4 μA/pixel (analog/digital)



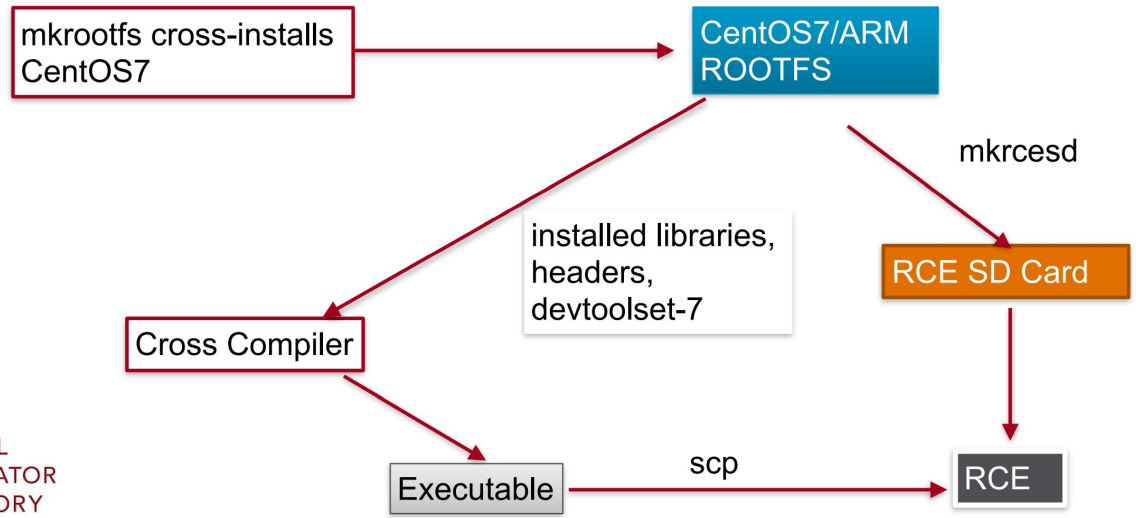


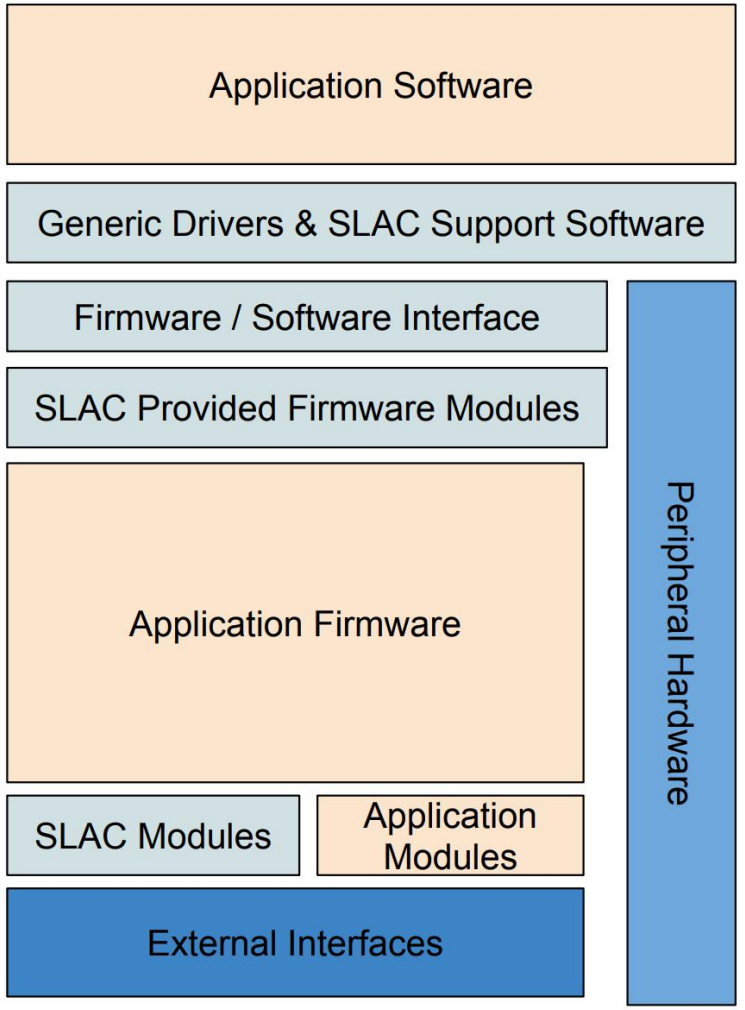
ITkPixV1 chip on Single Chip Card (SCC) [3]

https://indico.cern.ch/event/773447/contributions/3357417/attachments/1822775/2982085/AIDA_2020_Gaioni_v3.pdf

Software for Pixel RD53a/b readout

- Operating Systems:
  - **CentOS7:** 64bit for ZCU102, 32bit for RCE gen-3 (COB and HSIO2)
  - Arch Linux(abandoning), RTEMS without SMP support (abandoned)
  - Ubuntu LTS might be an option for next version ARM OS
- RCE Software Development Kit
  - host OS: CentOS7 x86/64
  - format and install root file system to SD card
  - Crosstool-ng to generate cross compilers for ARM and AARCH64





What is RCE(Reconfigurable Cluster Element)

Design of the SLAC RCE Platform: A general purpose ATCA based data acquisition system

Publisher: IEEE | Cite This | PDF

14 Author(s) R. Herbst ; R. Claus ; M. Freytag ; G. Haller ; M. Huffer ; S. Maldonado ; K. Nishimura ; C. O'Grady ; J. Pa… View All Authors

1 Paper Citation
125 Full Text Views

Abstract

Document Sections

I. Introduction

II. RCE Platform Architecture

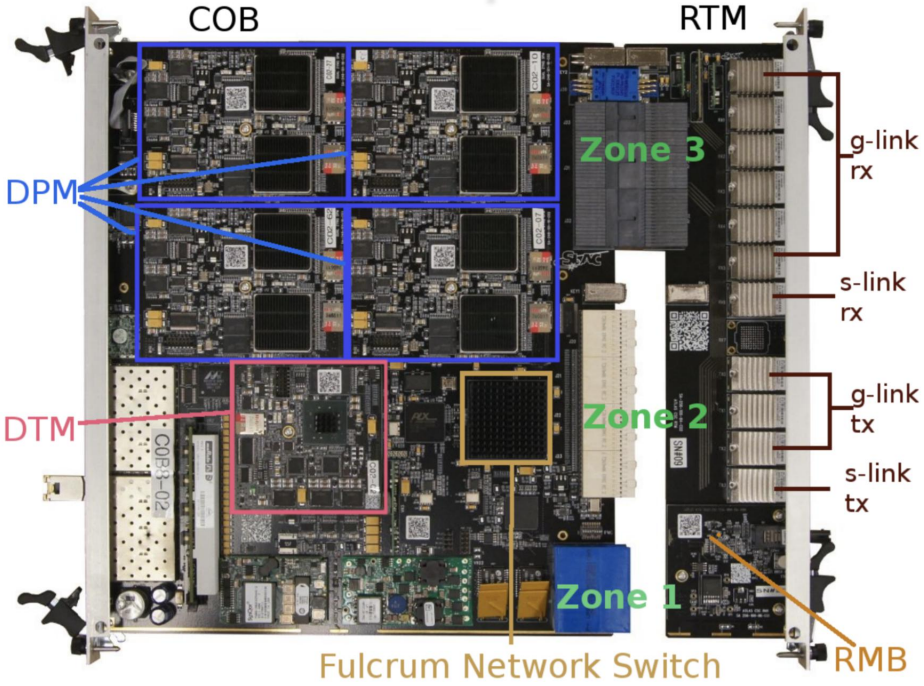
III. Packaging and Dataflow

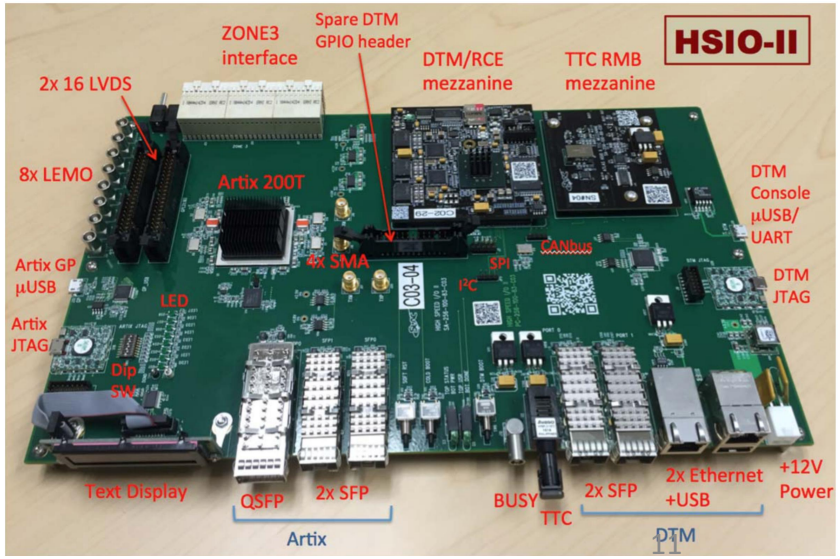
Abstract:
The SLAC RCE platform is a general purpose clustered data acquisition system implemented on a custom ATCA compliant blade, called the Cluster On Board (COB). The core of the system is the Reconfigurable Cluster Element (RCE), which is a system-on-chip design based upon the Xilinx Zynq family of FPGAs, mounted on custom COB daughter-boards. The Zynq architecture couples a dual core ARM Cortex A9 based processor with a high performance 28nm FPGA. The RCE has 12 external general purpose bi-directional high speed links, each supporting serial rates of up to 12Gbps. 8 RCE nodes are included on a COB, each with a 10Gbps connection to an on-board 24-port Ethernet switch integrated circuit. The COB is designed to be used with a standard full-mesh ATCA backplane allowing multiple RCE nodes to be tightly interconnected with minimal interconnect latency. Multiple shelves can be clustered using the front panel

https://ieeexplore.ieee.org/document/7431254

Old Gen RCE SoC Readout

- Reconfigurable Cluster Element (RCE) is a SLAC R&D project for a generic SoC DAQ/trigger system, starting from 2007
  - CPU and FPGA fabric within a same chip for a compact data passage
  - tight coupling between firmware and software
- Gen 3 (2013) based on Zynq-7000 SoC
  - **COB** (Cluster-On-Board): ATCA blade with custom RTM specialized for user applications, up to 4 DPMs and 1 DTM
  - **HSIO2**: standalone bench-top readout board, 1 DTM
  - RCE mezzanine card:
    - **DPM** (Data Processing Modules) and **DTM** (Data Transmission Module, network connectivity)
    - Xilinx Zynq-based SoC, multi-core ARM CPU, GB of memory, DMA interface to FPGA fabric, Versatile high/low-speed I/O interface



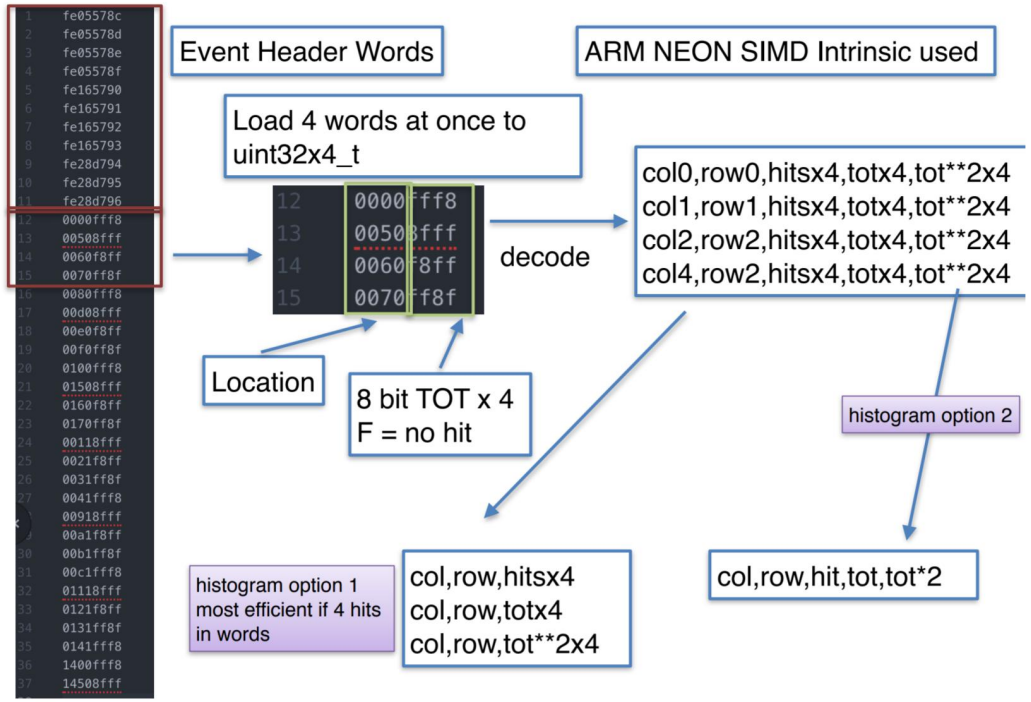


SLAC NATIONAL ACCELERATOR LABORATORY

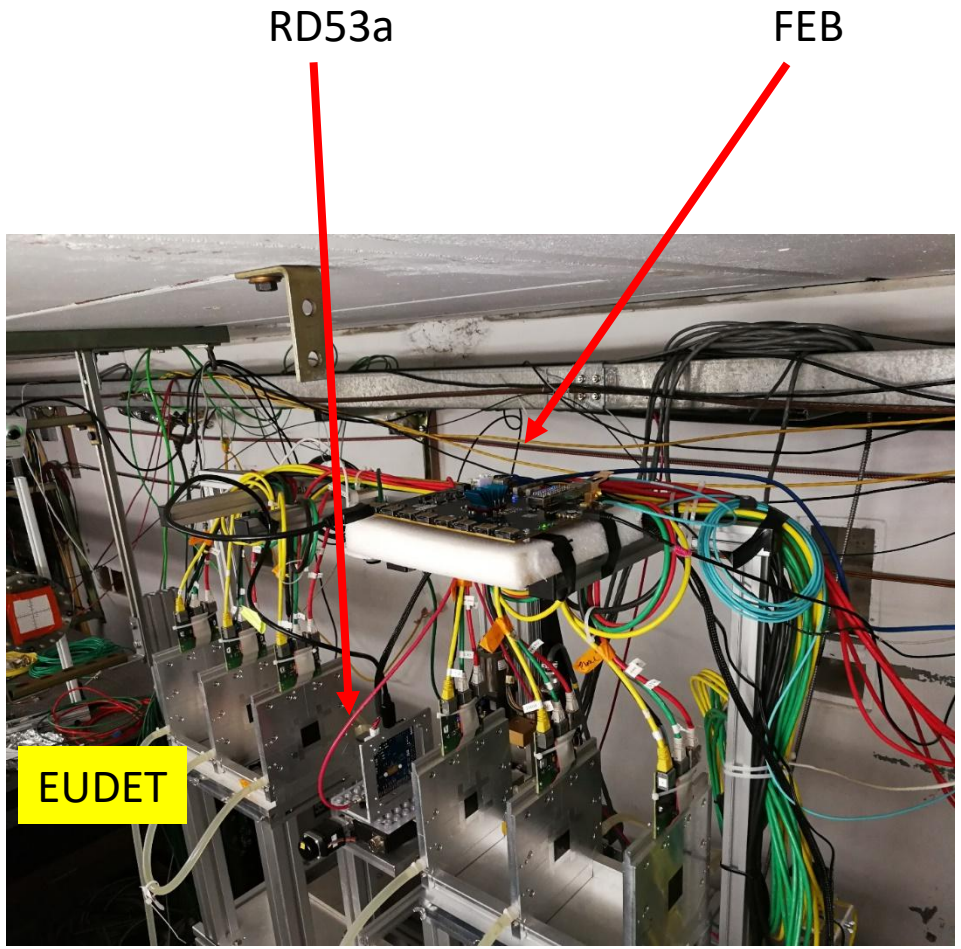
Rd53 chip calibration Performance Optimization

- only aim for Rd53 pixel chip
  - utilizing all the features of Rd53 for best performance
    - event header/ID in data stream => could async Tx and Rx during digital scan
    - high I/O throughput
  - pixel chip digital scan: 5.0 sec with vanilla ITk SW on RCE => 1.2sec after modification

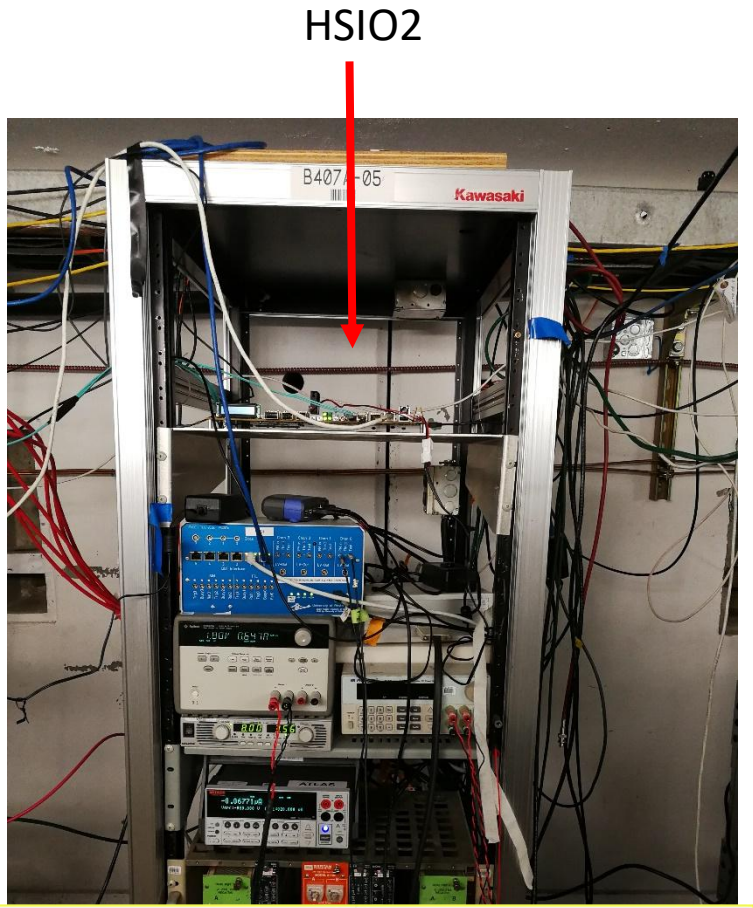
- ARM **SIMD**(single instruction multiple data)
  - improves CPU/software efficiency
  - Similarly, SSE for INTEL CPU
  - speed up raw data decoding by a factor of ~3
  - digital scan takes 0.7 sec with SIMD
  - (*raw Cmd+Data transferring needs ~0.4sec, limited by the chip I/O speed)
  - other types of scan got similar improvements

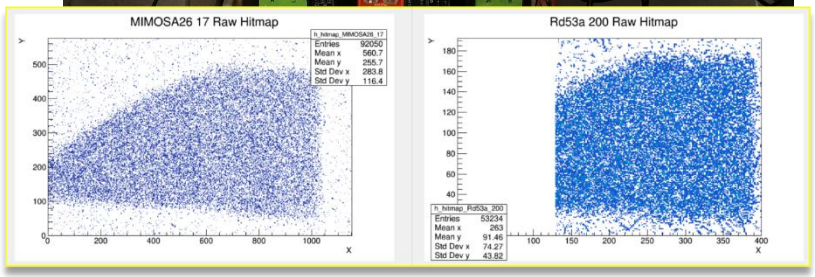


RCE for Pixel RD53a/b Testbeam



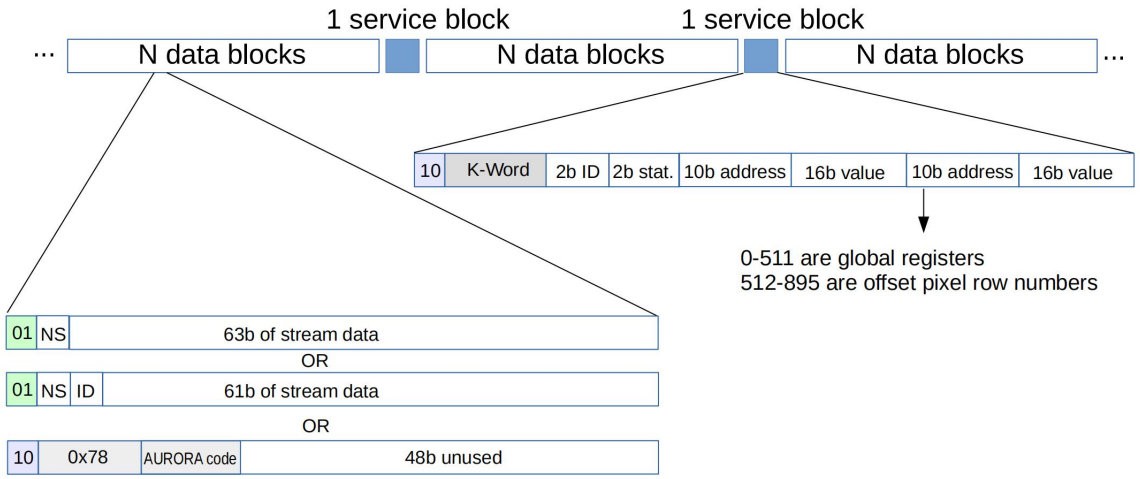
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=242295409





ATLAS ITkPixV1 (Rd53b)

10.2 Aurora and RD53B Data



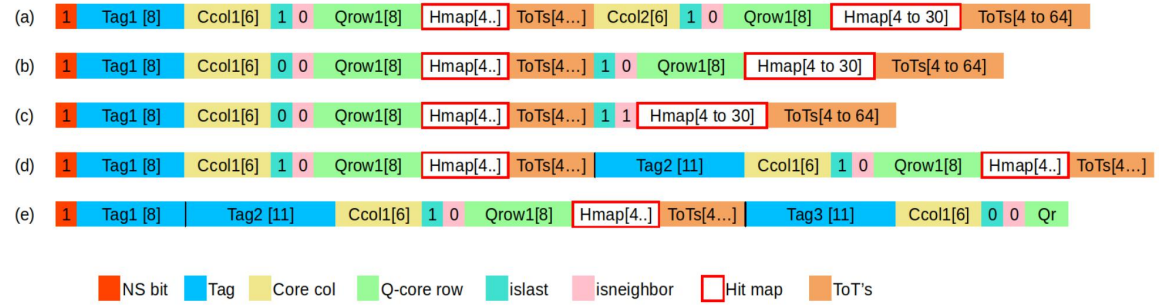


Figure 59: Examples of encoded stream data with no Aurora block boundaries shown: (a) one hit quarter-core each in two ccols (note last hit bit is set for both), (b) two separated quarter-cores hit in same ccol (last hit set only for second), (c) two neighbor quarter-cores hit in same ccol, (d) one hit quarter-core each in two different events, (e) an empty event followed by an event with one hit quarter-core, followed by another event. A color key to the field types is shown at the bottom. The number of bits in each field is shown in square brackets.

http://cds.cern.ch/record/2665301