DAQ-ROC4Sens

F.R. Palomo$^1$, F.Muñoz$^1$, R.Millán$^1$, P.Blanco$^1$, J.Jiménez$^1$, J.M.Hinojo$^1$·I.Vila$^2$

fpalomo@us.es

$^1$Departamento Ingeniería Electrónica, Escuela Superior de Ingenieros
Universidad de Sevilla, Spain

$^2$Instituto de Física de Cantabria
CSIC, Santander
In a Nutshell

• Backend DAQ ROC4Sens
• Project Description
• Technologies involved
• FPGA Logical Design
• Human Interface
• Custom Hardware
• State of the Project and prospectives
General Description

• USE-IFCA+PSI joint task
• Development of a DAQ backend for the Roc4Sens chip using Xilinx Zynq UltraScale technology
• Client-Server Architecture
• One card will serve up to 4 ROC + Detector assemblies + 4 Trigger TLU channels
• The UltraScale Hybrid FPGA runs the data protocols (I2C, fast Roc4Sens protocol) and also a full Ubuntu 14.04 LTS on the ARM 4-core.
• Fast Data Rate (ADC’s at a maximum of 170 MS/s , 12 bits, 1.1 GHz Bandwith)

• The main idea is to simplify the typical testbeam backend setup (AIDA2020 telescope) by using state of the art hybrid FPGAs (logic fabric for logical design + a set of multicore ARM processors) and high bandwith communications (Gigabit Ethernet mainly)
General Layout

- Three main boards:
  - UltraZed Board (UZB): comm services (TCP/IP on Gbit Ethernet, I2C, etc)
  - Custom Board (CB): ADC (x4), Trigger ports (x4), Level conversion (TTL/NIM)
  - UltraZed SOM (System On Module, UZ-SOM) supporting the Zynq UltraScale Hybrid FPGA

- 4 Roc4Sens served

- 4 Trigger channels served (from the Trigger Logic Unit, TLU)
• Hybrid FPGA: One logic fabric (Kyntex), 2 ARM R5 dual core (600 MHz), 4 ARM A53 quad core (1.5GHz).
• AXI internal fast bus for connection between programmable logic and microprocessors
• Shared memory for communications between cores
• ARM R5 for fast microprocessor tasks (no operating system, tipically proxy patterns)
• ARM A53 quad core able to run Ubuntu 14.04 LTS (Long Term Support)

With this computing power at hand the whole backend collapses to only one system, able to manage a full testbeam. The client remote computer (control room) is connected to the DAQ backend by an Ethernet cable.
DAQ Roc4Sens: State of the Project

- Logical Design (protocols in the Kyntex logic fabric): finished
- Local Application Software (on the ARM A53 quad cores): finished
- Client Interface (on the remote laptop): finished
- Custom Board (ADC’s x 4, TLU ports x 4): under manufacturing in a couple of companies
- We expect to have the whole assembly in the beginning of December
- Full system working (foreseen): February 2019
Logical Design

- Kyntex Logic Fabric: responsible of 2 sets of finite state machines:
  1. I2C (slow protocol)
  2. Roc4Sens proprietary (fast protocol)

Also responsible of the routing IP designs: local registers for control signals and AXI DMA (Direct Memory Access controller) to discharge data into the local RAM. LVDS/SE* services also enabled.

- Up to this moment we do not use the 2core ARM R5 but they are available for proxy tasks (data compression by analysis, data selection, etc).

- We make extensive use of the 4core ARM A53 to run
  - A full Ubuntu 14.04 LTS Operating System
  - The main DAQ-App, responsible of
    - Remote Procedure Calls
    - UDP/TCP data transfers
    - Full duplex data transfer (4x duplicated ports so two data transactions for each ROC are served)

*Low Voltage Differential Signaling/Single Ended signaling
Fast Protocol

- Partially Tested:
  - We can generate all Roc4Sens commands and we have signal captures (only at the FPGA)
  - Waiting for the Custom Board for Signal Integrity test
I2C communications

 DMA

• DMA tested
• In simulation, Acq. Speed is between 1.3 – 2.6 ms
Backend Server

SERVER (on ARM A53 4core, Zynq UltraScale)
- C++ Program
  - Multiple Threads (4)
  - RPC (Remote Procedural Calls) Server
  - Hardware DMA Manager
  - TCP/UDP Sockets Data Transfer Server (pixel matrix data transfer for each ROC-sensor assembly)

Communications
- PC (ports 3550-3553 & 3554-3557 for data transfer, full duplex for triggered and non triggered data transfer)
  - RPC for Command Transfer ✓
  - TCP/UDP Sockets for Data Transfer ✓
    - (handshake-transmission-close procedure)
- Programmable logic
  - Hardware DM (Direct Memory Access) ✓
  - AXI4 Fast Bus Transfer ✓

*from https://www.cs.rutgers.edu/~pxk/417/notes/08-rpc.html
Remote Interface (remote laptop)

- Runs on a remote laptop
- The Client Communicates with the backend by Remote Procedural Calls (RPC) and by UDP/TCP data transfer
- Four pixel displays
- Full set of configuration controls
- Shell available for Python scripts
- Full Duplex (8 ports, 2 associated to each ROC for 2 simultaneous data transfer, triggered and not triggered)
- Client/Server and Server/Client if necessary (architecture duality)
Custom Board

- Printed Circuit Board, responsible of ADC’s (x4), Trigger ports (x4) and NIM2TTL conversion
- HV bias pixel sensor goes in parallel by a dedicated cable (not in the PCB, it is safer)

- High frequency design (differential and single ended lines fully isochronous)
- Ten layers stacked
- Fabrication expected release on second week of December 2018
Custom Board

ADC 12C170 Features

- Resolution: 12 bits
- Sample Rate (Max): 170 MSPS
- SNR: 67.2 dB
- ENOB: 10.8 Bits
- SFDR: 85.4 dB
- Power Consumption (Typ): 715 mW
- Differential Voltage Input
- Input Voltage Level: 0 - 2.6 V (limited by ESD)
- Single Parallel Output CMOS
- Output Voltage Level: 1.8 V
- 1 channel

TTL-to-NIM CONVERTER

NIM-to-TTL CONVERTER
Conclusions

• DAQ-Roc4Sens at 85% of finalization
• Drastic simplification of the typical testbeam backend arrangement: only one backend acting as a server
• High Data Rate Transfer (>> Mbs), it allows data processing in real time
• Fully comprehensive interface (shell included)
• Availability expected on February 2019
Thanks for your attention

fpalomo@us.es