METHODOLOGY FOR THE DEVELOPMENT OF INTELLIGENT DATA AND IMAGE ACQUISITION SYSTEMS USING EPICS AND FLEXRIO TECHNOLOGY

D. Sanz\(^1\), M. Ruiz\(^1\), A. Bustos\(^1\), S. Esquembri\(^1\), E. Barrera\(^1\), E. Bernal\(^1\), R. Castro\(^2\), J. Vega\(^2\)

\(^1\)UPM, Universidad Politécnica de Madrid, Madrid, Spain; Grupo de investigación en Instrumentación y Acústica Aplicada
\(^2\)Asociación EURATOM/CIEMAT, Madrid, Spain
Outline

- Introduction
- RIO & FlexRIO Devices
  - Reconfigurable hardware based on FPGA
  - Interchangeable physical front ends for FlexRIO devices
- NIRIO-EPICS Device Support
  - I-RIO Driver
  - EPICS Device support for every RIO/FlexRIO configuration
- Design Methodology
  - Procedure
  - Example of FPGA implementation
- Applications
  - ITER
  - ESS Bilbao
- Conclusions
- Future works
- References
Introduction

• Nowadays large scale experiments require EPICS and reconfigurable DAQ devices based on FPGA.
• EPICS provides a powerful software framework.
• FPGAs provide reconfigurable hardware with deterministic data preprocessing capabilities.

Reconfigurable hardware:
Logic gates, flip-flops, RAM memory, I/O registers etc..
Introduction

- We propose a design methodology for RIO/FlexRIO devices, to be supported by a NIRIO EPICS Device Support (NIRIO-EDS) based on asynDriver.

- NIRIO-EDS will support every RIO/FlexRIO configuration.
RIO/FlexRIO Devices

FlexRIO device

Interchangeable Adapter modules

Write registers

Read registers

ADCs
DACs
Digital I/Os
CameraLink

FPGA
DMA
NIRIO-EDS
HOST IOC

PVs
ai ao di do mbbi Mbbo string Waveforms ...

Fixed I/O

Write registers

Read registers

ADCs DACs Digital I/Os
NIRIO-EDS based on asynDriver

- **I-RIO Driver** detects in run time FPGA configuration, and FPGA I/O registers and DMA channels are dynamically interfaced with EPICS.
Design Methodology for RIO/FlexRIO devices

- Records database file is **generated automatically** using FPGA mapped resources. Including all files in EPICS software unit, the IOC is ready to control & monitor the RIO/FlexRIO devices.

**Step 1**
Scientific requirements

**Step 2**
Electronics specifications: Signal conditioning, AI/O, DI/O sampling rates, preprocessing algorithms, etc.

**Step 3**
RIO/FlexRIO configuration with LabVIEW FPGA templates
FPGA (RIO/FlexRIO) configuration example

- Write registers
  - DAQStartStop
  - SamplingRate0
  - GroupEnable0

- Read registers
  - InitDone
  - FPGAVIversion
  - Fref
  - DeviceType
  - DeviceTemp

- FPGA
  - ADCs
  - DACs
  - Digital I/Os
  - CameraLink

- Interchangeable Adapter modules

- Host
- IOC
- NIO
- NIRIO-EDS

- DMA
  - DMA PCI/PCIe to CPU
  - FIFO
  - NCHperDMATtoHOST0

- Processing algorithm
  - auxAI32_0
  - auxAI32_1
  - auxAI32_2
  - auxDO_0

- State machine

- BUS PXI trigger lines
- PXI_IN0
FlexRIO configuration
FlexRIO configuration

1. Fill each of the next fields according to the required DAQ implementation.

1. a. Introduce the firmware version for this implementation.

1. b. Introduce the reference time clock for the DAQ acquisition module.

1. c. Available profiles: DAQ--->0; IMAQ--->1

1. d. Number of channels per DMA.

1. e. Introduces the number of waveform generators

1. f. If FlexRIO adapted module is required, then configure "TRUE" this constant, it is required include the IO Module Initialization replace for the true constant. If not configure it "FALSE"

1. g. Introduce "here below" the FIFO Clear method used in all the project, and select "Clear" method for every FIFO used in the FPGA design.
FlexRIO configuration

Step 2. By every DMA Channel, it is required a GroupEnable[0-N], and SamplingRate0

Step 3. This sequencer struct should be replicated in case of more DMA channels required.

Step 3a. If processing is configured "FALSE", Remove the corresponding Processing Timed loop.
FlexRIO configuration

Step 4. Add every hardware logic algorithm, processing using auxiliary registers, using name convention.

100 MHz Clock

Waveform generator. Change all the N numbers of the controls by the number desired. Every control must have the same number suffix.

Accelerometer
PhaseShift
SigAmp
AOEnable0
AO(UpdateRate0)

SimulatedSignal
SimulatedAO0

Step 5. If is programmed Processing, it is possible to use this timed loop for data processing.
Use case applications

- ITER PXIe Fast controller prototype for data and Image acquisition (cameralink).
Use case applications

- ITER Fission chamber diagnostic use case application based on FlexRIO technology: It integrates deterministic diagnostic into the FPGA (4 ADC sampling at 125MS/s). Processing to obtain counts, RMS and cancelling implemented in the FPGA.
Use case applications

- ESS Bilbao. NIRIO-EDS will support the RIO PXI-7852R for the control and monitoring of the Ion Source Hydrogen Positive (ISHP).
Other Applications

• Integration of Hardware clock into the RIO/FlexRIO devices synchronized with IEEE1588-V2 (tenth of nanoseconds accuracy)
  – It allows real-time timestamping in the DAQ for all acquired data (or blocks) without CPU intervention. Very useful for timestamped data streaming.
  – It requires a PXI device compliant with PTP-V2.
Conclusions

• We have defined a design methodology for implementing intelligent data and image acquisition into RIO/FlexRIO devices, supported by the NIRIO-EDS.

• We have developed LabVIEW patterns and libraries, that permit configure RIO/FlexRIO devices in an easy way.

• The NIRIO-EDS will support every RIO/FlexRIO configuration (implementation), without rewriting the device support.
Next Steps

• We will communicate soon to EPICS community the availability of this EPICS device support, and all RIO/FlexRIO hardware description templates and libraries (in LabVIEW for FPGA code).
• We are implementing direct data transfer communication among FlexRIO devices and GPUs without CPU intervention. This application will also include an EPICS device support to use the GPU.

Diagram:
- CPU (Controller)
- PCIe 1.1 x4 1GB/s
- PCIe 2 X4 2GB/s
- NVIDIA TESLA
- DMA direct to the GPU
- GPU
- NI PXie 7962R + NI CameraLink adapter module
- EoSens_3CL_MC3010
- PCI Express 1.1 x4 1GB/s
- PCI Express 2 X4 2GB/s
- CPU (Controller)
- NVIDIA TESLA
- DMA direct to the GPU
- GPU
- NI PXie 7962R + NI CameraLink adapter module
- EoSens_3CL_MC3010
References


METHODOLOGY FOR THE DEVELOPMENT OF INTELLIGENT DATA AND IMAGE ACQUISITION SYSTEMS USING EPICS AND FLEXRIO TECHNOLOGY

D. Sanz¹, M. Ruiz¹, A. Bustos¹, S. Esquembri¹, E. Barrera¹, E. Bernal¹, R. Castro², J. Vega²

Technical University of Madrid

Thank you very much for your attention!! questions?