IRIO Technology: Developing Applications for Advanced DAQ Systems using FPGAs

Mariano Ruiz, S. Esquembri, A. Carpeño, J. Nieto, A. Bustos, E. Bernal, D. Sanz, E. Barrera

mariano.ruiz@upm.es

Universidad Politécnica de Madrid
Outline

• Motivation
• RIO, FlexRIO and cRIO Devices
• Development cycle
• IRIO Project
  – IRIO Library
  – NIRIO-EPICS Device Support
  – IRIO NDS C++ classes
• Design Methodology
• Applications
  – ITER fission chamber
  – Image acquisition system
  – Advanced applications: using GPUs for ITER diagnostics
• Conclusions
Motivation

- FPGAs provide reconfigurable hardware with deterministic data preprocessing capabilities
- Graphical tools such as LabVIEW for FPGA reduces development and integration time
- The combination of both technologies with EPICS simplifies the development of complex control, data acquisition and processing systems

IRIO is a set of software tools simplifying the integration of RIO devices in EPICS
The developer defines the functionality programming the FPGA.

EPICS connects a user defined device with PVs for configuration and supervision.
Development cycle: LabVIEW for FPGA

1. LabVIEW VI
2. LabVIEW FPGA
3. VHDL
4. Bitfile
5. FPGA Target

FPGA project developer

User Generated
Auto Generated

XILINX compiler
Reducing development time in FPGAs using templates

This module implements Data Acquisition using 1 DMA. If more is required, then more modules should be added. The new number input, and so on. Correspondingly new FIFOs in 16

Step 3. If it is required a new waveform generator replicate this VI, with correlative numbers in the VI number input.

Note: Every waveform generator module that this one uses look-up tables (memory), take into account the FPGA resources spent on it. If it is not required, then remove it.

DMA Channel 1 Frame Generation & Transfer Control

DMA0 Controller

DMA Enable

Sampling Rate

Daq Start Stop

Debug Mode

DMA0 Data

DMA Channel 1

Frame

DMA Start Stop

Debug Mode

DMA0 Data

DMA Enable

Sampling Rate

Daq Start Stop

Debug Mode

DMA0 Data

DMA Channel 1

Frame
Using RIO devices in Linux

1. LabVIEW VI
2. FPGA project developer
3. VHDL
4. Bitfile
5. XILINX compiler
6. C/C++ application

- LabVIEW FPGA
- User Generated
- Auto Generated

- FPGA
- RIO/FlexRIO Device

- FPGA Target
- NI RIO library
- NI RIO kernel modules
- Linux User space
- Linux Kernel space

- Linux
- C API

- Hardware
- 1-5

- FPGA project developer
- User Generated
- Auto Generated
**IRIO Project: IRIO Library**

- Identification of the resources implemented in the FPGA
  - The Design Rules document describes the rules for the FPGA implementation
- Provides an API simplifying the interface with the FPGA.
  - Access to FPGA registers
  - Analog input
  - Digital I/O
  - DMA acquisition
  - Image acquisition using camera link
    - Serial line for camera configuration
  - Signal Generation (DDS)
IRIO resources mapping

1. irio_initDriver("test", "rio_serial", "rio_model", "FlexRIOMod6581", "V1.0", 1, bitFilePath, bitFilePath, &DrvPvt, &status);

2. irio_setAuxAO(&DrvPvt, 1, 250, &status);

3. irio_getAuxDI(&DrvPvt, 0, &aValue, &status);

NI RIO API Library

NiFpga_WriteU16(session, 0x1004, 1250);

NiFpga_ReadBool(session, 0x10F0, &value);

LabVIEW Code

VHDL

XILINX compiler

FPGA bitfile

FPGA header file

Adapter Module

IRIO Library

Software-hardware interface

Resource mapping
IRIO Project: EPICS driver using asynDriver

- EPICS device driver using asynDriver implementation for RIO devices (FlexRIO and cRIO) using IRIO library
  - Automatically connects the PVs with FPGA resources using IRIO library
- If the user changes the FPGA design no compilation is needed
- ITER SDD generates the complete software unit
IRIO Project: C++ classes for Nominal Device Support

- Nominal Device Support approach defines a set of classes and PVs to be used for EPICS driver implementation.
- NDS-irio is the set of NDS extended classes to use FlexRIO devices
- Simplify the implementation of EPICS device support for FlexRIO using NDS

User Space

EPICS Device Support

Application Program

NDS-IRIO

IRIO Library

FPGA C API

Kernel

NI RIO Linux Device Driver (kernel)

Hardware (I/O Module)
Design Methodology

**Windows Host**

1. FPGA project developer
2. LabVIEW VI
3. VHDL
4. Bitfile
5. FPGA Target
6. XILINX compiler

**User Generated**

**Auto Generated**

**Linux Host with CCS**

6. IRIO user

- **Use IRIO Library**
- **IRIO user application**

**Epics core**

- **CA**
- **NI-RIO EPICS**
- **NDS-IRIO**
- **IRIO Library**

**Linux User space**

- **NI RIO library**

**Linux Kernel space**

- **NI RIO kernel modules**

**Hardware**

- **FPGA**
- **RIO/FlexRIO Device**

**Codac Core System**

- **Using NDS?**
- **NO**
- **YES**

- **EPICS application?**
- **NO**
- **YES**

- **EPICS device support implemented with NDS**
- **Use NDS-IRIO**

- **EPICS device support implemented with ASYN**
- **Use NDS-IRIO**

- **NI-RIO EPICS Device driver**

**Use IRIO Library**

**IRIO user application**

**Using NDS?**

**YES**

**NO**

**1-5**

**I-O core**

**IOC app**

**C/C++ application**

**FPGA project developer**

**LabVIEW FPGA**

**VHDL**

**Bitfile**

**FPGA Target**

**XILINX compiler**

**User Generated**

**Auto Generated**

**LabVIEW FPGA**
ITER Fission chamber diagnostic use case application based on FlexRIO technology

- Integrate deterministic diagnostic into the FPGA (4 ADC sampling at 125MS/s). Data processing to detect/count pulses, RMS, and campbelling.

![Diagram of data processing flow]

1. **125MHz data acquisition**
   - AI[0-2]
   - Software trigger
   - Hardware trigger

2. **Low Pass Filter**
   - ON/OFF activation
   - Cutoff freq selectable
   - N factor configurable

3. **Downsampling**
   - N factor configurable

4. **FPGA Real Time Preprocessing**
   - DMA transfers to HOST
   - Measurement info
   - DMA 0
   - DMA 1
   - Pulses detection
   - Current
   - Raw data acquired
   - TimeStamps generation
   - Pulse info:
     - sample peak detection
     - Width & heigh

NI PXIe-7966R + 5761

14 bit at 250MS/s per channel

125MHz data acquisition

I/O Registers

| Pulses detection | Campbelling | Current |

Every 1ms for

PXI trigger line 2

20th Real Time Conference 2016 June 9th
ITER PXIe Fast controller: Image acquisition (cameralink).

Can we add more processing capabilities? Is it possible to add a GPU?
• NI-RIO Linux Device Driver modified to implement direct DMA from FPGA to GPU
Conclusions

• We have defined a design methodology for implementing advanced data and image acquisition applications with RIO/FlexRIO devices, integrated with EPICS using IRIO software
• We have developed different LabVIEW/FPGA patterns and libraries for RIO devices
• It is not necessary to rewrite or even recompile the EPICS device support for every cRIO/FlexRIO configuration
• IRIO tools integrated in ITER CODAC Core System V5.2 (February 2016)
• IRIO tools are GPLV2
• Current users of IRIO:
  – ITER Diagnostics use cases, KSTAR project, Russian DA (cRIO)
  – ESS Bilbao
IRIO Technology: developing applications for advanced DAQ systems using FPGAs

Mariano Ruiz & Sergio Esquembri
Universidad Politécnica de Madrid, Spain

mariano.ruiz@upm.es

Technical University of Madrid

Thank you very much for your attention!!
questions?