Interfacing EPICS IOC and LabVIEW for FPGA Enabled COTS Hardware

Arun Veeramani, Kyle Tetmeyer
National Instruments

Rok Sabjan, Anze Zagar

Cosylab

Several attempts have been made to integrate EPICS functionality with National Instruments LabVIEW. With existing EPICS code, labs want to reuse the code while still being able to use LabVIEW to interface with FPGA enabled embedded controllers and other COTS hardware. In this paper, we will show how we can run EPICS IOC simultaneously with LabVIEW on VxWorks based hardware. We will go into the implementation details and the benchmarks that will be obtained from the LANSCE-R project at Los Alamos National Labs. We will also examine ways to implement a Channel Access(CA) server natively in LabVIEW. This will open up the opportunity to use a variety of IO and different operating systems that LabVIEW can interface with. The native LabVIEW CA server will implement all Channel Access functionality exposed by a standard EPICS IOC such as synchronous and asynchronous publishing of data, alarm processing, and response to connection requests by CA clients. We will finally cover the programming of FPGA allowing for custom solutions.

Agenda

- EPICS IOC on COTS hardware
  - Current challenges
  - Potential solution
- CA server in LabVIEW
- Test results from LANL

Current Challenges

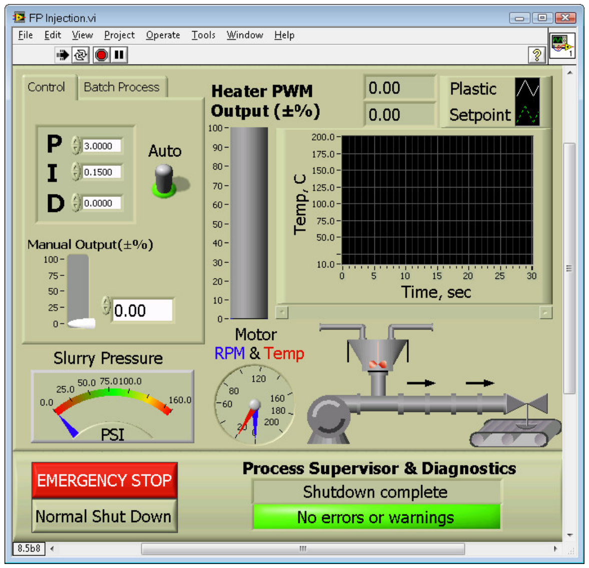
- Limited budget and man power
- Limited man power
- Custom driver development
- Challenging to develop driver for all COTS hardware
- Shorter development time

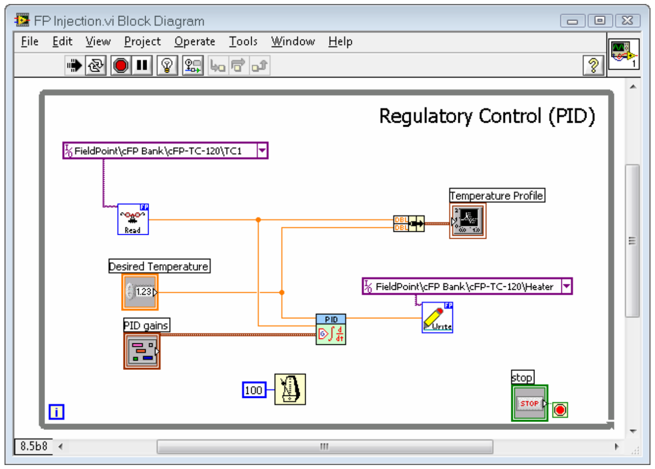
Proposed Solution

- Interface with commercially available software – LabVIEW
  - Advantage: Enable use of wide range of COTS hardware
- Options
  - #1 – EPICS IOC communicating with LabVIEW on VxWorks
  - #2 – Native Channel Access Server in LabVIEW
  - #3 – Dedicated EPICS hardware module

What is LabVIEW?

- Graphical programming language
- Natural multi-core support
- LabVIEW FPGA to program FPGAs without VHDL / Verilog
- Interface with COTS hardware
- Connectivity to different devices
    - OPC, Modbus, TCP/IP, UDP, Serial

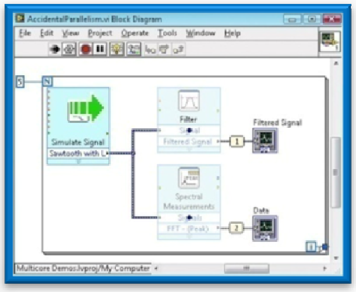



FP Injection.vi Block Diagram
File Edit View Project Operate Tools Window Help
Regulatory Control (PID)
FieldPoint\cFP Bank\cFP-TC-120\TC1
Read
Temperature Profile
Desired Temperature
FieldPoint\cFP Bank\cFP-TC-120\Heater
PID
Write
PID gains
100
stop
STOP

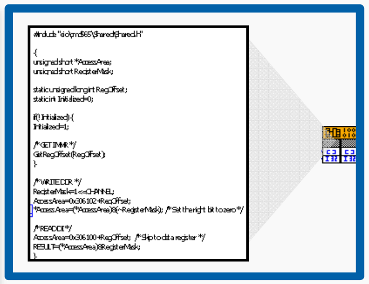
NATIONAL INSTRUMENTS™

High-Level Design Models

Data Flow

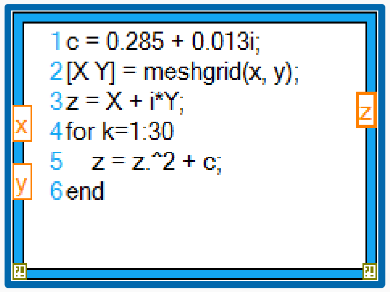

AccidentalParallelism.vi Block Diagram
File Edit View Project Operate Tools Window Help
Simulate Signal
Filter
Filtered Signal
Spectral Measurements
Data

C, VHDL



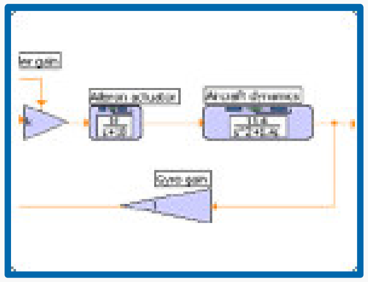
```
#include "sio\mod865\Shared\Shared.H"

{
unsigned short *AccessArea;
unsigned short RegisterMask;

static unsigned long int RegOffset;
static int Initialized=0;

if(!Initialized){
Initialized=1;

/*GET IMMR */
GetRegOffset(RegOffset);
}

/*WRITE DDR */
RegisterMask=1<<CHANNEL;
AccessArea=0x306102+RegOffset;
*AccessArea=(*AccessArea)&(~RegisterMask); /*Set the right bit to zero*/

/*READ DDI*/
AccessArea=0x306100+RegOffset; /*Skip to data register */
RESULT=(*AccessArea)&RegisterMask;
}
```

Textual Math

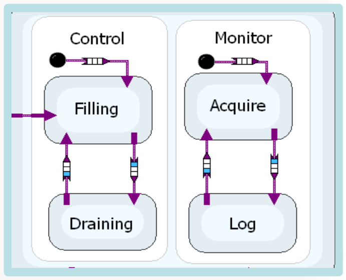


```
c = 0.285 + 0.013i;
[X Y] = meshgrid(x, y);
z = X + i*Y;
for k=1:30
   z = z.^2 + c;
end
```

Simulation



Statechart



NATIONAL INSTRUMENTS LabVIEW™

Graphical System Design Platform

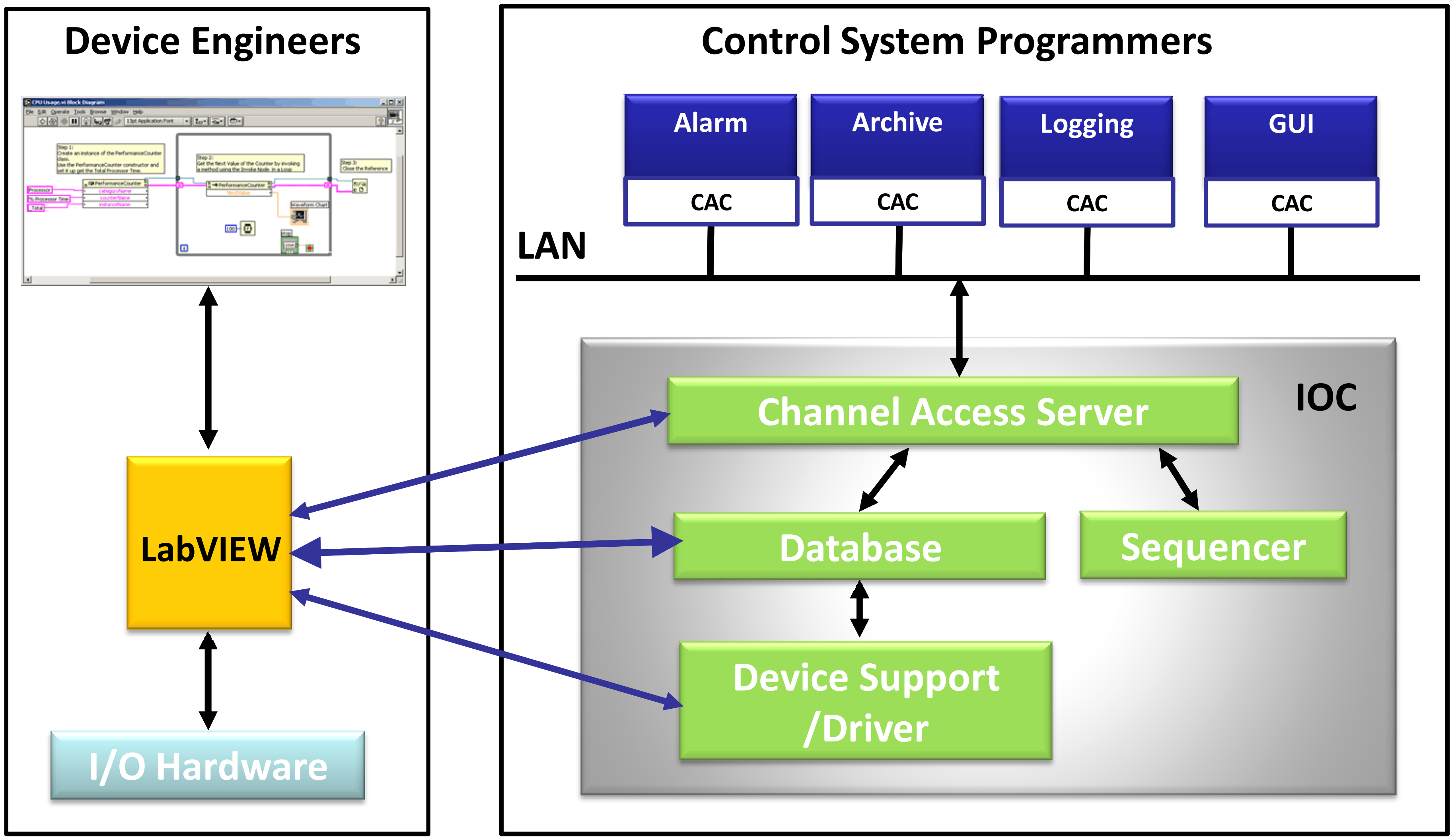
Desktop

Real-Time

FPGA

Microprocessors

EPICS IOC Architecture

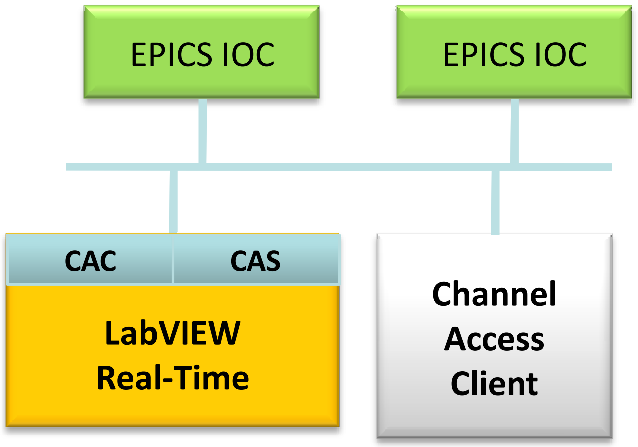


Option #1: EPICS on FPGA-based PACs

- Project initiated by the LANL's LANSCE-R upgrade
- Specific requirements due to existing control systems, infrastructure and people skills at LANL
  - EPICS IOC has to run on cRIO (VxWorks Knob software)
  - Graphical programming is required for LV RT and FPGA
  - 2-way communication between EPICS and LV RT
- Existing solutions; no solution would be directly applicable, changes needed
  - **SNS Shared memory solution**
  - LV's DSC Module
- LV-RT was designed "master" – it's tasks have higher priority

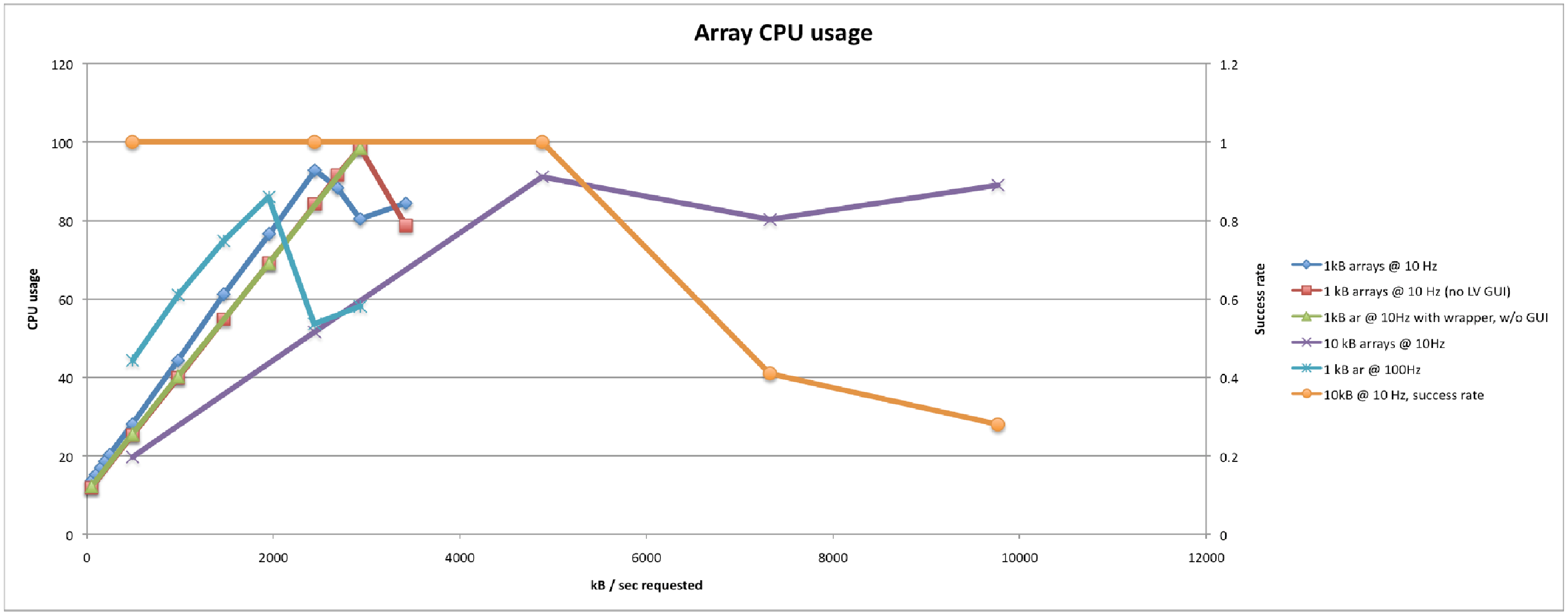
Advantages of Option #1

- Ability to use existing EPICS IOC records and function blocks
- Use LabVIEW to:
  - Perform any “left out” functions
  - Abstract complexities of FPGA



Prototype Implemented

- EPICS cohabitating with LabVIEW RT and cRIO
- Inter-thread communication
- Three data types supported in polling mode
- Arrays were tested
- BSP modified to support NFS (network boot, save/restore) and NTP
- 2-5 MB/sec roundtrip rates were reached



Main Issues (1/2)

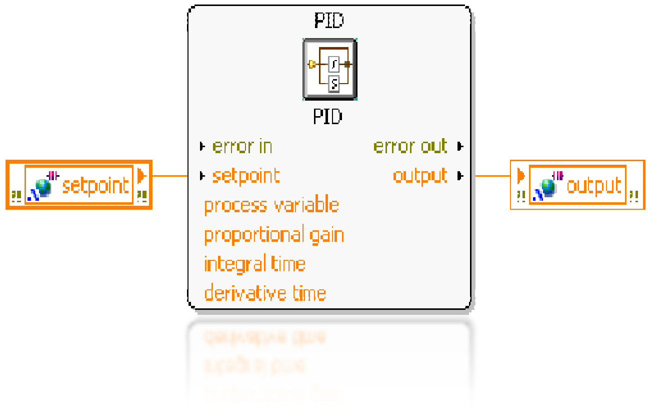
- Core (Shared Memory API) needs changes
  - Shared Memory API Independent vs. LV/EPICS specific?
  - Event-driven communication support (callbacks on LV side, I/O interrupt scanning on EPICS)
  - Timestamp support
  - Alarms?
  - Telnet server
- Re-implementation of EPICS side with Asyn driver?
  - Can use all device supports that talk with Asyn ports (EPID, AsynRecord etc.)
    - Easier application development with extended debugging, more structured code
    - Proper setpoint initialization
  - Support all basic EPICS datatypes
  - Support most common records (still missing: bi,bo, mbbi/o, mbbi/oDirect)

Main Issues (2/2)

- Development processes
  - Mapping of records to (sub-) VIs
    - By discipline (naming conventions)?
    - Automatic (two-way generator)?
  - Shared Memory Configuration (size etc.) inside LV
  - Manage device addressing (aliases?)
  - Documentation, tutorials, example templates
- Deployment
  - Startup and shutdown procedure
  - EPICS life-cycle is a problem (reboot required for stopping)
  - Binaries of EPICS modules
  - Boot from network / CF
- Other platforms
  - Similar work can be done for LV-RT - EPICS on PXI
    - But the operating system is different

Option #2: CA Server in LabVIEW

- Program required functionality in LabVIEW
- Expose required data points as PVs using CA server
- Ability to run on any LabVIEW enabled platform



Option #3: Dedicated EPICS hardware

- Plug-in hardware card with separate CPU
- EPICS crash does not affect LabVIEW RT
- Freedom: do with EPICS as you wish, use any CS and OS
  - Well defined and supported API to communicate with LV-RT
  - Customer will have free hands
- Faster performance (2 CPUs)

Summary

- EPICS interface with LabVIEW to tackle challenges
- Different options available based on needs
- Wide array of COTS hardware available through LabVIEW