EPICS IOC On SoC FPGA based LLRF Controllers

Shriraj Kunjir
Radio Frequency Controls Engineer

MICHIGAN STATE UNIVERSITY

U.S. DEPARTMENT OF ENERGY | Office of Science

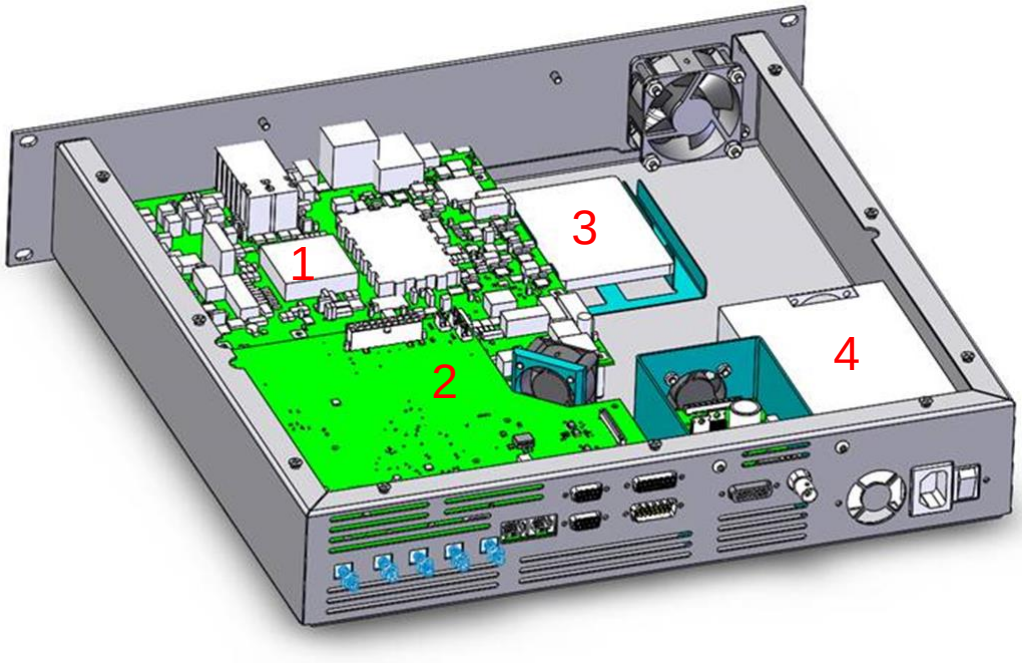
This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University. Michigan State University operates FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.

Outline

- LLRF Controller
- LLRF Controller SoC FPGA
- EPICS IOC Implementation
- Test
- Advantages
- Challenges
- Summary

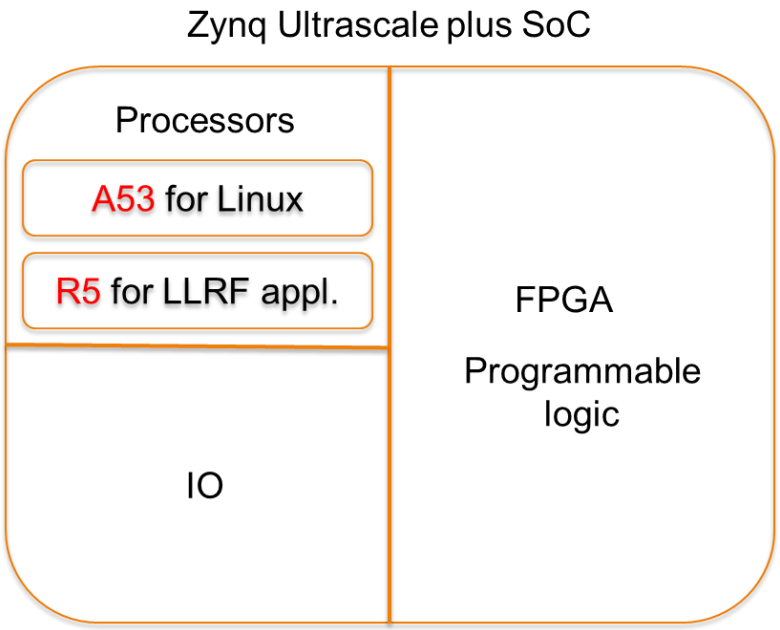
LLRF Controller

- Designed and developed for
  - Future FRIB 644 MHz upgrade cavity
  - MSU ultra-fast electron microscopy UEM 1 GHz cavity
- Currently installed and operational at MSU UEM lab.
- Specs -
  - 1x Xilinx FPGA board (1)
  - 1x RF front-end board (2)
  - 1x 500 GB solid-state drive (3)
  - 1x 1U Power supply (4)



LLRF Controller SoC

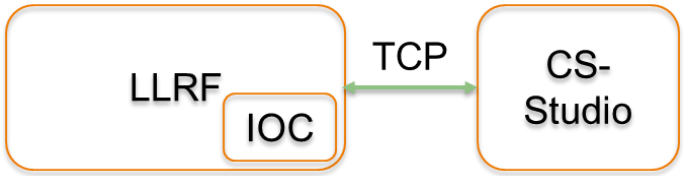
- Specs -
  - Arm Cortex A53 Quad-core processor
    - » ARM v8 architecture
    - » Total CPU frequency 1.2 GHz
    - » 4 GB RAM
    - » Debian 11 Bullseye
  - Arm Cortex R5 dual-core real-time processing unit
  - IO peripherals
    - » PCIe
    - » Gigabit Ethernet
    - » USB 3.0
    - » SATA
    - » SD
  - FPGA
    - » 600k LUTs
    - » 2520 DSP slices
    - » 24x 16.3 Gb/s Transceivers
    - » 328 I/O pins



EPICS IOC Implementation

- In the current implementation, LLRF controllers communicate with IOC servers over UDP.
- Added latency due to network traffic.
- UDP connection is unreliable and lossy.
- Asyn port driver that uses EPICS timers as its main communication mechanism.

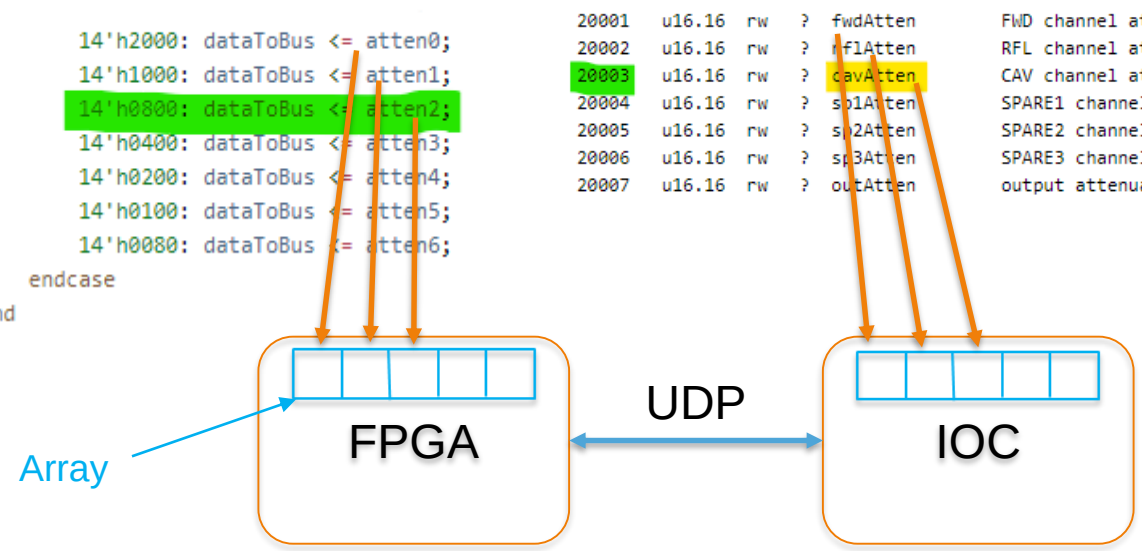
- In the new implementation, there's no need for UDP since the IOC server runs on the LLRF controller.
- Minimum latency since IOC server has access to FPGA registers via Linux userspace.
- Asyn driver adapted from softGlue driver available inside synApps.



FPGA – IOC interface (old)

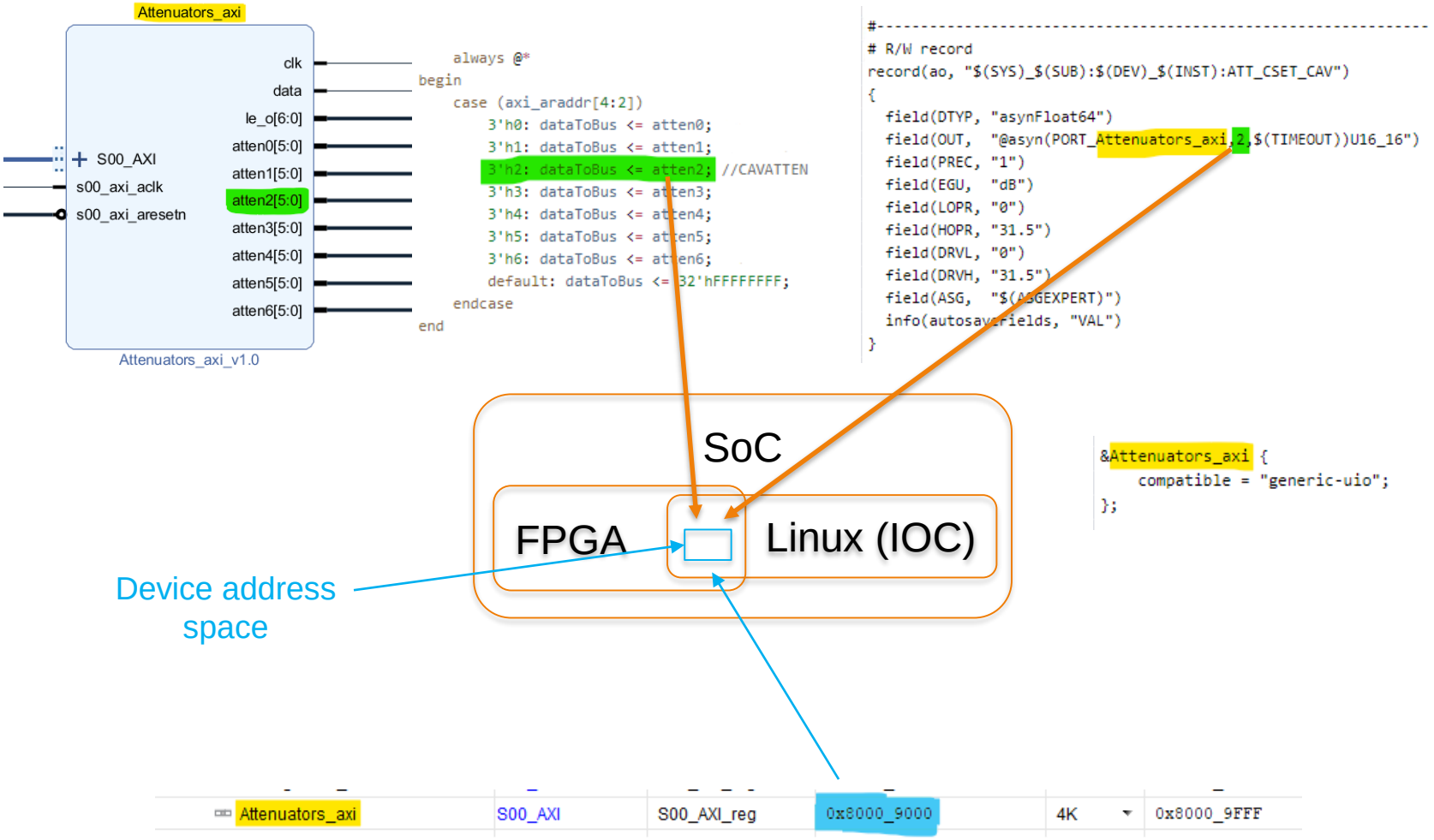
```
always @*
begin
    case (Bus2IP_RdCE)

        14'h2000: dataToBus <= atten0;
        14'h1000: dataToBus <= atten1;
        14'h0800: dataToBus <= atten2;
        14'h0400: dataToBus <= atten3;
        14'h0200: dataToBus <= atten4;
        14'h0100: dataToBus <= atten5;
        14'h0080: dataToBus <= atten6;
    endcase
end
```

```
20001   u16.16  rw   ?  fwdAtten      FWD channel attenuator value (in dB) (truncates to 1/2 dB steps)
20002   u16.16  rw   ?  rflAtten      RFL channel attenuator value (in dB) (truncates to 1/2 dB steps)
20003   u16.16  rw   ?  cavAtten      CAV channel attenuator value (in dB) (truncates to 1/2 dB steps)
20004   u16.16  rw   ?  sp1Atten      SPARE1 channel attenuator value (in dB) (truncates to 1/2 dB steps)
20005   u16.16  rw   ?  sp2Atten      SPARE2 channel attenuator value (in dB) (truncates to 1/2 dB steps)
20006   u16.16  rw   ?  sp3Atten      SPARE3 channel attenuator value (in dB) (truncates to 1/2 dB steps)
20007   u16.16  rw   ?  outAtten      output attenuator value (in dB) (truncates to 1/2 dB steps)
```



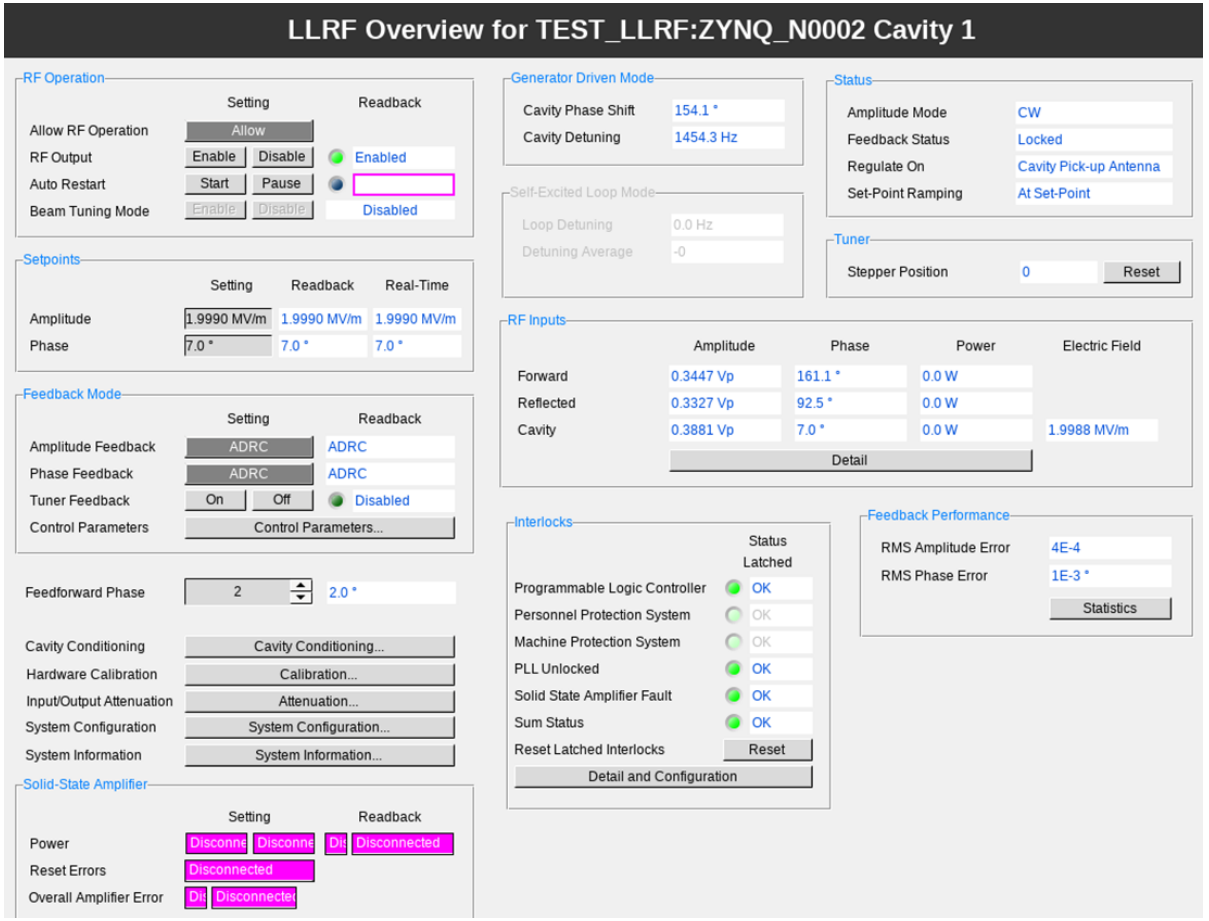

```
#-----------------------------------------------------------------
# R/W record
record(ao, "$(SYS)_$(SUB):$(DEV)_$(INST):ATT_CSET_CAV")
{
  field(DTYP, "asynFloat64")
  field(OUT,  "@asyn($(SYS)_$(SUB):$(DEV)_$(INST),0,$(TIMEOUT)) cavAtten 0x20003 Float64 U16_16")
  field(PREC, "1")
  field(EGU,  "dB")
  field(LOPR, "0")
  field(HOPR, "31.5")
  field(DRVL, "0")
  field(DRVH, "31.5")
  field(ASG,  "$(ASGEXPERT)")
  info(autosaveFields, "VAL")
}
```

FPGA – IOC interface (new)



Test

- LLRF controller on test network.
- Work in progress to add more PVs, OPI pages, archiver support and features.
- 25 CS-Studio screens connected to IOC.
- CPU load - ~ 0.02%
- Memory utilization - ~ 0.03%
- 350 PVs connected.



Advantages

- Useful in lab environments where it is difficult to have a standalone IOC server.
- Targeted maintenance.
  - Reduce risk of affecting other devices and IOCs.
- Distributed computing.
  - Split hardware computing resources.
- Direct Memory Access (DMA) support for fast data capture and plotting.
  - Gives hardware-level access to data and allows for interrupts to the IOC from the FPGA.
- Reduction in network traffic.
  - Eliminating unreliable UDP connection frees up network resources.
- Reliable TCP connection.
- Reliable fault tolerant fallback scenarios for boot mechanism.
  - Supports multiple fallback boot image files in case of primary boot image corruption.
- Sufficient local storage for log files and data in case of remote archiver failure.

Challenges

- Increase in total number of IOC servers under management.
- Maintenance of spares.
  - Keep up-to-date with latest packages/versions.
- Central configuration management system (Puppet) not supported for Debian on Arm 64-bit architecture yet.
  - Hopefully that changes in the future.
  - Develop alternate mechanism in the meanwhile.
- Define network/system architecture with all stakeholders (controls group, business IT, system administrator).
- Determine Arm 64-bit architecture support for all Debian packages currently installed on IOC virtual machines.

Summary

- Proof of concept has been demonstrated.
- Future work involves addressing challenges and taking advantage of interrupt and DMA support.
- More discussions need to be done to determine maintenance of the LLRF controllers running IOCs.