The ESS FPGA Framework and its Application on the ESS LLRF System

C. Amstutz[1], M. Mohammednezhad[2], M. Donna[1], A. J. Johansson[1,3]

[1]European Spallation Source ERIC, Lund, Sweden
[2]Sigma Connectivity, Lund, Sweden
[3]Lund University, Lund, Sweden

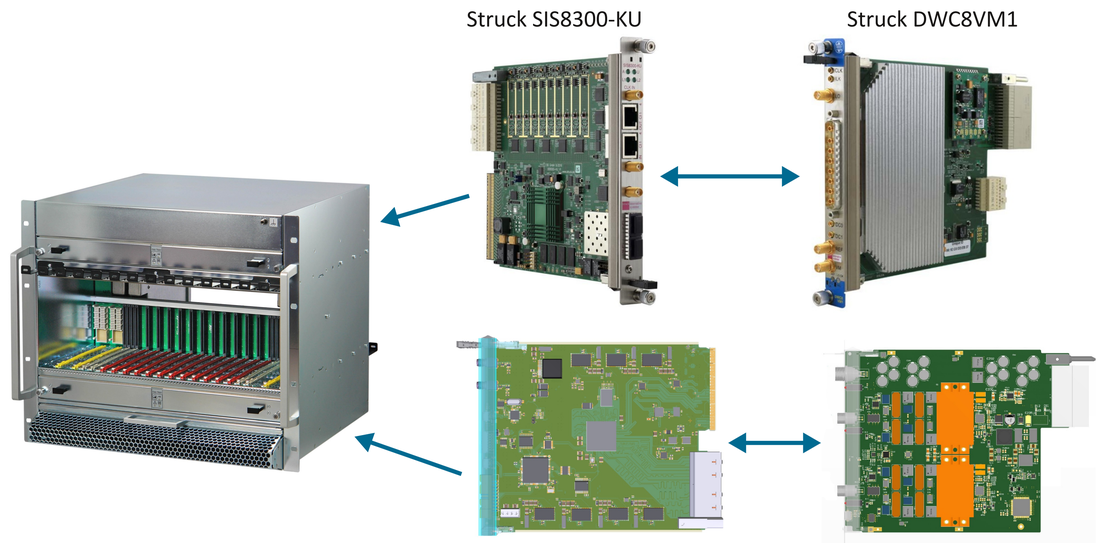
LLRF control system at ESS

At ESS, a MTCA.4-based system is used to control the RF field and tuning of the different accelerator cavities of the ESS linac. An exemplary setup of the system consists of:

- CPU board running CentOS Linux with EPICS control system
- Struck SIS8300-KU: digitizer board
- Struck DWC8VM1: 8 channel downconverter & upconverter
- PEG[1)] Piezo controller board
- PEG RTM carrier to host piezo controller board

More on the LLRF system at ESS is part of the talk by Anders J. Johansson

[1)] PEG: Polish Electronics Group — collaboration between the National Centre for Nuclear Research (NCBJ), Warsaw University of Technology and Technical University of Lodz (TUL)

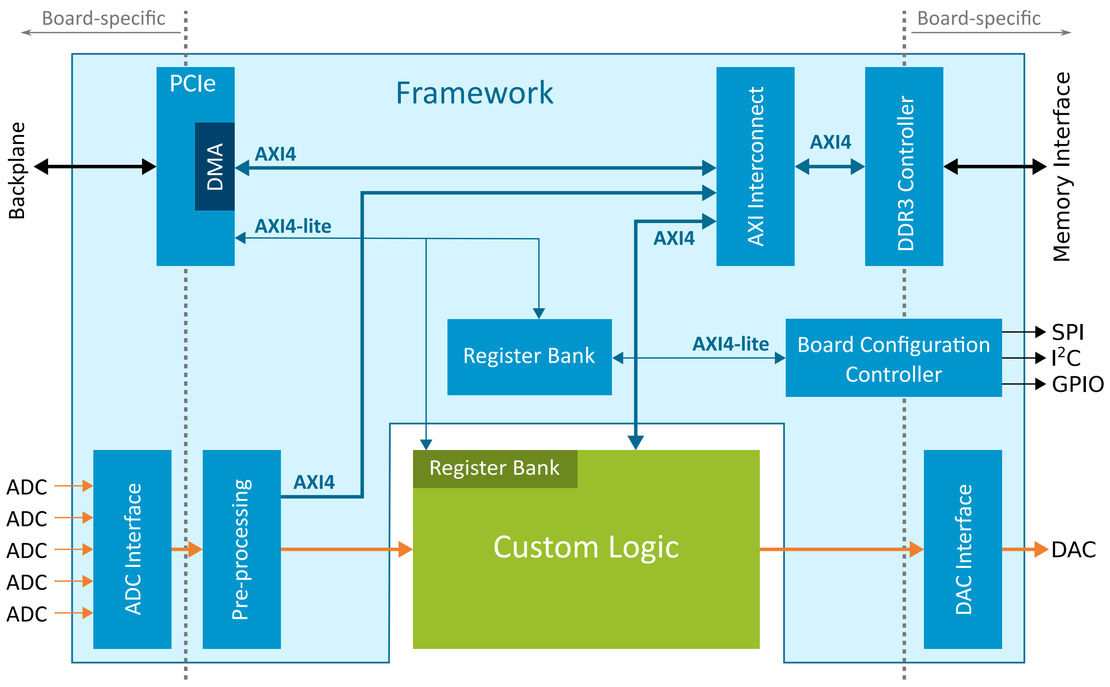


Requirements for the FPGA framework

The main goal of the ESS FPGA Framework is to provide a common platform for the FPGA developments at ESS. Beside the different boards as used by the LLRF system, the framework must also be applicable to other applications, e.g. beam instrumentation.

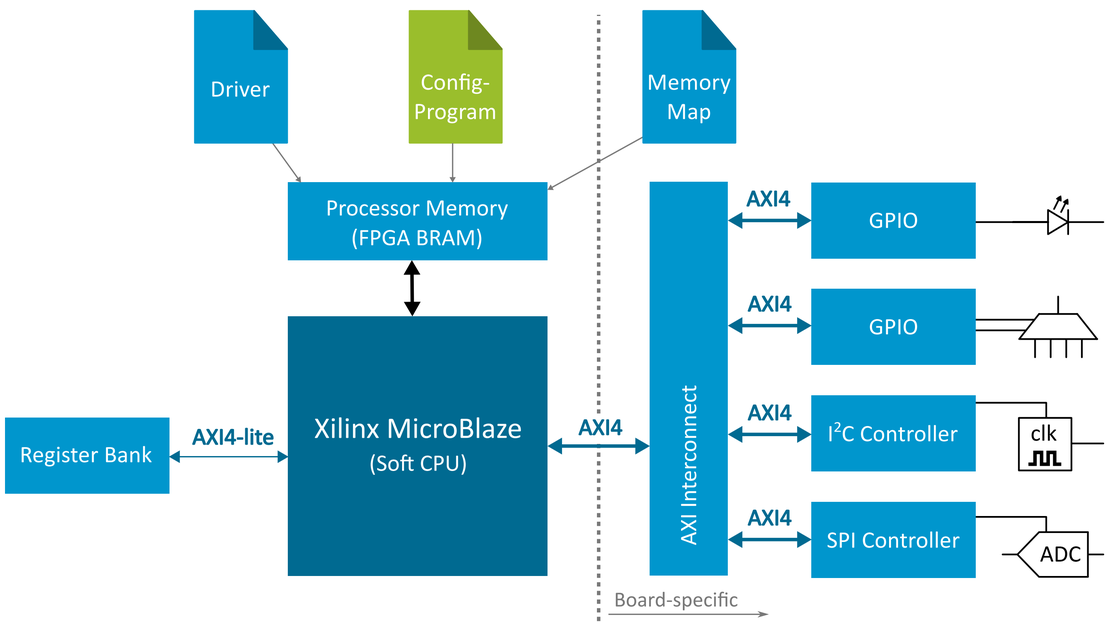
- Configurability: Different boards / different applications
- Long-time support: Reduce dependencies on closed sources
- Simplify driver development and integration with the control system (EPICS)
- Seamless integration to the workflow at ESS

Structure of the ESS FPGA Framework



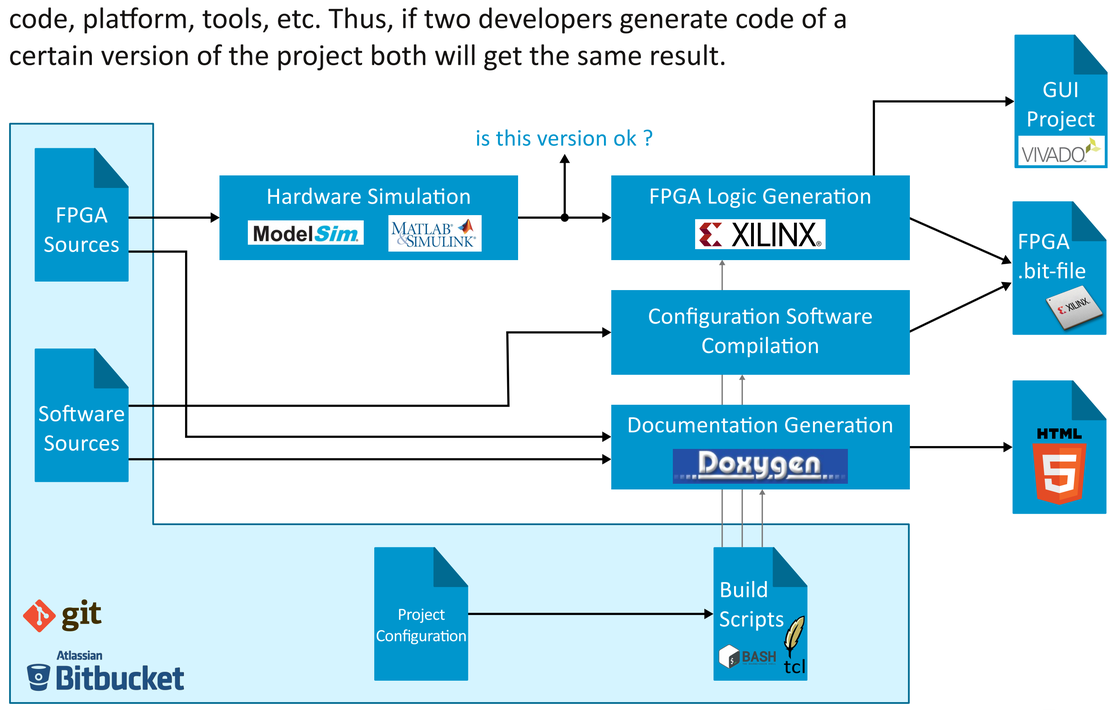
Board configuration by MicroBlaze soft-CPU

Although every board and application needs a specific configuration of the peripherals on the board, the configuration is quite static for most peripherals during the operation. To remove complexity from the driver of the board, the configuration of the peripherals has been implemented on a Xilinx MicroBlaze soft core. This configuration code can then be deployed together with the FPGA configuration code.



Workflow for automated firmware generation

Within the development of the framework, also an automated workflow based on common tools for software development has been designed. The goal of such an automation is to remove any ambiguity between different versions of code, platform, tools, etc. Thus, if two developers generate code of a certain version of the project both will get the same result.



Future: Integrate code generation to the Jenkins continuos integration system and fix the toolchain by using Docker containers.

Results

A first version of the ESS FPGA Framework has been implemented, supporting the Struck SIS8300-KU board. The current version of the LLRF control system has been integrated into the framework. The utilization of FPGA resources has been evaluated under the following conditions:

Device:	Xilinx Kintex Ultrascale xcku040-ffa1156-1-c
Clock frequency:	125 MHz (processing parts)
AXI4 bus:	512 bit @ 200 MHz
Tool:	Xilinx Vivado 2017.1

FPGA Framework

Resource	Utilization	Available	Utilization %
LUT	51463	242400	21.2
Flip-Flop	69973	484800	14.4
BRAM	173	600	28.8
DSP	86	1920	4.5
MMCM	1	10	10.0
PLL	3	20	15.0

FPGA Framework & LLRF Application

Resource	Utilization	Available	Utilization %
LUT	90768	242400	37.4
Flip-Flop	113805	484800	23.5
BRAM	262.5	600	43.8
DSP	235	1920	12.2
MMCM	1	10	10.0
PLL	3	20	15.0

For more information and access to the code repository: christian.amstutz@esss.se

Acknowledgement: We would like to thank Struck Inovative Systeme GmbH for their support in the development of the framework.