A proposal to apply an embedded MicroBlaze® Processor in FPGA module of the cRIO target control

Jerzy Tarasiuk

National Centre for Nuclear Research, Warsaw
and Faculty of Physics, University of Warsaw

PANDA CM 21/1 Target Session 2021-03-10

Introduction

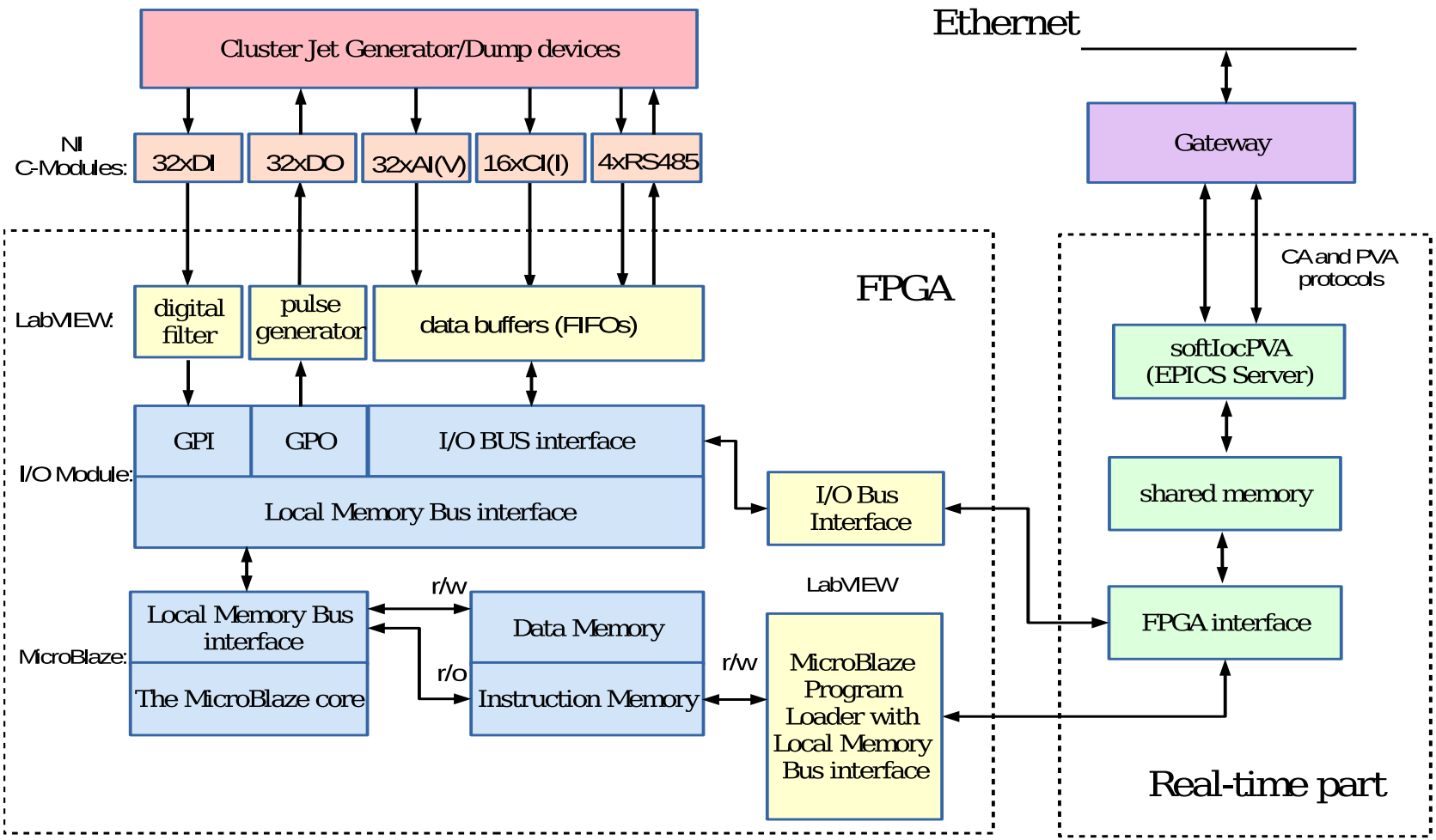
Initially we planned the Cluster-Jet Target control program having all its logic in the Real-Time part of the CompactRIO (cRIO-9039); but it poses a risk of malfunction due to the Operating System failure during the control operation.
To what extent could FPGA help to resolve this problem? I made a test of communication between FPGA program and a C program running on the RT part and found that the first is not affected by the RT system reboot, so:

- The control logic located in FPGA can be functioning during the reboot, and take the necessary actions quickly; if **all the logic and device I/O** is in the FPGA, only the supervisory system communication is unoperable during the reboot.
- The FPGA can reboot the Operating System if it detects the latter is unoperable.

MicroBlaze in FPGA

Having a complex algorithm to implement, I would like to have a processor implemented in the FPGA - this can simplify the control system design and help to organize it. I have chosen MicroBlaze, which has a detailed guide on integration with LabVIEW, and I got it working.
The processor is 32-bit, has memory-mapped I/O, and among other peripherals has GPIs/GPOs (General Purpose I/O-s up to 4x32 each); it has Harvard architecture (separate data/instruction access paths).
The controlled devices are connected via C-Modules; together with communication with Real-Time (RT) part of the cRIO they can be accessed from the LabVIEW program only - the processor needs to be "wrapped" in the LabVIEW FPGA program.
Also, the LabVIEW FPGA program mediates in loading the control program to the processor's memory.

Scheme



A suggested scheme of incorporating the MicroBlaze processor into NI-9039 cRIO FPGA.

Roles of the scheme elements

- The MicroBlaze and its additions (memory, I/O Module) are shown as blue boxes; the processor's access to its instruction memory is read-only to protect its program.
- Yellow boxes represent interface elements in LabVIEW: these above the MicroBlaze connect it with C-Modules, those at the right hand side provide communication between the MicroBlaze and an "FPGA interface" program running in the RT part.
- Green boxes are software elements in the RT part.

The MicroBlaze is to have a "scanner" architecture: its main loop will examine signals (external - from the LabVIEW blocks, and its internal) and have an action assigned for each of them.

Thank you for your attention