Application of an embedded MicroBlaze® processor in FPGA-based cluster-jet slow-control system

Jerzy Tarasiuk

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

PANDA CM 21/1 DCS Session 2021-03-09

Introduction

Let me remind the Audience that the Cluster-Jet Target intends to use NI CompactRIO 9039 (cRIO-9039) as its controller, so I recall here some of its characteristics:

- Considering its programming, it consists of a Real-Time (RT) part and an FPGA part.
- The RT part has: 2GB RAM, 16GB flash (as a disk), with an installed RT Linux operating system; it also has 2 Ethernet ports, 2 USB ports, RS485 and RS232.
- The FPGA (type **Xilinx Kintex-7** XC7K325T) has a 2MB Block RAM, connections to C-Module slots and a 128MB DDR3 RAM.

Why the control logic in the FPGA?

Our first control program proposal presented at CM 20/3, which had all its logic in the RT part, was criticized. A risk of malfunction due to the OS failure and/or reboot was mentioned. 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:

- It can keep information needed by the RT programs for continuing their action - like copy of values put into EPICS records - allowing the RT programs to return to their pre-reboot actions.
- 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 OS if it detects the latter is unoperable.

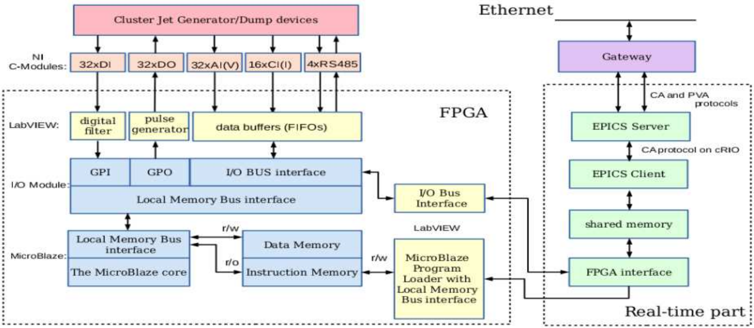
Why MicroBlaze?

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. Which processor to use?

- The NI LabVIEW forum recommended MicroBlaze and Cortex-M1 (or -M3). Both are 32-bit.
- The MicroBlaze has a detailed guide on integration with LabVIEW, and following this guide I got it working.

Basic MicroBlaze properties: 32 registers (some reserved for special purposes), 32-bit instructions; highly configurable; Local Memory Bus (LMB) connecting it with memory (separately for data and instructions), and an I/O Module (memory-mapped I/O). Its C compiler is supplied as Vivado part with LabVIEW FPGA. The I/O Module has GPIs/GPOs (General Purpose I/O-s up to 4x32 each), I/O Bus, timers, external interupt inputs.

Scheme



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

Roles of the scheme elements

- digital filter: eliminates short transient state changes from the signal passed to the GPI;
- pulse generator: forms pulses of a defined duration after getting a trigger signal (edge) from the GPO;
- buffers eliminate the need of tight timing synchronization between C-Module accesses (which take some time) and the MicroBlaze program;
- the MicroBlaze has to have a program designed for processing the data it gets from Cluster Jet Target components, taking decisions about hardware actions, and sending data and status information to the RT part's "FPGA interface";
- programs in the RT part transfer data and status to the EPICS Server records; the server (also in the RT part) makes them available to the Supervisory System.

MicroBlaze enhances flexibility

Using the MicroBlaze inside LabVIEW provides extra capabilities:

- the same FPGA bitimage (compiled LabVIEW program) can be used with different MicroBlaze programs ⇒ provides an ability to test the bitimage using a dedicated program; and to re-use it later, with other external devices (but the same C-Modules, or a their subset);
- the same MicroBlaze program can be used with different FPGA bitimages ⇒ provides an ability to create a test environment for the MicroBlaze program, without need for real devices;
- when a modification of the control is needed (e.g. when some device is replaced, and the new needs a different control sequence), it can be done by uploading a new program to the MicroBlaze, instead a necessity to use LabVIEW, and more, a matching LabVIEW version.

Thank you for your attention