Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Frascati

Belle II

Data Concentrator remote firmware update

matteo.beretta@lnf.infn.it

24 May 2018

Table of Contents

Data Concentrator firmware update

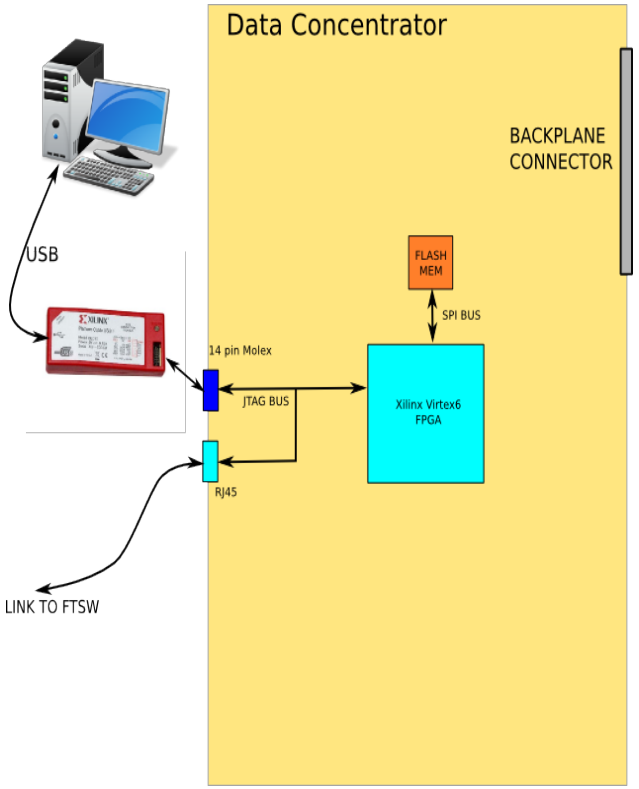
Actual situation

Actually the update of the firmware in the Data Concentrator is done one by one directly on the crate using Xilinx USB programmer.
This is a very time consuming operation (SPI Flash memory indirect programming).
Moreover this procedure can be done only when the machine is off.

JTAG Architecture

Data concentrator has two JTAG interfaces:

- the first (14 Pins MOLEX connector) is used for Xilinx USB programmer connection;
- the second (RJ45 connector) is connected to FTSW board



Remote Indirect Programming

Firmware

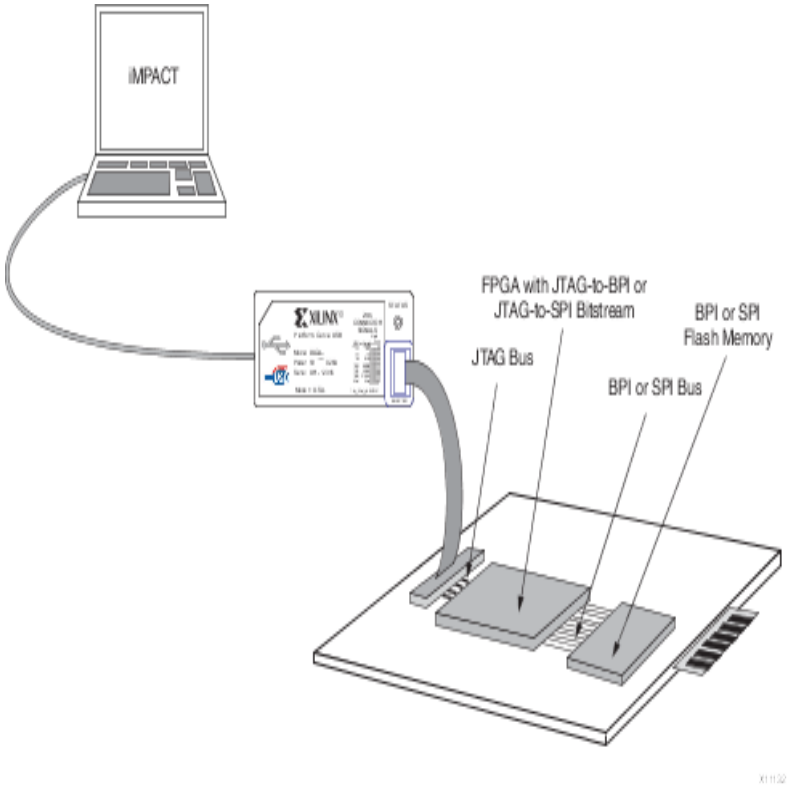
The firmware for data concentrator contain even the firmware for KLM RPC boards in the same crate. At the crate power on Data Concentrator first of all upload his firmware from SPI flash memory than download firmware in all the KLM RPC boards.

FTSW JTAG programming

We would like to give the possibility to update the Data concentrator firmware throught the FTSW board without the needed to access Data Concentrator directly.

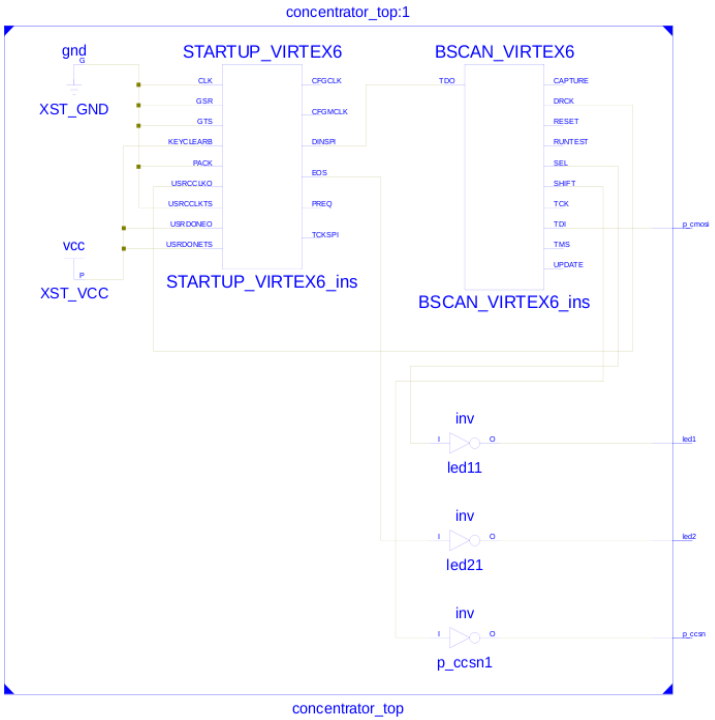
Xilinx Indirect Programming

We have to add to the Firmware of the Data Concentrator a piece of code that implement a bridge between the JTAG interface and SPI interface.
Xilinx FPGAs have 4 JTAG User registers. We use one of these registers to access the bridge.



JTAG to SPI bridge

We have used two primitive of the Virtex6 FPGA to realize the bridge. The simplicity of the bridge is due to the fact that JTAG communication protocol is very similar to SPI protocol. We have only to take care of the polarity and timing of some signals.



Area and debug

Area

The area occupied by the firmware must be reduced to a minimum so as not to reduce resources for the data concentrator's firmware.

Debug

It was not possible to debug the firmware using standard porcedure with chipscope because the JTAG interface was occupied by the remote programming. To develop and debug we have used OpenOCD Free Software with UM232H USB to JTAG module interface.

Conclusion

Firmware

The firmware:

- has been written and implemented in the FPGA;
- debug was done writing and reading several time the flash memory;
- upload was made to a repository provided by Brandon.

We are waiting for the software in FTSW.

Writing Flash Memory

Erase

The first operation to be done when you have to wrte data to the flash memory is to reset the memory. In this way all the memory locations are put to OxFFFFFFFF. The bulk memory reset is a time consuming operation, last from 120 to 240 seconds, cannot be avoided.

The endianness of data to be sent to memory is inverted with respect to that of the FPGA. This problem has to be taken into account when the software will be written.

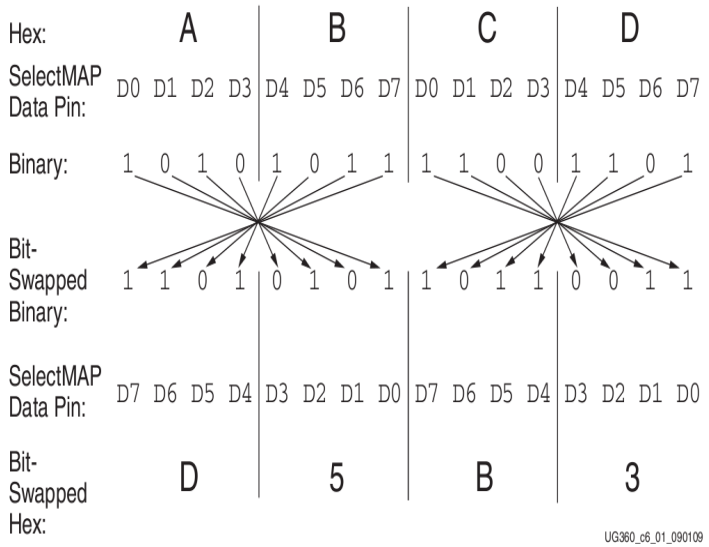


Figure 6-1: **Bit Swapping Example**

Writing Flash Memory

Erase

The first operation to be done when you have to wrte data to the flash memory is to reset the memory. In this way all the memory locations are put to OxFFFFFFFF. The bulk memory reset is a time consuming operation, last from 120 to 240 seconds, cannot be avoided.

Supposing JTAG frequency 5Mhz the programming time is

$$\cong 33Mb/5MHz = 6.6sec$$

Adding erase time

$$6.6sec + 240sec = 246.6sec$$

FPGA	Configuration Bits
Data Concentrator (XC6VLX75T)	26,239,328
RPC Front-End (6SLX25)	6,440,432