European Synchrotron Radiation Facility

# DSP DESIGN USING SYSTEM GENERATOR

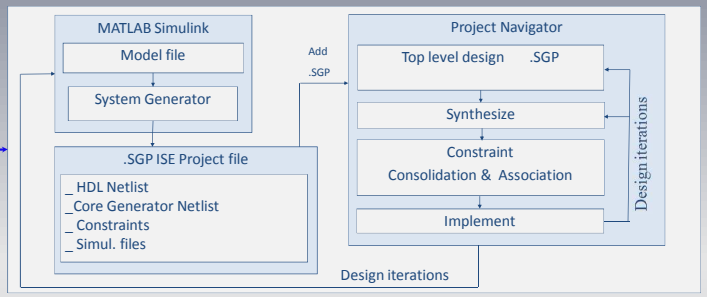
*Jean Marc Koch - ESRF Grenoble, France*

When designing a real time control system, a fast data transfer between the different pieces of hardware must be guaranteed since synchronization and determinism have to be respected. One efficient solution regarding these constraints is to embed the data collection, the signal-processing and the driving of the acting devices in an FPGA even if this solution does not come without difficulties. To overcome one of these difficulties, it is possible to open the development of the signal processing to non HDL (Hardware Description Language) specialists; here, System Generator [1] has been chosen for development purposes, in Simulink / Matlab from Xilinx and The MathWorks. Another challenge with such a system design is the ability to integrate real time models on already pre-configured hardware platforms. Although the hardware can be ready for communication, standard PCI or PCI express bus and dedicated fast data links like Gigabit Ethernet, the corresponding interfaces must be carefully defined and designed to communicate with HDL System Generator control systems blocks. Therefore, the work in this paper describes with two examples how to take advantage of hardware delivered ready for use from a communication point of view and how to get it ready to integrate a design under System Generator.

## FPGA CODE DEVELOPMENT

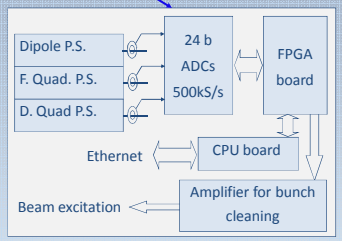
There are several possibilities for the development of a code using System Generator; in the following two examples it is used in the same way: System Generator design is part of a bigger project and as such needs to be connected to this one. All the tasks related to data communication external to the FPGA are coded in vhdl.

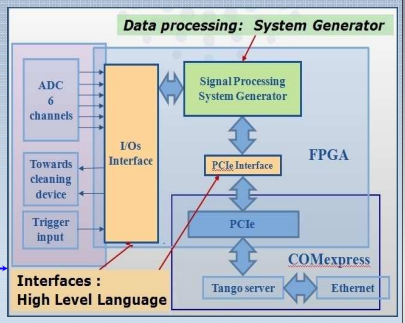
The Signal Processing is described with System Generator, simulated with the Simulink tools in computer simulation and, when the corresponding HDL code is generated, one has access to "hardware in the loop" tests. During this process a wrapper is produced describing the gateways which have to be connected with the "hand written" part of the code.




## 1) BOOSTER TUNES ESTIMATION FROM THE CURRENTS IN THE MAGNETS

The booster is a circular machine designed to accelerate the electron beam from 200Mev to 6 GeV energy. During the ramping, since the currents in the 3 main magnet families involved are not perfectly proportional all along the accelerating cycle and can slightly vary in an unpredictable way from one cycle to the next, the tunes are varying accordingly. As we need to know the value of the tunes to excite the beam for a cleaning process [3], a measurement and calculation in real time will be performed to drive an amplifier with the correct frequency to perform a blow-up of the unwanted bunches of the beam.



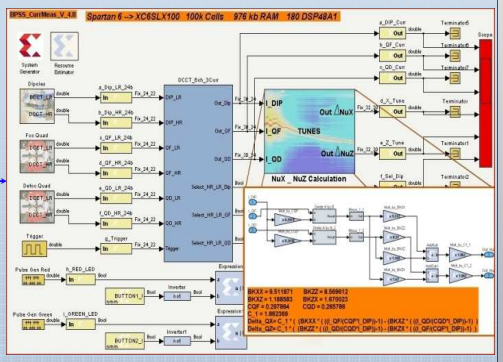

**REQUIRED INTERFACES IN THE FPGA →**

1) Driver for ADCs readout from serial to 24bits word
2) Misc I/Os for the trigger and cleaning device
3) PCIe interface with ComExpress board

**SIMULATION WITH SIMULINK →**

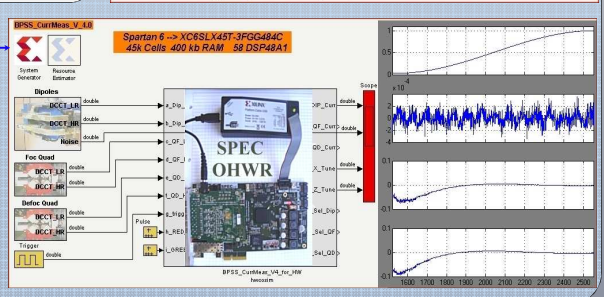
To achieve a very accurate measurement, the DCCTs have two ranges; an extra output has a full scale of only 10% of the full range. The selection between the two outputs is performed in the first System Generator block.

The second block shown above is dedicated to the calculation with fix point data of the variations of the tunes around their nominal values. Once the code has been simulated, it can be tested on a preconfigured hardware platform thanks to the "Hardware in the Loop" possibility of System Generator.




**HARDWARE IN THE LOOP →**

Any FPGA board equipped with a Xilinx FPGA and JTAG connector can be described thanks to the Board description Builder [1] available in the System Generator Compilation box .

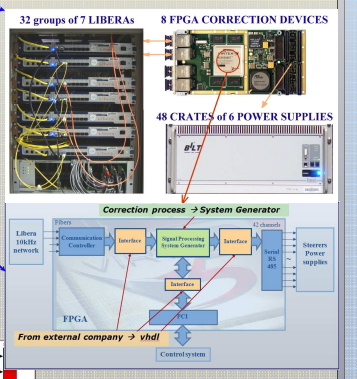


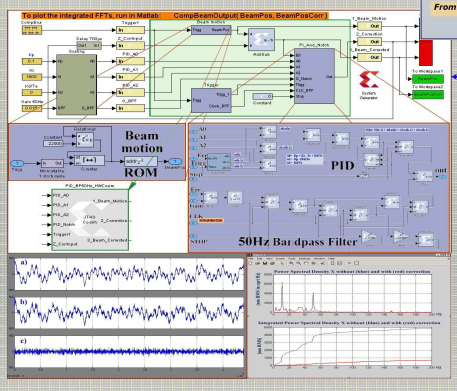

## 2) FPGA BASED FAST ORBIT CORRECTION SYSTEM

The architecture chosen is such that from the electrons beam to the power converters, the whole data processing is performed inside FPGAs [2]: the front-ends are digital BPMs "Liberas" and the data are dispatched to 8 stations equipped with Virtex-5 FPGAs through optic fibers thanks to a custom Communication Controller designed and implemented in VHDL at Diamond Light Source [4]

**FUNCTIONS OF THE CODE EMBEDDED IN THE FPGA**

1) Collect the data from the BPMs at 10 kHz with the Communication Controller and transfer to System Generator through shared memories and registers (coded in High Level Language) These interfaces have to conform to the symbolic names, the words length and memories depth defined in the vhdl development.
2) Obtain the parameters from the PCI (coded H.L.L.)
3) Process the corrections (coded with System Generator)
4) Send the set-points to the power supplies (coded H.L.L.)
5) Send the set-points for data recording to the Communication Controller (coded H.L.L.)
6) Synchronous start and stop of the correction on the 8 stations in the same 10 kHz cycle.



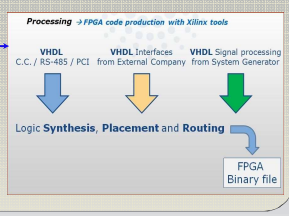




**← SIMULATION WITH SIMULINK**

To check the behavior of the model, real data are introduced at the initialization of the simulation and processed with the components from Xilinx and then, data obtained from the output can be visualized or saved for analysis under Matlab. In this simulation, the power supplies, magnets and vacuum chambers are modeled as a simple delay of 700µs between the correction output and the reading of the beam position.

**FPGA CODE PRODUCTION →**

All the functions except the signal processing were either developed or integrated by an external company specialized in vhdl coding. Providing that the interfaces cover the needs for all the data transfer, the data processing can evolve without needs of modifications for the other codes. The synthesis, placement and routing can be launched with the new files from System Generator all the others being frozen.




## CONCLUSION

Progress in FPGA technology along with availability of high level tools for modeling, simulation and synthesis have made FPGA a key platform. Today, it is a good choice for the hardware development and implementation of high-performance applications requiring rigorous calculations in real-time. Many developments at ESRF have proven that FPGAs can outperform DSPs and embedded processors in signal processing, while minimizing the hardware burden of multiple data paths. The right tools and techniques coupled with innovative features in silicon architecture can produce complex DSP functions in a single FPGA. Xilinx System Generator is a system level modeling tool that facilitates FPGA hardware design by extending Simulink / Matlab in numerous ways to provide a powerful modeling environment. It opened the field of FPGA design to non specialists through its design environment. In addition, the possibilities for testing and verification both at simulation level and combined simulation / hardware in the loop are an essential point to successfully implement a design.

*References*

[1] Xilinx System Generator for DSP Reference Guide
[2] E. Plouviez et al "The new fast orbit correction system of the ESRF Storage Ring", Proc DIPAC 11, May 2011
[3] Cleaning of parasitic bunches in the ESRF booster synchrotron E. Plouviez et al. EPAC 2004
[4] Diamond Light Source Fast Orbit Feedback Communication Controller Specification and Design Isa Uzun et al. January 2009

jmkoch@esrf.fr