FPGA based RF control

A.Burghardt, ASKON Beratungs GmbH, Hamburg, Germany
S.N.Simrock, DESY, Hamburg, Germany

Abstract

The RF control system of the TESLA Test Facility employs a digital control, feedback and feedforward system to provide flexibility of control algorithms, extensive diagnostics and high immunity against electromagnetic interference. The current implementation features digital signal processors (DSP) for computing and processing the necessary control functions. Despite the fact, that the selected 320TMS6701 DSPs from TI contain four parallel execution units, significant amount of serial processing is necessary. In contrast, Field programmable Gate Arrays (FPGA) are reprogrammable logic devices, which have the ability of much higher degree of parallel processing. Due to the recent advances in terms of available hardware and software design flow, FPGAs are well suited for high-speed and low latency RF control purposes.

1. Introduction

Recent developments and enhancements of programmable logic devices towards the needs of digital signal processing made it possible, that a lot of RF control functions can be implemented in a FPGA with the additional benefit of parallel processing and reconfiguration. Extended flexibility, higher processing speeds and lower latencies as well as cost reductions can be achieved with this new approach. Nearly all expensive components of our 2nd generation RF control board (DSP, 3 FPGAs for glue logic, Gigalink transceivers) can be integrated into one FPGA in our 3rd generation RF control board.
To evaluate the design process and the limitations of the new approach we have developed main RF control functions (e.g. Matrix Rotation, Vector Sum, Filters, Feedforward control) [1] and implemented them into the FPGA on the Xilinx DSP Xtreme board. The results were published in [2].

2. The Xilinx DSP Xtreme development Kit

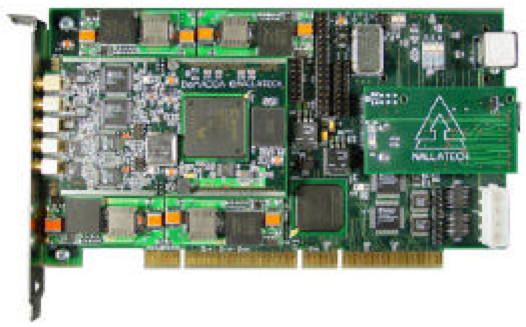


Figure 1 Xilinx DSP Xtreme board

The Xilinx DSP Xtreme development kit contains:

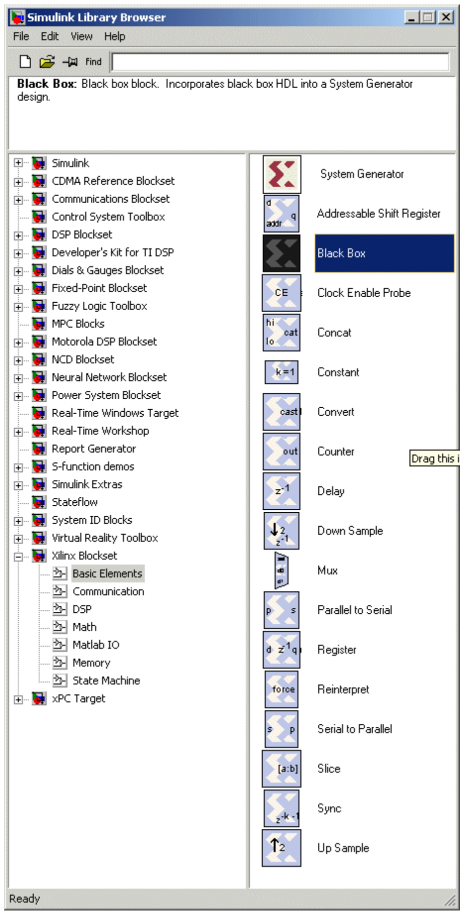
- PCI motherboard with a USB interface for standalone operation
- Daughterboard with
  - 2 AD6644 ADCs (14 bits, 65 MSPS)
  - 2 AD9772A DACs (14 bits, 150 MSPS)
  - 1 Virtex2 XC2V3000 FPGA
  - 1 Virtex2 XC2V80 FPGA for clock management
  - 4MB ZBTRAM
- Cables and power supply
- 30-day evaluation version Matlab 12.1
- 30-day evaluation version Xilinx ISE 4.2
- 90-day evaluation version Xilinx DSP System Generator
- USB driver on CD
- FUSE software for FPGA configuration
- C++ API for FPGA configuration, data transfer, diagnosis
- Pinning and constrain files
- Documentation on CD

The FPGAs are configured through the USB interface with the help of the FUSE software or with an own software based on the supplied C++ API. It is also possible to exchange data between a PC and the board with this API.

There is no special design-flow necessary for programming the FPGA. Traditional hardware description languages, like VHDL or Verilog can be used as well as the Xilinx DSP System Generator (Matlab/Simulink toolbox). Even a mixture of these design-flows can be used.

System Generator for DSP allows a system architect to quickly model and implement a custom data path processor using handcrafted IP. It provides the FPGA and DSP designer system-level access to key features in silicon, including embedded multipliers, shift register logic and block memories.

System Generator for DSP is seamlessly integrated with the Matlab/Simulink environment, bridging the gap between high-level system design and actual implementation in a Xilinx FPGA. By combining high-level abstractions with automatic generation of an FPGA circuit, System Generator for DSP decreases the development time, while providing greater opportunities for algorithmic refinement.



Black Box: Black box block. Incorporates black box HDL into a System Generator design.

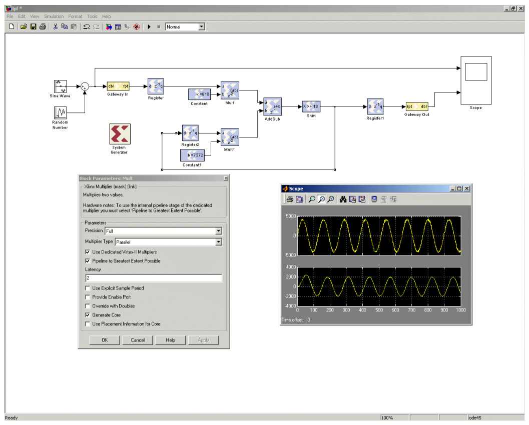
Simulink
CDMA Reference Blockset
Communications Blockset
Control System Toolbox
DSP Blockset
Developer's Kit for TI DSP
Dials & Gauges Blockset
Fixed-Point Blockset
Fuzzy Logic Toolbox
MPC Blocks
Motorola DSP Blockset
NCD Blockset
Neural Network Blockset
Power System Blockset
Real-Time Windows Target
Real-Time Workshop
Report Generator
S-function demos
Simulink Extras
Stateflow
System ID Blocks
Virtual Reality Toolbox
Xilinx Blockset
- Basic Elements
- Communication
- DSP
- Math
- Matlab IO
- Memory
- State Machine

xPC Target

System Generator
Addressable Shift Register
Black Box
Clock Enable Probe
Concat
Constant
Convert
Counter
Delay
Down Sample
Mux
Parallel to Serial
Register
Reinterpret
Serial to Parallel
Slice
Sync
Up Sample

Figure 2 Xilinx blockset for Matlab/Simulink

Several basic blocks as well as special DSP blocks (e.g. FFT, FIR filters) are predefined. Other blocks like multiply-add structures (MAC) can be easily assembled.



Block Parameters: Mult

Xilinx Multiplier (mask) (link)

Multiplies two values.

Hardware notes: To use the internal pipeline stage of the dedicated multiplier you must select 'Pipeline to Greatest Extent Possible'.

Parameters

Precision: Full

Multiplier Type: Parallel

☑ Use Dedicated Virtex-II Multipliers
☑ Pipeline to Greatest Extent Possible

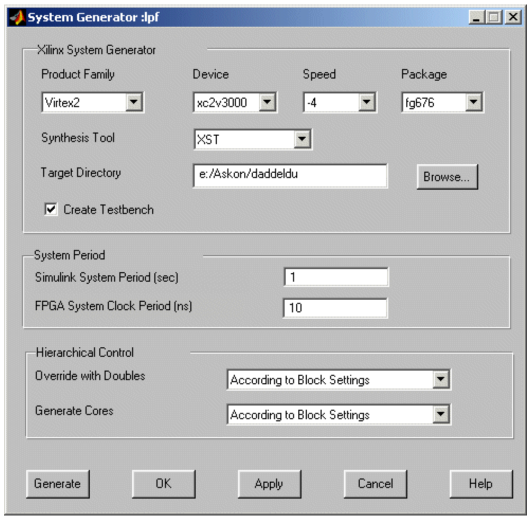
Latency: 2

☐ Use Explicit Sample Period
☐ Provide Enable Port
☐ Override with Doubles
☑ Generate Core
☐ Use Placement Information for Core

OK Cancel Help Apply

Figure 3 Simple low-pass filter built with Xilinx blockset

All blocks can be simulated in Simulink. They are using a fixed number of bits in hardware. The amount of used bits can be freely changed, therefore it is possible to define e.g. a 47-bit adder, if a calculation needs this precision. The overflow behavior can be switched between Wrap around, Saturation and Error flag. For simulation purposes it is possible to switch to double floating point calculation block by block or as a global setting. There is a information box available, which, if it is attached to a line, displays the overall rounding error. With these powerful tools it is quite easy to adjust and calculate the needed fixed point resolution for a given calculation precision.



Xilinx System Generator

Product Family	Device	Speed	Package
Virtex2	xc2v3000	-4	fg676

Synthesis Tool: XST

Target Directory: e:/Askon/daddeldu

☑ Create Testbench

System Period

Simulink System Period (sec): 1

FPGA System Clock Period (ns): 10

Hierarchical Control

Override with Doubles: According to Block Settings

Generate Cores: According to Block Settings

Figure 4 System Generator dialog

The system generator block generates all necessary files:

- VHDL file of the circuit, optimized for the target FPGA family
- VHDL testbench
- Synthesis scripts for different synthesizers (XST, Leonardo Spectrum, Synplify)
- Scripts for Xilinx Place & Route tool

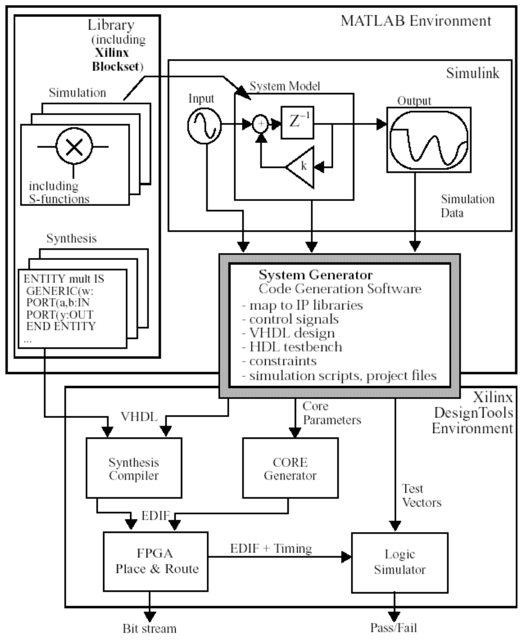


Figure 5: Design-flow System Generator

3. Programming considerations for FPGA boards

Even if the System Generator is a valuable help, there are some implementation areas, which needs attention:

- Special FPGA features
- Integration
- Timing & latency
- Reconfiguration

Special FPGA features: Only a subset of the available resources inside a FPGA are implemented in the Xilinx Blockset. Special hardware related features, e.g. Clock PLLs, differential I/O buffers or available IP-cores can only be instantiated in VHDL. These components increase the performance of the overall circuit tremendously. The best solution for this problem is to build a VHDL shell containing all necessary components around the DSP core.

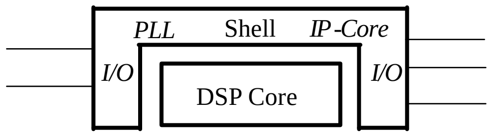


Figure 6 VHDL Shell around DSP Core

The advantage of this solution is, that the DSP designer don't have to know about specific FPGA problems, like clock distribution and can concentrate on the functionality of the DSP blocks. The names of the interface signals between DSP block and shell must be of course standardized.

Integration: Not only the FPGA resources must be taken into account, but also the resources or controlling elements on the board or the outer world. Which clocks are available (frequency, logic level), which reset signals, configuration signals for the ADCs/DACs and so on. Most of these tasks can be handled inside the VHDL shell, assuring that the DSP core gets the correct signals.

Timing & Latency: The DSP core is always a synchronous design, therefore a clock is always needed. The maximum allowable frequency is defined as 1/longest logic path in nanoseconds between two registers:

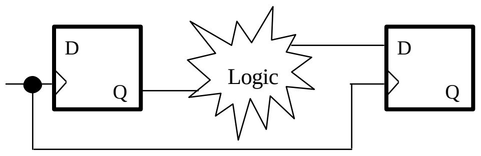


Figure 7 Register to Register delay

Xilinx delivers a static timing analyzer to calculate automatically the longest path and therefore the maximum allowable clock frequency. Logic delays are usually very long in arithmetic calculations like big adders or multiplications. There are three possible solutions, if the calculated clock frequency is less than the desired frequency:

1. Decrease the length of the logic path through adding more registers. It's possible to define the latency (amount of registers) in nearly every Xilinx block. Examination of the Xilinx timing report identifies the exact position of the block(s) with the longest path delay.
2. Rearrange the function blocks if possible and redistribute the used registers (manual register balancing)
3. Last option – decrease of the clock frequency

It is always a tradeoff between clock frequency and latency to achieve the highest possible computation speed.

Reconfiguration: FPGAs are usually based on SRAM structures. At power-up they have to load their configuration data from a memory or an external processor. The configuration process can be started even during operation, thus giving the opportunity to change the functionality on the fly. The configuration time depends on the size of the FPGA and the configuration source, ranging from some milliseconds to some 100 milliseconds. This can be a viable option to do some special processing between beam pulses.

4. Comparison Xilinx – Altera

Xilinx [3] and Altera [4] are the biggest competitors in the field of programmable logic devices (PLD). They have a long and successful history since 1984. Their combined market share is approx. 90% of the PLD market.

Xilinx:

Device family:	**Virtex2**
Availability:	Now
Manufacturing process:	0.15µ, 8 metal layers
Registers:	512 – 93184
Logic cells:	512 – 93184
Memory:	72 – 3024 kBits
PLLs:	4 – 12
Multipliers (18x18):	4 – 168
Typical system freq.:	100 - 300 MHz
Usable user I/O pins:	88 - 1108
Price:	$28 - $11600

Special features: Impedance controlled I/Os, LVDS output up to 840 Mb/s

Future device family:	**Virtex2pro**
Availability:	Q4/2002
Manufacturing process:	0.13µ, 9 metal layers
Registers:	3168 – 125136
Logic cells:	3168 – 125136
Memory:	216 – 10008 kBits
PLLs:	4 – 12
Multipliers (18x18):	12 - 556
Typical system freq.:	150 - 350 MHz
Usable user I/O pins:	204 - 1200
Price:	$132 - $?

Special features: up to 4 PowerPC Cores PPC 405 (300 MHz, MMU) included, 4-24 Rocket IO blocks with 3.125 Gb/s per pair and 8b/10b en-/decoder, Impedance controlled I/Os

Altera:

Device family:	**Stratix**
Availability:	Q4/2002
Manufacturing process:	0.13µ
Registers:	10570 – 114140
Logic cells:	10570 – 114140
Memory:	920 – 10000 kBits
PLLs:	6 – 12
Multipliers (18x18):	24 – 112
Typical system freq.:	100 - 300 MHz
Usable user I/O pins:	335 - 1314
Price:	$310 - $10000

Special features: Impedance controlled I/Os, Serial link outputs up to 840 Mb/s, special DSP blocks with MAC structure

Device family:	**Stratix GX**
Availability:	Q1-2/2003
Manufacturing process:	0.13µ
Registers:	10570 – 41250
Logic cells:	10570 – 41250
Memory:	920 – 3420 kBits
PLLs:	4 – 8
Multipliers (18x18):	24 – 56
Typical system freq.:	100 - 300 MHz
Usable user I/O pins:	335 - 544
Price:	$? - $?

Special features: 4-20 serial links max. 3.125Gb/s with 8b/10b en-/decoder, Impedance controlled I/Os, special DSP blocks with MAC structure

Both companies have suitable devices for high performance DSP applications as well as a Matlab toolbox. An advantage for Xilinx is the availability of the Virtex2 evaluation board and the soon available Virtex2pro family with integrated PowerPC CPU Cores. These CPU cores can be programmed with the standard GNU tools. Altera is offering a Softprocessor (NIOS), which can be exactly tailored to the required functionality, but it occupies some of the logic resources inside the FPGA. The serial high-speed interface of the Virtex2pro and Stratix GX is another big advantage for transmitting/receiving data or to connect some FPGA boards to a big computing cluster.

5. Design considerations for next generation RF control

5.1 Hardware issues and solutions

The most limiting factor of the Xilinx demo board for RF control tasks is the presence of only two ADCs. As stated in [2] it is possible to solve this problem using analog 8:1 multiplexer. The FPGA will generate the necessary select and control signals. The same principle can be used, if more DACs are needed.

Another problem is the enormous range of the values up to a 10^{20} difference in a matrix. The fixed point representation of these numbers must use at least 67 Bits ($\log(10^{20})/\log(2)$), but it is more realistic to use at least 80 Bits for accuracy. Such a large number of bits slows down the resulting operation speed of adders or multipliers. There are four possible solutions for this problem:

- **Pipelining:** Add more registers to increase the clock frequency, but this will result in a latency penalty.
- **FPU core:** Add a floating point core to the FPGA. This could be an interesting alternative, depending on the accuracy (single, double) and the license fee. There are some license free cores available on the internet [5]. One company is currently developing a floating point co-processor for the PowerPC of the Virtex2Pro FPGA [6], but it is not available yet.
- **FPU processor:** Using an external floating point processor is not a good solution, because there are no longer dedicated FPU chips, like the i8087 or MC68882, available. Furthermore they require additional board space, FPGA pins and maybe voltage adapters for the I/O pins.
- **External DSP with FPU:** Very expensive, but can be used as auxiliary processor for special tasks.

The easiest way is to test the pipelining approach and check the overall computation speed. This should be done first. A FPU core could be a good solution combined with the PowerPC core, but a license fee has to be paid, which is in a range of several \$1000-\$10000. Due to the lack of available stand-alone FPU chips, the third option can be neglected. The last option with an external floating point DSP is expensive, but maybe useful for special tasks, e.g. doing some statistics, while the FPGA handles the real-time data. An integrated processor, like the PowerPC in a Virtex2pro, can do the same task, but is more cost effective than a separate DSP.

5.2 Software issues and considerations

The Xilinx System Generator as well as the Altera DSP builder are only available for PC/Windows environments, therefore Unix workstations can't be used for the new design flow.

The programming of the VHDL shell (see chapter 3) can be only performed by an experienced VHDL programmer who understands the hardware. A student or a beginner does not have the necessary experience in handling the complex details of clock distribution or different I/O configuration. The expert VHDL designer can also provide customized VHDL blocks, which can't be created with the System Generator tool.

Another topic is the partitioning of the overall design. In an early stage of the project it must be decided, which design features should be implemented in the FPGA, which in the PowerPC core, and which in the embedded CPU in the VME crate or in the CPU in a mainframe computer. This discussion will have a big impact on the overall design of the new TESLA DSP board. Therefore a decision is necessary, before the design of the DSP board can be started.

6. Conclusion

The proposed new design flow offers a powerful and smooth transition towards the use of FPGAs in RF control applications. The parallel data processing helps to speed up the control algorithms. Matlab toolboxes are a big help for DSP designers. They can use their standard tools to develop applications for FPGAs, without knowledge of VHDL design and the underlying hardware. Even more possibilities exist when using the integrated CPU cores in which certain functional blocks can be realized more efficiently. In this case the flexibility of adding newly developed algorithms in a later stage of a project is greatly improved. The total cost of this solution is equal or even less than the actual implementation with a dedicated DSP.

7. References

[1] T. Schilcher, "*Vector Sum Control of Pulsed Accelerated Fields in Lorentz Force Detuned Superconducting Cavities*", Hamburg, 1998

[2] W. Zabolotny, K. Pozniak, R. Romaniuk, T. Czarski, I. Kudla, K. Kierzkowski, T. Jezynski, A. Burghardt, S. Simrock, "*Design and Simulation of FPGA Implementation of RF Control System for Tesla Test Facility*", not yet published

[3] Xilinx homepage, **http://www.xilinx.com**

[4] Altera homepage, **http://www.altera.com**

[5] Opencores.org, **http://www.opencores.org**

[6] Dillon Engineering, **http://www.dilloneng.com**