The **DSP** *for* **FPGA Primer**

Course Aim:

To present theory, algorithms, design techniques and actual practicalities of the implementation of DSP algorithms and digital communications architectures using FPGA technology.

Course Presentation Style:

This is an intensive 2 day course which will educate using a comprehensive set of notes on DSP for FPGAs. Key points will be lectured upon with derivations and technical details provided in the course notes for later self study. Following each lecture, hands-on lab sessions will be run using Xilinx FPGA hardware and software.

Delivery will be 40% lecture, 20% demonstrations, and 40% hands-on labs using FPGA hardware and software.

Who Should Attend?:

University lecturers interested in using Xilinx devices for teaching, research and development. Also analog, RF, digital, DSP or FPGA/ASIC engineers who are interested in knowing the relevant design strategies and philosophies for implementing algorithms and applications on FPGAs may find the course beneficial. A background in some of the fundamentals of DSP (sampling, quantization, frequency domain, digital filtering) is useful, but not essential.

Course Notes, Hardware and Software:

All attendees will receive printed and electronic copies of the “DSP for FPGAs Primer” notes. These materials are open source and available for attendees to reuse by appropriate reference to original source. University lecturers and professors with direct involvement in teaching DSP and/or FPGA design who attend the course can receive the hardware and software via donation from the Xilinx University Programme (XUP).

Attendees will use ***System Generator*** for Matlab/Simulink software to flow into the ***Xilinx ISE Tools***, v7.1 to design DSP systems for the ***Xilinx XUP Virtex-II Pro*** development kit consisting of the XC2VP30 (with 30816 logic cells).

Learning Objetives

- Understand the current and relevant DSP applications for FPGAs
- When to use an FPGA or a DSP processor - or both!
- Arithmetic issues - How to implement multiplies and adds - efficiently!
- The (sometimes serious!) impact of rounding versus truncation
- Dealing with overflow and underflow scenarios
- Advanced Arithmetic - When we need square roots, divides and more?
- Design techniques for minimizing sample wordlengths
- Efficient FIR (finite impulse response) filter design and implementation
- The use of IIR (infinite impulse response) filters in DSP for FPGA applications
- The importance of retiming, pipelining, and multichannel filters
- The cost and relevance of special filters such as CIC (cascade integrate-comb) filters
- The requirements and implementation of adaptive filtering algorithms
- The implementation of IF modulation and demodulation techniques
- Why and how to implement numerically controlled oscillators (NCOs)
- Techniques for synchronisation & digital comms timing recovery
- System architecture and implementation of direct Digital DownConverter (DDC)
- DSP/FPGA components to implement a QAM (Quadrature Amplitude Modulator) transceiver
- How to efficiently implement multichannel filters for 3G applications
- Design strategies for implementation of orthogonal frequency division multiplexing (OFDM)
- Using the QR algorithms for adaptive equalisation and beamforming
- Implementation of an FPGA enabled physical layer for 802.16

Syllabus

DSP for FPGA Technology & Application Review

- DSP for FPGA applications
- Wordlengths issues - DSP on Xilinx FPGAs is not just 16 bits!
- Design for applications sampling at > 100MHz
- FPGA applications examples: 3G, 802.16, cdma2000
- FPGAs, DSP processors, ASIC - what to use - when and where
- Linear algebra - matrices, vectors
- Calculating the matrix inverse and DSP requirements

FPGA Technology

- The Xilinx DSP for FPGA technology roadmap
- Clocking rates, data rates and sample rates
- Bits, Slices, Configurable Logic Blocks, and Multipliers
- MIPs and MACs performance ratings
- FPGA families and sources
- Case Study - the Virtex 4 and DSP48 slices
- Review of an HDL design flow from algorithm to implementation

Tools for DSP for FPGA Design

- Working with Matlab and Simulink
- Xilinx System Generator
- High level design flow – from algorithm to Simulink to FPGA
- Hardware in the loop

Arithmetic Fundamentals

- 2's complement fixed point arithmetic
- Adders and multipliers, and introducing...division and square root
- Wordlength issues & Fixed point arithmetic
- Overflow/underflow and Truncation/Rounding issues
- Complex arithmetic (real and imaginary) requirements for DSP
- The role of arithmetic approximation algorithms and CORDICs

Digital Filtering for FPGAs

- Symmetric / Linear Phase Filters - Xilinx efficieny & optimisation
- Upsampling/interpolation & Downsampling/decimation
- Trade-offs with wordlength, sampling rate and filter lengths.
- Retiming techniques
- Cut-set delay for transpose and systolic FIR filters
- Half-band, moving average, comb filters and CIC filters
- Multichannel filter implementation
- Polyphase filter implementation

Adaptive Filtering for FPGAs

- The issues from numerical feedback and how to deal with them
- The LMS (least mean square) algorithm
- LMS implementation and application
- The RLS (recursive least squares) algorithm
- RLS implementation - the QR algorithm - Classical linear algebra
- Numerical integrity and stability issues

QAM (Quadrature Amplitude Modulation) Systems

- The DSP emabled IF Radio architecture (software radio)
- Design of numerically controlled oscillators (NCOs)
- Design of transmit and receive matched digital filters
- Carrier timing recovery, and symbol synchronisation techniques
- Constellations, phase rotations, and test scenarios
- Spread spectrum strategies and requirements

FPGA System Level DSP Applications

- A 3G, fs = 80MHz, 4 x 5MHz oversampled multichannel filters
- Bluetooth compatible direct digital downconverter (DDC) design
- Adaptive LMS based equalisation for wireline applications
- Adaptive QR algorithm for wireless digital beamforming
- Design of NCO, FIR filter for Generic QAM transmitter