

V4 - FPGA Optimization

Hardware Architecture

Objectives

- Design and optimize FPGA-based systems for performance, area, and power
- Understand timing, pipelining, and clock domain crossing challenges
- Apply efficient memory and resource utilization strategies
- Develop hardware implementations of DSP and mathematical algorithms
- Master RTL coding techniques for synthesis and hardware reliability
- Perform simulation, verification, and timing analysis
- Understand physical design: floorplanning, place & route, and constraints
- Integrate FPGA designs with embedded software systems
- Explore hardware/software co-design and acceleration techniques
- Use ARM NEON for software-side performance optimization

Prerequisites

- VHDL or Verilog concepts
- C Language knowledge (see for example our L2 training course)
- Familiarity with FPGA concepts

Course environment

- Theoretical course
 - PDF course material (in English)
 - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance
- Practical activities
 - Practical activities represent from 40% to 50% of course duration
 - Example code, labs and solutions
 - Vivado or Libero for design, synthesis, and timing analysis; ModelSim or Vivado for simulation

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais) et une version imprimée lors des sessions en présentiel
 - Cours dispensé via le système de visioconférence Teams (si à distance)
 - Le formateur répond aux questions des stagiaires en direct pendant la formation et fournit une assistance technique et pédagogique
- Activités pratiques
 - Les activités pratiques représentent de 40% à 50% de la durée du cours
 - Elles permettent de valider ou compléter les connaissances acquises pendant le cours théorique.
 - Exemples de code, exercices et solutions
 - Pour les formations à distance:
 - ▶ Un PC Linux en ligne par stagiaire pour les activités pratiques, avec tous les logiciels nécessaires préinstallés.
 - ▶ Le formateur a accès aux PC en ligne des stagiaires pour l'assistance technique et pédagogique
 - ▶ Certains travaux pratiques peuvent être réalisés entre les sessions et sont vérifiés par le formateur lors de la session suivante.
 - Pour les formations en présentiel:
 - ▶ Un PC (Linux ou Windows) pour les activités pratiques avec, si approprié, une carte cible embarquée.
 - ▶ Un PC par binôme de stagiaires s'il y a plus de 6 stagiaires.
 - Pour les formations sur site:
 - ▶ Un manuel d'installation est fourni pour permettre de préinstaller les logiciels nécessaires.

- ▶ Le formateur vient avec les cartes cible nécessaires (et les remporte à la fin de la formation).
- Une machine virtuelle préconfigurée téléchargeable pour refaire les activités pratiques après le cours
- Au début de chaque session (demi-journée en présentiel) une période est réservée à une interaction avec les stagiaires pour s'assurer que le cours répond à leurs attentes et l'adapter si nécessaire

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Course Outline

First Day

Architecting Speed

- High Throughput
- Low Latency
- Timing
 - Add Register Layers
 - Parallel Structures
 - Flatten Logic Structures
 - Register Balancing
 - Reorder Paths

Exercise : Example of Optimizing a Multiply-Accumulate Block

Architecting Area

- Rolling Up the Pipeline
- Control-Based Logic Reuse
- Resource Sharing
 - Impact of Reset on Area
 - Resources Without Reset
 - Resources Without Set
 - Resources Without Asynchronous Reset
 - Resetting RAM
 - Utilizing Set/Reset Flip-Flop Pins

Exercise : Example of analyzing, comparing and optimizing multiple designs

Architecting Power

- Clock Control
 - Clock Skew
 - Managing Skew
- Input Control
- Reducing the Voltage Supply
- Dual-Edge Triggered Flip-Flops
- Modifying Terminations

Example Design: The Advanced Encryption Standard

- AES Architectures
 - One Stage for Sub-bytes
 - Zero Stages for Shift Rows
 - Two Pipeline Stages for Mix-Column
 - One Stage for Add Round Key

- Compact Architecture
- Partially Pipelined Architecture
- Fully Pipelined Architecture
- Performance Versus Area
- Other Optimizations

Second Day

High-Level Design

- Abstract Design Techniques
- Graphical State Machines
- DSP Design
- Software/Hardware Codesign Thread Fundamentals

Clock domain

- Crossing Clock Domains
 - Metastability
 - Solution 1: Phase Control
 - Solution 2: Double Flopping
 - Solution 3: FIFO Structure
 - Partitioning Synchronizer Blocks
- Gated Clocks in ASIC Prototypes
- Clocks Module
- Gating Removal Runtime Statistics

Exercise : Show the effects of metastability when crossing asynchronous signal

Exercise : Measure the probability of metastability by simulating with random input changes

Implementing Math Functions

- Hardware Division
 - Multiply and Shift
 - Iterative Division
 - The Goldschmidt Method
- Taylor and Maclaurin Series Expansion
- The CORDIC Algorithm

Exercise : Example Design: I2S Versus SPDIF

Exercise : Example Design: Floating-Point Unit

Third Day

Reset Circuits

- Asynchronous Versus Synchronous
 - Problems with Fully Asynchronous Resets
 - Fully Synchronized Resets
 - Asynchronous Assertion, Synchronous Deassertion
- Mixing Reset Types
 - Nonresetable Flip-Flops
 - Internally Generated Resets
- Multiple Clock Domains

Exercise : Observe the differences between async and sync resets on flip-flops

Advanced Simulation

- Testbench Architecture

- Testbench Components
- Testbench Flow
- Main Thread
- Clocks and Resets
- Test Cases
- System Stimulus
 - MATLAB
 - Bus-Functional Models
- Code Coverage
- Gate-Level Simulations
- Toggle Coverage
- Run-Time Traps
 - Timescale
 - Glitch Rejection
 - Combinatorial Delay Modeling

Exercise : Understanding event bit group by synchronizing several threads

Floorplanning

- Design Partitioning
- Critical-Path Floorplanning
- Floorplanning Dangers
- Optimal Floorplanning
 - Data Path
 - High Fan-Out
 - Device Structure
 - Reusability
- Reducing Power Dissipation

Fourth Day

Static Timing Analysis

- Standard Analysis
- Latches
- Asynchronous Circuits
 - Combinatorial Feedback

PCB issues

- Power Supply
 - Supply Requirements
 - Regulation
- Decoupling Capacitors
 - Concept
 - Calculating Values
 - Capacitor Placement

Example Design: Microprocessor (RISCV NEORV32, IBX)

- SRC Architecture
- Synthesis Optimizations
 - Speed Versus Area
 - Pipelining
 - Physical Synthesis
- Floorplan Optimizations
 - Partitioned Floorplan

- Critical-Path Floorplan

Appendices

Memory Optimization in FPGA Design

- FPGA Memory Types
 - Flip-Flops (FF) vs LUT RAM vs Block RAM (BRAM) vs UltraRAM
- When NOT to use Flip-Flops
 - Resource explosion and routing impact
- Efficient Memory Mapping
 - Using BRAM for buffers and FIFOs
 - Inferring RAM in HDL
- Distributed RAM usage strategies

DSP and FFT Implementation

- DSP Blocks in FPGA
 - Multipliers and MAC units
- FFT Architectures
 - Radix-2 / Radix-4 basics
 - Pipelined vs iterative FFT
 - Fixed-point vs floating-point trade-offs
 - Throughput vs resource trade-offs

Exercise : Implement a dynamic FFT IP from the PS part