



ALT2 - FPGA Nios (Nios II / Nios V) implementation

Objectives

- Understand soft CPU concepts and Nios II / Nios V basics.
- Build SoCs in Platform Designer (Avalon-MM/Stream, clock/reset).
- Generate BSPs, bring up firmware, and debug via JTAG UART.
- Use GPIO, timers, interrupts; add UART/SPI/I²C comms.
- Configure SDRAM/Flash, linker placement, and boot options.
- Stream data with DMA, compare CPU vs DMA throughput.
- Run a small FreeRTOS app (tasks, ISRs, timing).
- Create a custom Avalon-MM IP and control it from C.
- Tune performance/power (caches, clocks, optimization).

Course Environment

- Theoretical course
 - PDF course material (in English) supplemented by a printed version for face-to-face courses.
 - Online courses are dispensed using the Teams video-conferencing system.
 - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
 - Practical activities represent from 40% to 50% of course duration.
 - Code examples, exercises and solutions
 - For remote trainings:
 - ▶ One Online Linux PC per trainee for the practical activities.
 - ▶ The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
 - ▶ QEMU Emulated board or physical board connected to the online PC (depending on the course).
 - ▶ Some Labs may be completed between sessions and are checked by the trainer on the next session.
 - For face-to-face trainings:
 - ▶ One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
 - ▶ One PC for two trainees when there are more than 6 trainees.
 - For onsite trainings:
 - ▶ An installation and test manual is provided to allow preinstallation of the needed software.
 - ▶ The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

Day 1

Nios architecture & SoC overview

- Soft CPU idea (fabric vs hard CPU).
- Nios II vs Nios V (high-level).
- Interconnect: Avalon-MM / Stream.
- Clock/reset domains.
- Memory map concept.

Exercise: Board & SoC tour

Tools & Platform Designer flow

- Quartus project basics.
- Add IP, connect masters/slaves.
- Clock source & reset bridges.
- HDL generation & top-level.
- Pin planner quick pass.

Exercise: GPIO “blinky” SoC

BSP & firmware bring-up

- BSP generation steps.
- HAL vs bare-metal.
- Linker script placement.
- JTAG UART console.
- Minimal driver calls.

Exercise: Hello over JTAG

GPIO, timer & interrupts

- PIO input/output usage.
- Interval timer basics.
- Interrupt controller path.
- Polling vs ISR patterns.
- Simple debounce idea.

Exercise: Button IRQ toggle

Day 2

Serial interfaces (UART/SPI/I²C)

- UART baud/format.
- SPI mode (CPOL/CPHA).
- I²C master ops.
- Blocking vs IRQ/DMA.
- Simple error checks.

Exercise: Comms demo

Memory & boot options

- On-chip RAM vs SDRAM.

- SDRAM controller timing.
- QSPI/Flash mapping.
- Boot: JTAG, Flash.
- Linker regions (.text/.data).

Exercise: SDRAM placement

DMA & throughput

- SG-DMA channels.
- M2M, M2P, P2M paths.
- Bursts and alignment.
- Cache coherency notes.
- Simple benchmarking.

Exercise: DMA vs CPU copy

RTOS quick start (FreeRTOS)

- Tasks & priorities.
- SysTick/timer tick.
- Queues/semaphores.
- ISR-safe APIs.
- Stack/heap sizing.

Exercise: Two-task demo

Day 3

Custom Avalon-MM IP

- Component editor basics.
- Slave regs and address map.
- Read/Write stubs in HDL.
- Export IRQ (optional).
- Driver header in BSP.

Exercise: LED pattern IP

Nios V specifics & migration

- ISA/toolchain note.
- CSR/interrupt differences.
- BSP template changes.
- Rebuild flow in tools.
- Compatibility tips.

Exercise: Rebuild for Nios V

Performance & power

- CPU vs DMA balance.
- I/D cache choices.
- Compiler flags (-O2/-O3).
- Clock gating idea.
- Simple profiling.

Exercise: Cache on/off test

Debug & production wrap-up

- SignalTap capture.
- UART boot logs.

- Version/CRC tags.
 - Update script outline.
 - Factory test hooks.
- Exercise:** SignalTap UART