

## 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/I<sup>2</sup>C 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).

### 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 ramène 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.

# Plan du cours

## 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/I<sup>2</sup>C)

- UART baud/format.
- SPI mode (CPOL/CPHA).
- I<sup>2</sup>C 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