



## FCC4 - e6500 implementation

*This course covers the e6500 core present in NXP T2 and T4 SoCs*

### Objectives

- *This course has 6 main objectives:*
  - *Introducing the hypervisor privilege level, the capability of supporting several guest operating systems, performing load balancing and virtualization.*
  - *Clarifying the basic mechanisms required in a multicore system: atomic sequences, doorbell interrupts, messaging.*
  - *Learning the exception mechanism, focusing on the proxy unit and providing guidelines to implement nesting.*
  - *Explaining the operation and initialization of the MMU and caches.*
  - *Highlighting the cache and TLB coherency issues and explaining the snooping.*
  - *Describing the debug units.*

*A more detailed course description is available on request at [training@ac6-training.com](mailto:training@ac6-training.com)*

### Prerequisites

- *Experience of a 32-bit processor or DSP is mandatory.*

### 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*
- *Au début de chaque demi-journée 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

### e6500 CORE OVERVIEW

- *New concept of cluster*
- *Software compatibility with e500mc and e5500*

### MULTI-THREADING, 64-BIT MODE AND HYPERVISOR

- *Dual-Threading*
- *64-bit fixed-point arithmetic*
- *e6500 hypervisor privilege level*
- *Collaboration between guest OS and hypervisor to reload TLBs*
- *Messaging within a coherency domain*

## INSTRUCTION PIPELINE

- *Dual issue, out of order execution*
- *Execution model*
- *Purpose of the isync instruction*
- *Instruction execution time, latency vs repeat rate*
- *Branch management*
- *Guarded memory*
- *Coding guidelines*

## FLOATING POINT UNIT

- *IEEE754 basics*
- *FPU operation*
- *Fully pipelined FPU*

## ALTIVEC UNITS

- *Vector vs scalar operation*
- *Altivec registers, VSCR initialization*
- *VRSAVE use in order no to save / restore all volatile vector registers*
- *ANSI C extension to support vector operators*
- *Altivec implementation on the e6500 : the VALU and the VPU execution units*
- *EABI extension to support Altivec*

## THE EXCEPTION MECHANISM

- *Exception management: building the handler table through IVPR,IVOR registers*
- *Finding the exact exception cause through syndrome registers*
- *Requirements to support exception nesting*
- *Interrupt proxy*
- *Multicore exceptions, doorbells and messages*
- *Integrated timers, time base, decrementer, FIT and WDT*
- *Watchdog service routine*

## DATA AND INSTRUCTION PATHS

- *Exclusive resource management*
- *Implementation of a spin lock routine*
- *wait instruction*
- *Decorated storage facility*
- *Memory barriers*

## THE MEMORY MANAGEMENT UNIT

- *Address translation, understanding the interim 86-bit virtual address*
- *Process protection through TID*
- *Two-level MMU architecture*
- *TLB organization*
- *Software TLB reload*
- *TLB invalidate local instruction*
- *Taking benefit of large pages*
- *" Virtualization fault*
- *" Hardware TLB reload for 4-KB page*
- *" TLB parity protection*
- *" Logical to Real Address Translation cache*

## L1 AND L2 CACHES, SNOOPING

- *Cache basics*
- *e6500 L1 cache: Write-Through operation*
- *Clustered shared L2 cache*
- *Cache related instructions*
- *L1 and L2 error checking and correction, error injection*

## CORE AND CLUSTER POWER MANAGEMENT

- *Connection to platform PM unit*
- *Power states, thread, core and cluster*
- *Wait for interrupt instruction*
- *AltiVec drowsy state*

## DEBUG

- *Performance monitor*
- *Debug interrupt*
- *Instruction and data breakpoints, programming address ranges*
- *Debug data acquisition message*
- *Debug Notify Halt instruction*
- *Nexus trace*
- *Enabling and disabling ownership trace*