FCC1 - e500mc implementation

This course covers the e500mc core present in 32-bit QorIQ SoCs

Objectives

- This course provides a detailed description of the e500mc internal architecture as well as the associated low level routines.
- Coherency mechanisms required in multiple e500mc platforms are explained through sequences.
- All mechanisms required in a multiple core system are described: atomic sequence through lwarx/stwxc. instruction pair, doorbell interrupts.
- The course focuses on the benefits of the hypervisor: running several operating systems, partitioning, load balancing and virtualization.
- The operation of the MMU is studied, particularly the TLB software reload routines.
- The course details the interrupt proxy unit and provides guidelines to implement nesting.
- Note that for on-site course, the contents can be tailored to specific customer needs.

- This course has been designed in collaboration with NXP

A more detailed course description is available on request at formation@ac6-formation.com

Prerequisites

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

Exercise: The environment used to build and debug software labs are based on the GNU compiler / linker and the debugger from Lauterbach.

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- 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

CORE ARCHITECTURE

- Block diagram
- CoreNet interface
- Highlighting differences between e500 and e500mc

HYPERVISOR

- Privilege levels: user, guest supervisor, hypervisor
- Logical partition
- Hypervisor call instruction

- Bare-metal operation

PIPELINE

- e500mc pipeline implementation
- Issue queue resource requirements
- Execution model
- Branch management: dynamic prediction
- Guarded memory

INTERNAL DATA / INSTRUCTION PATHS

- L1 and L2 cache loading, hit under miss, miss under miss
- The load miss queue
- The store miss merging mechanism
- Clarifying the difference between msync and lwsync

e500mc USER LEVEL PROGRAMMING

- Implementing atomic sequences in multiple core systems, mdors instruction
- Decorated load and store instructions
- Integer arithmetic and logic instructions
- FPU operation : FPSCR register, IEEE vs non-IEEE mode
- Float load / store instructions
- Float arithmetic instructions
- Convert instructions
- The EABI

SUPERVISOR / HYPERVISOR LEVEL PROGRAMMING

- Accessing special registers, understanding the required synchronizations
- Implementing low power modes, wait instruction
- Core timers

THE EXCEPTION MECHANISM

- Exception management: building the handler table through IVPR,IVOR registers
- Finding the exact exception cause through syndrome registers
- New machine check features
- Interrupt proxy
- Doorbell interrupts

THE MEMORY MANAGEMENT UNIT

- 4 GB effective address space, 64 GB real address space
- Address translation, understanding the interim 48-bit virtual address
- WIMGE attributes
- Two-level MMU architecture
- Software TLB reload
- Managing a page descriptor table in a SMP system
- Virtualization fault
- External PID load and store instructions

L1 AND L2 CACHES, SNOOPING

- Cache basics
- L1 data cache flush
- L2 cache organization

- Cache coherency basics
- The MESI L1 data line states
- MESI snooping sequences involving two e500mc and a PCI Express master
- Cache-to-cache transactions
- Cache related instructions
- Cache entry locking
- Stashing capability
- L1 and L2 error checking and correction, L2 cache error injection
- Write shadow mode

DEBUG

- Performance monitor
- Nexus debug unit
- Instruction and data breakpoints
- Debug data acquisition message