



## **PG2 - PowerPC System Design**

**This course has been designed for developers involved in a PowerPC development who want to understand generic mechanisms**

### **Objectives**

- This course explains the objectives of mechanisms used to boost the performance and the way they are implemented in various PowerPCs : cache / cache coherency, pipeline, MMU, exceptions.
- This gives to the attendees a wider overview of the state of the art in these domains.
- The course details the instructions required to write program in supervisor mode to adapt the behaviour of the core to specific needs.
- A tutorial designed by Mr Guillaume PERON, PowerPC expert since 1995, is used to quickly understand PowerPC low level programming.
- It clarifies the use of sections required for good management of caches and memory.
- Task switch requirements are highlighted.
- Debug facilities implemented in PowerPCs (hardware breakpoints, real-time trace, watchpoints) are studied through the use of Lauterbach TRACE32 debugger.

*A lot of programming examples have been developed by ACSYS to explain the PowerPC assembly language.  
-They have been developed with GNU compiler and are executed under Lauterbach debugger.*

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

### **Prerequisites**

- Basic knowledge of processor or DSP.

### **Plan**

#### **PowerPC PROGRAMMING**

- PowerPC programming environment : 32-bit PowerPC architecture, Book E, 64-bit architecture
- Register set, GPR vs SPR, HID registers
- Data type instantiation for PowerPC
- Pointers management (Addressing modes)
- User and supervisor functions call and return (EABI, C-to-assembly interface)
- Sections, benefits of small data sections
- Locating code and data in memory , linker command file
- Reset, what is to be done before calling the main() : Cstart program

## **PIPELINE**

- Superscalar operation
- Mechanisms used to boost performance : branch prediction, branch target address cache, link stack
- Guidelines to optimize execution time
- Serializations, isync instruction, determining when this instruction is really required

## **DATA PATH AND DECOUPLING**

- Highlighting the frequency domains present in PowerPC : core and bus interface
- Decoupling the core from cache and bus through load and store buffers
- Default ordering of load and store transactions
- Enforcing the ordering through eieio (called mbar in Book E) and sync (called msync in Book E) instructions
- Purpose of the Guarded attribute
- Consequence for high level development of IO drivers

## **MEMORY MANAGEMENT UNIT**

- Requirements for kernels enabling dynamic memory mapping
- Single process multi-thread versus multiprocess multi-thread kernels
- Objectives of the MMU : page protection, definition of page attribute, address translation
- Segment and page translation
- Table search mechanisms : benefits of a software table search
- Operation of TLB caches
- TLB programming, static initialization

## **CACHE AND DATA COHERENCY**

- Introduction to cache memory
- Cache organization
- Write policies
- Replacement algorithms
- Data flow between external main memory
- Distinguishing private memory that is accessed only by the core and shared memory that can be accessed by the core and other masters (DMA or CPU)
- Software enforced coherency
- Hardware enforced coherency

## **EXCEPTION MECHANISM**

- Software exceptions vs interrupts
- Save / restore registers
- Organization of an exception handler : prolog, body and epilog
- How to find the cause of the exception, syndrome registers
- Design of a generic exception handler based on a vector table
- Interrupt management, addition of a critical interrupt in Book E
- Integrated interrupt controller
- Requirements for interrupt nesting

## **MULTITASK**

- Management of boolean semaphores, lwarx / stwxc. instruction pair
- Stack switch, use of SPRG registers
- Definition of the set of registers that determine the stack state
- Management of task lists in single and multi processor systems

## PowerPC DEBUG SOLUTIONS

- On-chip debug logic
- Restrictions of JTAG debug
- Hardware breakpoints
- Real-time trace
- Debugging software when caches are active
- The performance monitor

## Renseignements pratiques

**Duration : 4 days**  
**Cost : 1950 € HT**



SARL au capital de 15400€ - SIRET 449 597 103 00026 - RCS Nanterre - NAF 722C - Centre de Formation : 19, rue Pierre Curie - 92400 Courbevoie  
Siège social et administration : 21, rue Pierre Curie - 92400 Courbevoie - Tél. 01 41 16 80 10 - Fax. 01 41 16 07 78

Last site update: Tue 22 May 2012 10:50:29 CEST

<http://www.ac6-formation.com/>