

POWER PC SYSTEM DESIGN

Ref : 003830A

Duration : 4 days

OBJECTIVES

- This course has been designed for developers that are going to be involved in a PowerPC development, and intend to understand generic mechanisms specific to PowerPC
- It 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. It presents the assembly language to write more efficient programs in C. 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

RELATED COURSES

- MVD has designed a large set of PowerPC courses that detail the operation of particular PowerPCs from Freescale and IBM Microelectronics (see our web site)
- C language for Embedded Applications (002603A)

PREREQUISITES

- A basic understanding of microprocessors and microcontrollers
- A basic understanding of assembler or C programming would be useful but not essential

PARTNERS

- This training course is approved by Freescale

**WIND RIVER****NeoMore**

Contact

Tel : 05 62 13 52 32
Fax : 05 61 06 72 60
training@mvd-fpga.com

Course also available
customized

Next sessions, see : <http://www.mvd-fpga.com/en/formationsCalend.html>

TOPICS

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

DOCUMENTATION

- Training manuals will be given to attendees during training in print.

- 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
- How it communicates with the debug station : BDM or JTAG connection
- Hardware breakpoints
- Real-time trace
- Debugging software when caches are active
- The performance monitor