Formation MPC837XE implementation: This course covers PowerQUICC II Pro MPC837XE - Processseurs PowerPC: NXP Power CPUs

**FPQA - MPC837XE implementation**

This course covers PowerQUICC II Pro MPC837XE

### Objectives
- The course focuses on the internal interconnect architecture, based on the CSB bus.
- Cache coherency protocol is introduced in increasing depth.
- The 32-bit e300 core is viewed in detail, especially the MMU and the cache.
- The boot sequence and the clocking are explained.
- The course focuses on hardware implementation of the MPC837X.
- An in-depth description of the PCI controller is performed.
- The course highlights both hardware and software implementation of gigabit / fast / Ethernet controllers and the parameterizing of the level 2, 3 and 4 acceleration mechanisms.
- The USB interface is also detailed.
- The course explains how to initialise both the Serdes block and the SATA controller to detect and communicate with an external hard disk.
- Generation of a Linux image and Root File System by using LTIB can also be included into the training.
- This course has been delivered several times to companies developing telecom infrastructure equipments.

A lot of programming examples have been developed by ACSYS to explain the boot sequence and the operation of complex peripherals, such as SATA and Ethernet.

- They have been developed with Diab Data compiler and are executed under Lauterbach debugger.
- A more detailed course description is available on request at training@ac6-training.com

### Prerequisites and related courses
- Experience of a 32 bit processor or DSP is mandatory.
- The knowledge of the following interconnect standards may be required:
  - PCI-X, see our course reference IC3 - PCI-X 2.0 course
  - PCI Express, see our course reference IC4 - PCI Express 3.0 course
  - Gigabit Ethernet, see our course reference N1 - Ethernet and switching course
  - USB 2.0, see our course reference IP2 - USB 2.0 course
  - S-ATA, see our course reference IS3 - Serial ATA III course

### Plan

**INTRODUCTION TO MPC837X**
## Overview

- General features
- Enhancements compared to MPC834X
- Memory map
- Block diagram: characteristics of each of the 3 internal modules e300 core, Platform and peripherals
- Features of the MPC8377E, MPC8378E and MPC8279E
- Application examples

## THE e300 CORE

### THE INSTRUCTION PIPELINE

- Pipeline
- Branch processing unit
- Branch instructions

### DATA PATHS

- Load / store architecture
- Load / store buffers
- Sync and eieio instructions

### CACHES

- Cache basics
- Cache locking
- L1 caches
- Shared resource management, lwarx and stwcx. instructions
- Cache coherency mechanism, snooping, related signals
- Management of cache enabled pages shared with PCI DMAs
- Cache related instructions

### SOFTWARE IMPLEMENTATION

- e300 registers
- Addressing modes, load / store instructions
- Integer instructions
- IEEE754 basics, floating points numbers encoding
- Floating point load / store instructions
- Floating point arithmetical instructions
- The PowerPC EABI
- Linking an application with Diab Data, parameterizing the linker command file

### THE MMU

- Introduction to real, block and segmentation / pagination translations
- Real mode restrictions
- Memory attributes and access rights definition
- Virtual space benefit, page protection through segmentation
- TLBs organization, related instructions, MMU initialization routine
- Segmentation: process ID definition
- Pagination: PTE table organization, tablesearch algorithm
- MMU implementation in real-time sensitive applications
<table>
<thead>
<tr>
<th>THE EXCEPTION MECHANISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save / restore registers</td>
</tr>
<tr>
<td>Exception management mechanism</td>
</tr>
<tr>
<td>RI bit use in non-maskable interrupt handlers</td>
</tr>
<tr>
<td>Registers updating according to the exception cause</td>
</tr>
<tr>
<td>Requirements to allow exception nesting</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>THE DEBUG PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTAG emulation, restrictions</td>
</tr>
<tr>
<td>Real time trace requirements</td>
</tr>
<tr>
<td>Hardware breakpoints</td>
</tr>
<tr>
<td>Performance monitor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>THE PLATFORM CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power management control</td>
</tr>
<tr>
<td>Reset causes</td>
</tr>
<tr>
<td>Configuration signals sampled at reset</td>
</tr>
<tr>
<td>Reset configuration words source</td>
</tr>
<tr>
<td>Boot from SPI</td>
</tr>
<tr>
<td>Utilization of the I2C boot sequencer</td>
</tr>
<tr>
<td>Clocking in PCI Host mode, system clock domains</td>
</tr>
<tr>
<td>External clock inputs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLATFORM CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address translation and mapping</td>
</tr>
<tr>
<td>Arbiter and bus monitor</td>
</tr>
<tr>
<td>General purpose inputs / outputs</td>
</tr>
<tr>
<td>Timers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>THE DDR2 MEMORY CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDR-SDRAM operation</td>
</tr>
<tr>
<td>Jedec specification basics, mode register initialization, bank selection and precharge</td>
</tr>
<tr>
<td>Differences between DDR1 and DDR2</td>
</tr>
<tr>
<td>Command truth table</td>
</tr>
<tr>
<td>ECC error correction</td>
</tr>
<tr>
<td>Initial configuration following Power-on-Reset</td>
</tr>
<tr>
<td>Timing parameters programming</td>
</tr>
<tr>
<td>Initialization routine</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCAL BUS CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplexed or non-multiplexed address and data buses</td>
</tr>
<tr>
<td>Burst support</td>
</tr>
<tr>
<td>Dynamic bus sizing</td>
</tr>
<tr>
<td>GPCM, UPMs states machines</td>
</tr>
<tr>
<td>NAND flash controller</td>
</tr>
</tbody>
</table>
PCI BUS INTERFACES

- Bridge features
- Data flows: Read prefetch and write posting FIFOs
- Inbound transactions handling, Outbound transactions handling
- PCI bus arbitration
- PCI hierarchy configuration when operating as host

PCI EXPRESS INTERFACE

- Implementation of a unique VC
- Selectable operation as agent or root complex
- Address translation
- Error management
- Power management

INTEGRATED DMA CONTROLLER

- Priority between the 4 channels
- Support for cascading descriptor chains
- Selectable hardware enforced coherency
- Concurrent execution across multiple channels with programmable bandwidth control
- Messaging unit

INTEGRATED PROGRAMMABLE INTERRUPT CONTROLLER

- Definition of interrupt priorities
- System critical interrupt
- Interrupt management, vector register
- Requirements to support nesting
- Machine check interrupts

INTEGRATED PERIPHERALS

ENHANCED SECURE DEVICE HOST CONTROLLER

- Introduction to MMC and SD card
- Storing and executing commands targeting the external card
- Multi-block transfers
- Moving data by using the dedicated DMA controller
- Read transfer sequence
- Write transfer sequence
- Dividing large data transfers
- Card insertion and removal detection

SECURITY ENGINE

- Overview of the encryption mechanism
- Introduction to DES, 3DES and AES algorithms
- Data packet descriptors
- Crypto channels
- Link tables
THE ETHERNET CONTROLLERS
- MAC address recognition, 256-entry hash table for unicast and multicast
- Interface with the PHY, RGMII, RTBI or SGMII
- Buffer descriptors management
- Flow control
- Level 2, 3 and 4 hardware acceleration mechanisms
- Quality of service support

SATA CONTROLLER
- SATA basics
- 2 ports compliant with SATA 2.5, 1.5 and 3 Gbps operation
- Electrical specification
- Bringing the SATA controller online/offline
- Native command queuing, command descriptor
- Interrupt coalescing
- Initialization steps

THE USB 2.0 CONTROLLER
- Dual-role (DR) operation
- EHCI implementation
- Periodic Frame List
- UTMI / ULPI interfaces to the transceiver
- OTG support
- Endpoints configuration

LOW SPEED PERIPHERALS
- Description of the NS 16450/16550 compliant Uarts
- I2C protocol fundamentals
- Transmit and receive sequence
- SPI protocol basics
- Master vs slave operation

Linux Target Image Builder (LTIB)

GENERATING THE LINUX KERNEL IMAGE
- Introducing the tools required to generate the kernel image
- What is required on the host before installing LTIB
- Common package selection screen
- Common target system configuration screen
- Building a complete BSP with the default configurations
- Creating a Root Filesystems image
- e-configuring the kernel under LTIB
- Selecting user-space packages
- Setup the bootloader arguments to use the exported RFS
- Debugging Uboot and the kernel by using Trace32
- Command line options
- Adding a new package
- Other deployment methods
- Creating a new package and integrating it into LTIB
  - A lot of labs have been created to explain the usage of LTIB
Renseignements pratiques

Durée : 5 jours
Prix : 2100 € HT