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.
- A long introduction to DDR SDRAM operation is done before studying the DDR2 SDRAM controller.
- 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 cours IC3 - PCI-X 2.0
  - PCI Express, see our course reference cours IC4 - PCI Express 3.0
  - Gigabit Ethernet, see our course reference cours N1 - Ethernet and switching
  - USB 2.0, see our course reference cours IP2 - USB 2.0
  - S-ATA, see our course reference cours IS3 - Serial ATA III

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.

Plan du cours

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

THE EXCEPTION MECHANISM

- Save / restore registers
- Exception management mechanism
- RI bit use in non-maskable interrupt handlers
- Registers updating according to the exception cause
- Requirements to allow exception nesting

THE DEBUG PORT

- JTAG emulation, restrictions
- Real time trace requirements
- Hardware breakpoints
- Performance monitor

THE PLATFORM CONFIGURATION

POWER, RESET AND CLOCKING

- Power management control
- Reset causes
- Configuration signals sampled at reset
- Reset configuration words source
- Boot from SPI
- Utilization of the I2C boot sequencer
- Clocking in PCI Host mode, system clock domains
- External clock inputs

PLATFORM CONFIGURATION

- Address translation and mapping
- Arbiter and bus monitor
- General purpose inputs / outputs
- Timers

THE DDR2 MEMORY CONTROLLER

- DDR-SDRAM operation
- Jedec specification basics, mode register initialization, bank selection and precharge
- Differences between DDR1 and DDR2
- Command truth table
- ECC error correction
- Initial configuration following Power-on-Reset
- Timing parameters programming
- Initialization routine

LOCAL BUS CONTROLLER

- Multiplexed or non-multiplexed address and data buses
- Burst support
- Dynamic bus sizing
- GPCM, UPMs states machines

- NAND flash controller

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