FPQ5 - MPC8309 implementation

This course covers PowerQUICC II Pro MPC8309, MPC8306 and MPC8306S

Objectives

- The course explain the architecture of the MPC8309, particularly the operation of the coherency module that interconnects the e300 to memory and high-speed interfaces.
- Cache coherency protocol is introduced in increasing depth.
- The e300 core is viewed in detail, especially the MMU .
- The boot sequence and the clocking are explained.
- The course focuses on the hardware implementation of the MPC8309.
- A long introduction to DDR SDRAM operation is done before studying the DDR1/2 SDRAM controller.
- The course describes the sophisticated QoS mechanisms supported by the UCC Ethernet Controller.
- Implementation of Precise Time Protocol is also studied.
- Generation of a Linux image and Root File System by using LTIB can also be included into the training.

- Products and services offered by ACSYS:
  - ACSYS is able to assist the customer by providing consultancies
  - Typical expertises are done during board bringup, hardware schematics review, software debugging, performance tuning.
  - Note that ACSYS has delivered several consultancies on NXP Netcomm SoCs to companies developing avionic equipments.

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

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

- *They have been developed with Diab Data compiler and are executed using Lauterbach debugger.*

Prerequisites and related courses

- Experience of a 32-bit processor or DSP is mandatory.
- The following courses could be of interest:
  - Ethernet and switching, reference N1 - Ethernet and switchingcourse
  - IEEE1588, reference N2 - IEEE1588 - Precise Time Protocolcourse
  - PCI, reference IC1 - PCI 3.0course
  - USB Full Speed High Speed and USB On-The-Go, reference IP2 - USB 2.0course
  - SD / MMC, reference IS2 - eMMC 5.0course
  - CAN bus, reference IA1 - CAN buscourse

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - 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.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed by quizzes offered at the end of various sections to verify that the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

INTRODUCTION TO MPC8309, MPC8306 and MPC8306S

SOC ARCHITECTURE

- Internal architecture
- Highlighting data paths inside the MPC8309
- Highlighting differences between MPC8309, MPC8306 and MPC8306S
- Application examples

THE e300c3 CORE

THE INSTRUCTION PIPELINE

- Superscalar operation, out-of-order execution, register renaming, serializations, isync instruction
- Branch processing unit, prediction
- Coding guidelines

DATA AND INSTRUCTION PATHS

- Load / store buffers
- Sync and eieio instructions
- Store gathering mechanism

CACHES

- Cache basics
- L1 caches
- Cache coherency mechanism, snooping, related signals
- Memory coherency required attribute
- The MEI state machine
- Basic snoop requests
- Management of cache enabled pages shared with DMAs
- Cache related instructions
- Software enforced cache coherency
- Cache flush routine

SOFTWARE IMPLEMENTATION

- PowerPC architecture specification, the 3 books UISA, VEA and OEA

- e300 registers
- Addressing modes, load / store instructions
- Floating point arithmetical instructions
- The PowerPC EABI
- Linking an application with Diab Data

THE MMU

- Introduction to real, block and segmentation / pagination translations
- Real mode restrictions
- Memory attributes and access rights definition
- TLBs organization
- Segment-translation
- Page-translation
- MMU implementation in real-time sensitive applications

THE EXCEPTION MECHANISM

- Critical interrupt, automatic nesting
- Exception management mechanism
- Registers updating according to the exception cause
- Requirements to allow exception nesting

THE DEBUG PORT

- JTAG emulation, restrictions
- Hardware breakpoints

THE PLATFORM CONFIGURATION

POWER, RESET AND CLOCKING

- Power management control
- Configuration signals sampled at reset
- Output signals state during reset
- Reset configuration words source
- Clocking

PLATFORM CONFIGURATION

- Address translation and mapping
- Arbiter and bus monitor
- Timers
- Dynamic power management

THE DDR2 MEMORY CONTROLLER

- DDR-SDRAM operation
- Jedec specification basics
- On-Die termination and calibration
- Hardware interface
- Bank activation, read, write and precharge timing diagrams, page mode
- Initial configuration following Power-on-Reset
- Timing parameters programming
- Initialization routine

ENHANCED LOCAL BUS CONTROLLER

- Multiplexed or non-multiplexed address and data buses

- Dynamic bus sizing
- GPCM, UPMs states machines
- Nand Flash Controller
- Booting from NAND flash

PCI BUS INTERFACE

- Bridge features
- Read prefetch and write posting FIFOs
- Inbound transactions handling, outbound transactions handling
- PCI bus arbitration

ENHANCED SECURE DEVICE HOST CONTROLLER

- Storing and executing commands targeting the external card
- Multi-block transfers
- Moving data by using the dedicated DMA controller
- Dividing large data transfers
- Card insertion and removal detection

DMA ENGINES

- DMA engine 1
  - Transfer control descriptor format
  - Channel-to-channel linking mechanism
  - Scatter/gather DMA processing
- DMA engine 2
  - Data chaining and direct mode
  - Priority between the 4 channels

INTEGRATED PROGRAMMABLE INTERRUPT CONTROLLER

- Definition of interrupt priorities
- System critical interrupt
- Interrupt management, vector register
- Machine check interrupts

FLEXCAN MODULE

- Hardware interface
- 64 message buffers (MB) of zero to eight bytes data length
- Individual Rx mask registers per message buffer
- Powerful Rx FIFO ID filtering
- Management of remote frames, overload frames
- Programmable transmission priority scheme
- Time stamp based on 16-bit free-running timer
- Global network time

THE USB 2.0 CONTROLLER

- Dual-Role operation
- EHCI implementation
- ULPI interfaces to the transceiver
- Dedicated DMA channels
- Endpoints configuration

LOW SPEED PERIPHERALS

- DUART
- I2C controller
- SPI controller

QUICC ENGINE

SYSTEM INTERFACE AND CONNECTION TO EXTERNAL COMMUNICATION PORTS

- Serial DMA
- Multi-threading
- NMSI vs TDM
- Baud-rate generators
- QUICC engine timers

BUFFER MANAGEMENT

- Utilization of Buffer Descriptors
- Chaining descriptors into rings
- Frame boundary definition

UNIFIED COMMUNICATION CONTROLLERS

- Handling UCC interrupts
- Initialization sequence
- UCC for fast protocols, virtual FIFOs
- Defining Tx- and Rx-FIFO thresholds

UCC ETHERNET CONTROLLER

- Physical interfaces to transceiver
- Auto-negotiation
- IP header checksum
- Frame filtering and address recognition
- Quality of Service
- Ethernet scheduler, traffic shaper
- BD and Parameter RAM description
- Ethernet host command set

IEEE1588 ASSIST

- Timestamp unit key features
- Real Time Clock
- How QuiccEngine and host software interact

QUICC MULTI-CHANNEL CONTROLLER

- QMC and serial interface
- UCC Base and Global multichannel parameters
- Channel-specific HDLC parameters
- QMC host commands

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

Duration : 5 days
Cost : 2100 € HT