FPQ9 - MPC8360E implementation

This course covers PowerQUICC II Pro MPC8360E

Objectives

- The course explains how to optimize the internal traffics flowing through the interconnect 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 MPC8360E.
- A long introduction to DDR SDRAM operation is done before studying the DDR2 SDRAM controllers.
- An in-depth description of the PCI controller is performed.
- Two controllers present in the QuiccEngine are particularly studied : Ethernet on UCC and multi-channel, and the course explains how to implement an inter-working between TDM lines and Ethernet.
- The course highlights both hardware and software implementation of gigabit / fast / Ethernet controllers.
- The USB controller is also detailed.
- 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 USB 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, see our course reference IC1 - PCI 3.0course
  - Gigabit Ethernet, see our course reference N1 - Ethernet and switchingcourse
  - USB 2.0, see our course reference IP2 - USB 2.0course

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 MPC8360E

Overview

- Highlighting data paths inside the MPC8360E
- Block diagram : characteristics of each of the 3 internal modules e300 core, Platform, QuiccEngine
- Software migration from MPC82XX/MPC85XX families

THE e300 CORE

THE INSTRUCTION PIPELINE

- e300 pipeline
- Branch processing unit
- Coding guidelines

DATA PATHS

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

CACHES

- Cache basics
- Cache locking
- L1 caches
- Cache coherency mechanism
- The MEI state machine
- Management of cache enabled pages shared with PCI DMAs
- Software enforced cache coherency
- Cache flush routine

SOFTWARE IMPLEMENTATION

- e300 registers
- Addressing modes, load / store 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

- Thread vs process
- Real mode restrictions
- Memory attributes and access rights definition

- Virtual space benefit
- TLBs organization
- Segment-translation
- Page-translation
- MMU implementation in real-time sensitive applications

THE EXCEPTION MECHANISM

- Exception management mechanism
- Registers updating according to the exception cause
- Requirements to allow exception nesting

THE DEBUG PORT

- JTAG emulation, restrictions
- Hardware breakpoints
- Performance monitor

THE PLATFORM CONFIGURATION

POWER, RESET AND CLOCKING

- DC and AC electrical characteristics
- Configuration signals sampled at reset
- Reset configuration words source
- Utilization of the I2C boot sequencer
- PCI Host / Agent configuration
- Boot memory space
- 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
- Dynamic power management

THE DDR2 MEMORY CONTROLLER

- Jedec specification basics
- On-Die termination and calibration
- Differences between DDR1 and DDR2
- Command truth table
- Hardware interface
- ECC error correction
- DDR-SDRAM controller overview
- Address decode
- Timing parameters programming
- Initialization routine

LOCAL BUS CONTROLLER

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

PCI BUS INTERFACES

- Bridge features
- Data flows
- Inbound transactions handling, Outbound transactions handling
- PCI bus arbitration
- PCI hierarchy configuration when operating as host

INTEGRATED DMA CONTROLLER

- Priority between the 4 channels
- Scatter / gathering
- Concurrent execution across multiple channels
- Messaging unit

INTEGRATED PROGRAMMABLE INTERRUPT CONTROLLER

- Interrupt sources
- Definition of interrupt priorities
- System critical interrupt
- Requirements to support nesting

SECURITY ENGINE

- Overview of the encryption mechanism
- Introduction to DES, 3DES and AES algorithms
- Crypto channels
- Snooping by caches
- Implementation of IPSEC

LOW SPEED PERIPHERALS

- Description of the NS16450/16550 compliant Uarts
- FIFO mode
- Flow control signal management
- I2C protocol fundamentals
- Transfer timing diagrams, SCL and SDA pins
- Transmit and receive sequence

QUICC ENGINE

SYSTEM INTERFACE AND CONNECTION TO EXTERNAL COMMUNICATION PORTS

- Serial DMA
- QUICC engine external requests
- Multi-threading
- NMSI vs TDM
- CMX registers
- Baud-rate generators

BUFFER MANAGEMENT

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

SERIAL PERIPHERAL INTERFACE

- Introduction to SPI protocol
- SPI modes of operation in QUICC engine mode
- Transmit and receive sequence

UNIFIED COMMUNICATION CONTROLLERS

- UCC feature set
- Handling UCC interrupts
- Initialization sequence
- UCC as slow communications controllers, UART mode
- UCC for fast protocols, virtual FIFOs

UCC ETHERNET CONTROLLER

- Physical interfaces to transceiver
- Auto-negotiation
- Termination and interworking modes of operation
- IP header checksum
- Frame filtering and address recognition
- Header parsing
- Quality of Service
- Ethernet scheduler, traffic shaper
- BD and Parameter RAM description
- Ethernet statistics, MIB

IEEE1588 ASSIST

- Overview of the IEEE1588 standard
- Timestamp unit key features
- How QuiccEngine and host software interact
- PTP frame reception
- PTP frame transmission

MULTI-CHANNEL CONTROLLER

- Comparison with MPC82XX CPM MCC
- Channel-specific HDLC parameters
- Channel extra parameters
- MCC exceptions
- MCC host commands

QUICC MULTI-CHANNEL CONTROLLER

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

USB

- Host controller limitations
- Endpoint parameters block pointer
- Frame number
- USB BD ring
- 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