Formation P102X QorIQ implementation: This course covers NXP QorIQ P1020/P1011, P1021/P1012, P1022/P1013, P1023/P1017, P1024/P1015, P1025/P1016 - Processeurs PowerPC: NXP Power CPUs

FCQ11 - P102X QorIQ implementation

This course covers NXP QorIQ P1020/P1011, P1021/P1012, P1022/P1013, P1023/P1017, P1024/P1015, P1025/P1016

Objectives

- The course clarifies the architecture of the P102X, particularly the operation of the coherency module that interconnects the e500s to memory and high-speed interfaces.
- Cache coherency protocol is introduced in increasing depth.
- The e500 core is viewed in detail, especially the SPE unit that enable vector processing.
- The boot sequence and the clocking are explained.
- The course focuses on the hardware implementation of the P102X.
- A long introduction to DDR SDRAM operation is done before studying the DDR2/3 SDRAM controller.
- An in-depth description of the PCI-Express port is done.
- The course highlights both hardware and software implementation of gigabit / fast / Ethernet controllers.
- Communication interfaces are explained according to the exact reference of the SoC: either TDM or QuicEngine or DPAA.
- AC6 has developed an optimized SPE based FFT coded in assembler language.
- Performance for 1024 complex floating point single precision samples is:
  - 91_386 core clock cycles without reverse ordering, 94_124 with reverse ordering
- Performance for 4096 complex floating point single precision samples is:
  - 470_778 core clock cycles without reverse ordering, 511_227 with reverse ordering
- For any information contact formation@ac6-formation.com

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 following courses could be of interest:
  - Ethernet and switching, reference cours N1 - Ethernet and switching
  - IEEE1588, reference cours N2 - IEEE1588 - Precise Time Protocol
  - PCI express gen2, reference cours IC4 - PCI Express 3.0
  - USB Full Speed High Speed and USB On-The-Go, reference cours IP2 - USB 2.0
  - SD / MMC, reference cours IS2 - eMMC 5.0
## Plan

### INTRODUCTION TO P102X

**SOC ARCHITECTURE**
- Internal data flows, OCEAN switch fabric, packet reordering
- Implementation examples
- Address map, ATMU, OCEAN configuration
- Local vs external address spaces, inbound and outbound address decoding

### e500 CORES

**THE INSTRUCTION PIPELINE**
- Dual-issue superscalar operation
- Execution units
- Dynamic branch prediction

**DATA AND INSTRUCTION PATHS**
- The Core Complex Bus
- Store miss merging and store gathering
- Memory access ordering
- Lock acquisition and import barriers

**THE MEMORY MANAGEMENT UNIT**
- The first level MMU and the second level MMU, consistency between L1 and L2 TLBs
- TLB software reload
- Process protection
- 36-bit real addressing

**CACHES**
- The L1 caches
- Software cache coherency
- Level 2 cache
- Allocation of data transferred by external masters into the cache: stashing
- e500 coherency module
- Snooping mechanism
- Stashing mechanism
- L2 cache locking

**PROGRAMMING**
- Differences between the new Book E architecture and the classic PowerPC architecture
- Floating Point units, Double-Precision FP
- Signal Processing APU (SPE)
## EXCEPTIONS
- Book E exception handling
- Syndrome registers
- Core timers

## DEBUGGING
- Performance monitoring
- JTAG emulation
- Watchpoint logic

## INFRASTRUCTURE

### RESET, CLOCKING AND INITIALIZATION
- Platform clock
- Voltage configuration selection
- Power-on reset sequence, using the I2C interface to access serial ROM
- Power management
- eSDHC boot
- eSPI boot ROM

### e500 COHERENCY MODULE
- I/O arbiter
- CCB arbiter
- CCB interface

### DDR3 SDRAM MEMORY CONTROLLER
- On-Die termination
- Calibration mechanism
- Mode registers initialization, bank selection and precharge
- Command truth table
- Hardware interface
- Bank activation, read, write and precharge timing diagrams, page mode
- ECC error correction
- Initialization routine

## ENHANCED LOCAL BUS CONTROLLER
- Multiplexed or non-multiplexed address and data buses
- Dynamic bus sizing
- GPCM, UPMs states machines
- NAND flash controller

## PCI EXPRESS INTERFACE
- 4-lane PCI Express interface
- Modes of operation, Root Complex / Endpoint
- Transaction ordering rules
- Programming inbound and outbound ATMUs
<table>
<thead>
<tr>
<th>PROGRAMMABLE INTERRUPT CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC in multiple-processor implementation</td>
</tr>
<tr>
<td>Understanding interrupt masking</td>
</tr>
<tr>
<td>Interprocessor interrupts</td>
</tr>
<tr>
<td>Per-CPU register usage, message registers</td>
</tr>
<tr>
<td>Nesting implementation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTEGRATED DMA CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for cascading descriptor chains</td>
</tr>
<tr>
<td>Scatter / gathering</td>
</tr>
<tr>
<td>Selectable hardware enforced coherency</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PERFORMANCE MONITOR AND DEBUG FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold events</td>
</tr>
<tr>
<td>Chaining, triggering</td>
</tr>
<tr>
<td>Watchpoint facility</td>
</tr>
<tr>
<td>Trace buffer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUTS/OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE ETHERNET CONTROLLERS</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>Address recognition, pattern matching</td>
</tr>
<tr>
<td>Buffer descriptors management</td>
</tr>
<tr>
<td>Physical interfaces</td>
</tr>
<tr>
<td>Buffer descriptor management</td>
</tr>
<tr>
<td>256-entry hash table for unicast and multicast</td>
</tr>
<tr>
<td>Management of VLAN tags and priority, VLAN insertion and deletion</td>
</tr>
<tr>
<td>Quality of service, managing several transmit and receive queues</td>
</tr>
<tr>
<td>TCP/IP offload engine, filer programming</td>
</tr>
<tr>
<td>IEEE1588 compliant time-stamping</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ENHANCED SECURE DEVICE HOST CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storing and executing commands targeting the external card</td>
</tr>
<tr>
<td>Multi-block transfers</td>
</tr>
<tr>
<td>Moving data by using the dedicated DMA controller</td>
</tr>
<tr>
<td>Dividing large data transfers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USB CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHCI implementation</td>
</tr>
<tr>
<td>Periodic Frame List</td>
</tr>
<tr>
<td>ULPI interfaces to the transceiver</td>
</tr>
<tr>
<td>OTG support</td>
</tr>
<tr>
<td>Endpoints configuration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECURITY ENGINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crypto channels</td>
</tr>
<tr>
<td>Sequence to subcontract a crypto job to SEC</td>
</tr>
<tr>
<td>Link tables</td>
</tr>
</tbody>
</table>
### Managing Interrupts

#### LOW SPEED PERIPHERALS
- Description of the NS16552 compliant Uarts
- I2C controller
- Enhanced SPI controller

#### TDM INTERFACE (P1022/P1013 AND P1024/P1015)
- Serial interface
- Network mode of operation with up to 128 time-slots
- DMA configuration
- End-of-frame interrupt
- Configuring the TDM for I2S Operation

#### DISPLAY INTERFACE UNIT (P1022/P1013)
- Display interfaces
- Display color depth
- Pixel structure, alpha-blending
- Utilization of area descriptor
- Moving images through the dedicated DMA channel

#### QUICC ENGINE (P1021/P1012 AND P1025/P1016)

#### OVERVIEW OF QUICC ENGINE
- Integrated RISC CPU
- Communication between Host CPU and QE RISC CPU

#### INTEGRATED INTERRUPT CONTROLLER
- Priority management
- Steering the interrupt source to either Low priority or High priority input of the platform PIC

#### SYSTEM INTERFACE AND CONNECTION TO EXTERNAL COMMUNICATION PORTS
- Serial DMA
- QUICC engine external requests
- NMSI vs TDM
- Enabling connections to TSA or NMSI

#### BUFFER MANAGEMENT
- Utilization of Buffer Descriptors
- Chaining descriptors into rings
- Parameter RAM independent of protocol

#### UNIFIED COMMUNICATION CONTROLLERS
- UCC as slow communications controllers, UART mode
- UCC for fast protocols, virtual FIFOs
**UCC HDLC CONTROLLER**
- Flow control
- Setting global parameters
- Describing the parameter RAM

**UCC TRANSPARENT CONTROLLER**
- Transparent data encapsulation, frame sync and frame CRC
- Describing the parameter RAM

**SERIAL INTERFACE**
- Connecting TDM lines
- Parameterizing the timings related to Rx/Tx clock, sync and data signals
- Connecting the TDM line to UCC using Rx/Tx routing tables

**MULTI-CHANNEL CONTROLLER ON UCC - UMCC**
- Comparison with MCC and QMC
- Connecting time-slots to logical channels through Rx/Tx routing tables
- Implementing Rx/Tx channel buffers
- Interrupt management
- Channel-specific HDLC parameters
- Per channel exception management
- UMCC host commands

**DATAPATH PROCESSING SUBSYSTEM (P1023/P1017)**

**DPAA OVERVIEW**
- Definitions: buffer, buffer pool, frame, frame queue, work queue, channel
- Data formats
- Frame formats
- Packet walk through

**QUEUE MANAGER**
- Objectives if this accelerator
- Frame description
- Structure of frame queues
- Frame queue state machine
- Multiway resource arbiter
- Work queues and channels
- Enqueue and dequeue portals
- Class and intra-class scheduling rules
- Dequeue dispatcher operation
- Message ring
- Stash transaction flow control and scheduling
- Congestion avoidance
- CoreNet initiator scheduling and priority

**BUFFER MANAGER**
- Objectives if this accelerator
Software portals
Direct connect portals
Software interface, Command register, Management Response registers
Buffer Pool State Change Notifications
Buffer pool size programming
Performance Monitor

FRAME MANAGER
- Objectives if this accelerator, parsing, classifying and distributing in-line/off-line packet
- FMAN submodules
- Rx BMI features
- Tx BMI features
- Offline parsing, host command features
- Frame processing manager
- FMAn controller
- Parser
- Key generator
- Policer

DATA PATH THREE-SPEED ETHERNET CONTROLLERS
- MAC address recognition
- 256-entry hash table for unicast and multicast
- Suspending the transmitter, handling pause packets
- RMON statistic counters, carry registers
- Client IEEE1588 timers

SECURITY ENGINE
- Job management using QMan interface
- Input / output rings
- Job descriptor parsing
- Sharing descriptors
- Selecting the authentication / cryptographic algorithm
- Public Key Hardware Accelerator (PKHA)
- SNOW 3G Accelerator
- Example, implementing IPSec

Renseignements pratiques

Durée : 6 jours
Prix : 2700 € HT