FCC3 - e200z7 implementation

This course covers the e200z7 core present in NXP MPC56XX MCUs

Objectives

- This course has 5 main objectives:
  - Learning the exception mechanism, providing guidelines to implement nesting
  - Explaining the operation and initialization of the MMU and caches
  - Highlighting the cache coherency issues and explaining the snooping
  - Detailing low level programming, particularly the floating-point and SPE instructions
  - Describing the debug units.
- ACSYS has developed an optimized SPE based FFT coded in assembler language.
- For any information contact training@ac6-training.com

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

Prerequisites

- Experience of a 32-bit processor or DSP is mandatory.

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

CORE ARCHITECTURE

- e200 core family
- Main blocks, pipeline, MMU, cache, timers, debug unit

INSTRUCTION PIPELINE

- Prefetch queue
- Decode / dispatch stage
- Concurrent Instruction Issue Capabilities
- In order execution
- Completion, register write-back
- Dynamic vs static branch prediction
- Guarded memory

SUPERVISOR PROGRAMMING, EXCEPTION MECHANISM

- Building the exception vector table
- Exception taking sequence
- Implementing nesting among maskable interrupts
- Reset sequence

INSTRUCTION AND DATA PATH

- Studying cache reload transients
- Line-fill buffers
- Memory synchronization
- Spin-lock routine

MEMORY MANAGEMENT UNIT

- Assigning attributes to pages
- Assigning access permissions to page
- Page protection
- MMU-related exceptions
- 64-entry, fully associative TLB
- TLB software reload, using MAS registers

LEVEL ONE CACHES

- 4 way set-associative Harvard instruction and data caches
- Data and instruction prefetch instructions
- Cache software control, cache line lock
- Coherency issues when cacheable pages are shared with DMA
- Cache parity and EDC protection
- Cache memory access via software

PARALLEL SIGNATURE UNIT

- System integrity checking
- Monitoring the internal CPU read and write buses

USER-LEVEL PROGRAMMING

- EABI, small sections
- Tricky instructions
- C coding guidelines

SIGNAL PROCESSING ENGINE

- Half-precision floating-point format
- Floating point simple precision & double precision scalar instructions
- Floating point vector instructions
- Fixed point vector instructions, fractional format
- Vector data arrangement instructions
- Managing a circular buffer

VARIABLE LENGTH ENCODING

- VLE storage addressing
- MMU extensions
- Summary of instruction set

CORE TIMERS

- 64-bit time base
- Decrementer
- Software watchdog

DEBUG

- Performance monitor
- Nexus Class 3+ real-time development unit
- Hardware instruction and data breakpoints
- Debug interrupt
- Debug notify halt instruction
- Using debug data acquisition message
- Watchpoint programming
- Instruction and data trace

POWER MANAGEMENT

- Power-saving modes: doze, nap, sleep, and wait
- Debug considerations for power management