



## FM2 - MPC55XX implementation

*This course covers MPC5554 and MPC5567 NXP MCUs*

### Objectives

- The course explains how to design a MPC5554 board.
- The e200 core is studied in detail, especially the MMU, the cache and the SPE instruction set.
- The course explains how to develop a generic interrupt handler.
- The training highlights data paths between core and peripherals through the internal crossbar switch.
- The host programming of eTPU and eMIOS is viewed in details.
- This course has been delivered several times to companies developing automotive and avionics systems.

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

- 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](mailto: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:
  - FlexRay, reference [IA2 - FlexRay 2.1](#) course
  - CAN bus, reference [IA1 - CAN bus](#) course
  - Ethernet, reference [N1 - Ethernet and switching](#) course
  - eTPU, reference [FM3 - eTPU programming](#) course

### 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.

## Course Outline

### MPC555X OVERVIEW

#### Block diagram

- Internal architecture of the MPC55XX
- Functional pin multiplexing
- Memory map, internal register space
- Connection of peripherals to the core platform

# e200 CORE

## CORE ARCHITECTURE

- Differences between the new Book E architecture and the classic PowerPC architecture
- The instruction pipeline
- Integer and floating point execution units
- SPE instruction set, signal processing capability, new data types
- Vector and scalar floating point
- The MMU, 32-entry fully associative TLB, page size selection
- Hardware assist for TLB miss exception
- Page attributes WIMGE
- Process protection, variable number of PID registers and sharing
- TLB initialization
- The 32-kB unified L1 cache, pseudo round-robin replacement algorithm, 8-way set associativity
- 8-entry store buffer
- Cache-related instructions
- ABI : sections
- Book E exception handling
- Core timers

## CORE DEBUG

- Nexus emulation
- Watchpoint logic

## PLATFORM

### THE INTERRUPT CONTROLLER

- Up to 504 on-chip module interrupt sources
- Software vs hardware vector mode
- Hardware acceleration for ISRs : use of 9-bit vectors
- Preemption, priority management
- External IRQs

### HARDWARE IMPLEMENTATION

- FMPLL
- Configuration pins
- Reset configuration halfword
- Boot assist module, 4 different boot modes
- MMU configuration after BAM executes
- Initialization sequence
- External bus interface, pinout
- Memory controller with support for SDR flash and SRAM
- Compatibility with the external bus of the MPC5XX
- Support for external master accesses to internal addresses
- Burst support
- Chip-select programming

### ON-CHIP MEMORIES

- 2 MB on-chip flash
- Integrated ECC
- Censorship protection

- Read while write operation
- Erase and program sequences
- 111 kB on-chip SRAM : general purpose SRAM, cache and eTPU RAMs

## **eDMA AND CROSSBAR**

- Autonomous IO control
- Parallel memory bus architecture, concurrent accesses
- Programmable master priorities on a per-slave basis
- 64 independent channels with link capability
- Parking on slave ports
- Transfer control descriptors, inner and outer loops, modulo feature
- Scatter / gather feature
- DMA channel arbitration
- DMA error reporting

## **PERIPHERALS**

### **eTPUs**

- Real time hardware events processing, scheduling, priority scheme
- Microengine operation
- New arithmetic, logical and control instructions
- Angle clock hardware
- DMA support
- Dual eTPU shared resources
- Introduction to the eTPU functions QOM, NITC, PWM, SIOP, UART
- Channel service max latency time calculation
- eTPU development tools, Ashware debugger

### **eMIOS**

- Introduction to time functions supported by the 24 unified channels
- DMA request per channel
- Pin serialization / deserialization
- eMIOS interrupt requests
- Double action submodules
- PWM submodules, center aligned PWM
- Windowed programmable time accumulation
- Quadrature decode

### **eQADC**

- Analog inputs multiplexing
- 12-bit AD resolution
- Queue management, trigger sources
- Conversion queue priority scheme
- Conversion cycle times
- eQADC command / data flow
- Hardware interface
- ADC error correction

### **DSPI**

- SPI protocol explanation, master / slave operation
- Command queue
- Flexible programming transfer attributes on a per-frame basis

- Transmit and receive sequences

## **eSCI**

- UART basics
- Double buffering
- Wake up mode
- Transmit and receive sequences
- Support for LIN master operation

## **FlexCAN controllers**

- CAN protocol basics
- Message buffer structure
- Mask registers
- Listen-only mode capability
- Receive and Transmit processes
- Error counters

## **THE FAST ETHERNET CONTROLLER**

- Overview
- MII pinout
- Buffer descriptor description
- Initialization sequence
- Error management
- Interrupts

## **FLEXRAY CONTROLLER**

- FLEXRAY protocol basics
- FLEXRAY controller characteristic
- Message buffer structure
- Clock synchronisation mechanism
- Initialization
- Error management
- Interrupts