# PG3 - ColdFire System Design

*This course has been designed for developers involved in a ColdFire development who want to understand generic mechanisms*

## Objectives

- This course explains the objectives of mechanisms used to boost the performance and the way they are implemented in various ColdFires: cache / cache coherency, pipeline, MMU, exceptions.
- This gives to the attendees a wider overview of the state of the art in these domains.
- The course details the instructions required to write program in supervisor mode to adapt the behaviour of the core to specific needs.
- A tutorial is used to quickly understand PowerPC low level programming.
- It clarifies the use of sections required for good management of caches and memory
- Task switch requirements are highlighted.
- Debug facilities implemented in ColdFires (hardware breakpoints, real-time trace, watchpoints) are studied through the use of Metrowerks debugger.

A lot of programming examples have been developed by ACSYS to explain the ColdFire assembly language.
• They have been developed with Metrowerks compiler and are executed under CodeWarrior debugger.
A more detailed course description is available on request at formation@ac6-formation.com

## Prerequisites

- A basic understanding of processor / DSP is recommended.

## 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

## ColdFire PROGRAMMING

- ColdFire core versions : V2, V2E, V3, V4, V4E
- Register set, data, address and control registers
- Data type instantiation for ColdFire
- Pointers management (Addressing modes)
- User and supervisor functions call and return (EABI, C-to-assembly interface)
- Sections, benefits of small data sections
- Locating code and data in memory , linker command file
- Reset, what is to be done before calling the main() : Cstart program

## PIPELINE

- Explaining the difference between V2, V3 and V4 pipelines
- Mechanisms used to boost performance : branch prediction, branch target address cache
- Guidelines to optimize execution time
- Serializations, nop instruction, determining when this instruction is really required

## DATA PATH

- Highlighting the frequency domains present in ColdFires : core and bus interface
- Decoupling the core from cache and bus through load and store buffers
- Enforcing the completion of commited store transactions through nop instruction
- Consequence for high level development of IO drivers
- How to make bus errors recoverable

## MEMORY MANAGEMENT UNIT

- Requirements for kernels enabling dynamic memory mapping
- Single process multi-thread versus multiprocess multi-thread kernels
- Objectives of the MMU : page protection, definition of page attribute, address translation
- Page translation
- Table search mechanisms : benefits of a software table search
- Operation of TLB caches
- TLB programming, static initialization

## CACHE AND DATA COHERENCY

- Introduction to cache memory
- Cache organization
- Write policies
- Replacement algorithms, LRU, PLRU, FIFO
- Data flow between external main memory, L1 and load / store unit
- Distinguishing private memory that is accessed only by the core and shared memory that can be accessed by the core and other masters (DMA or CPU)
- Software enforced coherency

## EXCEPTION MECHANISM

- Software exceptions vs interrupts
- Format of the exception stack frame
- Vector table operation
- Development of basic functions that get or set a vector
- Requirements for interrupt nesting

## MULTITASK

- Management of boolean semaphores
- Stack switch
- Definition of the set of registers that determine the stack state
- Management of task lists

## ColdFire DEBUG SOLUTIONS

- On-chip debug logic
- How it communicates with the debug station : BDM connection
- Hardware breakpoints
- Real-time trace

- Debugging software when caches are active