AT91SAM7 IMPLEMENTATION

Ref : 004554A **Duration : 4 days**

OBJECTIVES

- The course details the hardware implementation of the AT91SAM7 microcontrollers
- The boot sequence and the clocking are explained
- Practical lab on integrated peripherals are based on I/O functions provided by Atmel
- The course focuses on the low level programming of the ARM7TDMI core
- The course provides examples of internal peripheral software drivers

RELATED COURSES

- USB training (Ref.002606A)
- CAN training (Ref.002601A)
- ARM-7 / ARM-9 System Design (Ref.002879A)
- ARM Software development using RealView (Ref.002580A)

PARTNERS

- This training is referenced by ATMEL

PREREQUISITES

- A basic understanding of microprocessors and microcontrollers is recommended
- A basic understanding of digital logic or hardware / ASIC design issues would be useful but not essential
- A basic understanding of assembler or C programming would be useful but not essential

PRACTICAL LABS

- For on-site courses, labs can be run under the following environments : Keil µVision, or IAR Workbench
- For open courses, labs are run under IAR Workbench

Contact

Tel : +33 (0)5 62 13 52 32
Fax : +33 (0)5 61 06 72 60
training@mvd-fpga.com

Course also available customized

Next sessions, see : *http://www.mvd-fpga.com/en/formationsCalend.html*

TOPICS

INTRODUCTION TO AT91SAM7 [2-hour]

- ARM core based architecture
- APB internal busses
- The main three blocks : platform, core and input / output peripherals

THE ARM7TDMI CORE [12-hour]

- Presentation of the core, architecture and programming model
- Operating modes : user, system, super, IRQ, FIQ, undef and abort
- Pipeline, calculation of the CPI
- ALU data path
- ARM vs Thumb instruction sets, interworking
- Access to memory-mapped locations, addressing modes
- Stack management
- Branch instructions, implementation of C call and return statements
- Benefits of condition set capability in ARM state
- C-to-Assembly interface
- Exception mechanism, handler table
- JTAG interface
- Debug facilities, 2-wire UART and support for debug communication channel interrupt
- Practical lab : from a basic program, extensions have to be added by the attendees in order to implement all interesting features of the ARM7TDMI core

INFRASTRUCTURE [6-hour]

- Power supplies, internal regulator
- Power-on sequence
- Clock generator, on-chip oscillator, PLL
- Reset controller based on Power-on reset cells and low-power factory-calibrated brownout detector
- SAM-BA default boot program
- External reset
- Memory controller, embedded flash controller, memory protection unit, abort status and misalignment detection
- Internal high-speed flash, prefetch buffer, sector lock capability, flash security bit
- Practical lab : erasing and programming the internal flash
- External Bus Interface, SDRAM controller, NAND flash controller
- Power management controller, slow clock mode, idle mode, 3 programmable external clock signals
- Advanced interrupt controller
- Individually maskable, 8-level priority, vectored interrupt sources
- External interrupt sources and fast interrupt source
- Practical lab : development of a generic interrupt handler that allows nesting
- Parallel input / output controller, programming the pin multiplexers, input change capability interrupt
- 4 high-current drive I/O lines
- Peripheral DMA controller

NON COMMUNICATION ORIENTED INPUT / OUTPUT PERIPHERALS [6-hour]

Timers

- Periodic Interval Timer
- Windowed Watchdog
- Real-time timer
- 3-channel timer / counter, double PWM generation, capture / waveform mode, up/down capability
- 16-bit PWM controller
- Practical lab : demonstration of a program that generates a PWM signal with a variable duty cycle

Analogue-to-Digital Converter

- 8-channel 10-bit ADC
- Successive approximation register
- Related pins
- Conversion sequence
- Conversion trigger
- ADC timings
- Practical lab : demonstration of a program that performs periodic analog-to-digital conversions from an analog channel

COMMUNICATION CONTROLLERS [6-hour]

2-wire interface

- I2C protocol basics
- Slave mode vs master mode
- Transmit and receive sequences

SPI

- SPI protocol basics
- Master / slave operation
- External chip-select
- Transfer sequence

USART

- Individual baud rate generators
- IrDA modulation / demodulation
- Support for Smart Card
- RS485 support
- Flow control
- Practical lab : Communication with a PC, using HyperTerminal

Synchronous Serial Controller

- Independent clock and frame sync signals for each receiver and transmitter
- I2S analog interface support
- Time Division Multiplexed support
- High speed continuous data stream capabilities

USB

- Full speed operation
- Endpoint configuration
- Practical lab : Communication with a PC, using Jungo software to initiate traffic

DOCUMENTATION

Training manuals will be given to attendees during training in print.