NXP LPC21XX/LPC22XX MICROCONTROLLERS IMPLEMENTATION

Ref : 004748A **Duration : 4 days**

OBJECTIVES

- The course details the hardware implementation of the LPC2294 microcontrollers
- The boot sequence and the clocking are explained
- The training helps to become familiar with the development environment chosen by the customer
- Practical labs on integrated peripherals are based on I/O functions provided by NXP
- The course focuses on the low level programming of the ARM7TDMI core
- The course provides examples of internal peripheral software driversThis course

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

RELATED COURSES

- USB training (Ref.002606A)
- CAN training (Ref.002601A)
- Ethernet training (Ref.003367A)
- ARM-7 / ARM-9 System Design (Ref.002879A)
- ARM Software development using RealView (Ref.002580A)

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 LPC2210 AND LPC2294 [1-hour]

- ARM core based architecture
- ARM7 local bus
- AMBA AHB/APB internal buses
- The main three blocks : platform, core and input / output peripherals
- APB Bridges
- Memory mapping, internal flash (2294) and SRAM

ARCHITECTURE OF THE ARM7TDMI CORE [3-hour]

- Presentation of the core, architecture and programming model
- Operating modes : user, system, super, IRQ, FIQ, undef and abort
- Pipeline, calculation of the CPI
- Effects of branches and exceptions on the performance
- ALU data path

SOFTWARE IMPLEMENTATION, V4T SPECIFICATION [8-hour]

- Parameterizing the RVDS linker to define sections
- Branch instructions, implementation of C call and return statements, long branch veneers
- ARM vs Thumb instruction sets, interworking
- Programming example : Highlighting the usage of veneers to perform a transition during the function call (Thumb-to-ARM or ARM-to-Thumb interworking)
- ARM instruction set
- Inline barrel shifter
- Access to memory-mapped locations, addressing modes
- Arithmetical and logic instructions
- Thumb instruction set, highlighting restrictions with regard to ARM instruction set
    - Compiler hints and tips, optimisations supported by RVCT
- Stack management
- Benefits of condition set capability in ARM state
- C-to-Assembly interface, ATPCS specification
- Programming example : Development of an assembly function that is called from a C function
- Programming example : Explaining the steps from reset to main

EXCEPTION MECHANISM [4-hour]

- Reset
- FIQ vs IRQ
- Exception return instructions
- Programming example : Implementing the SWI exception
- Latency estimation, impact of load and store multiple instructions
- Organization of the handler table, priority decoder, pre-emption and nesting
- ISR header and footer routines
- Development of a generic exception handler
- Programming example : Implementing a task scheduler that saves the state of a task when an interrupt event occurs

INTEGRATED DEBUG FACILITIES [2-hour]

- JTAG interface
- Debug facilities, hardware breakpoint
- Executing code from RAM to take benefit of software breakpoints

THE VECTORED INTERRUPT CONTROLLER [2-hour]

- Assigning a priority to each interrupt source
- Steering external interrupts and local interrupts to either the core FIQ or IRQ
- Developing a generic interrupt handler performing nesting according to peripheral priorities defined by the user
- Integrated timers
- Using timers to understand the operation of the VIC
- Programming example : Implementing nesting by using 2 timers having a different priority level

SYSTEM CONTROL [2-hour]

- Pin connect block
- Clocking
- Reset and wake-up timer
- Low power modes
- Watchdog timer
- Real-Time clock

ON-CHIP FLASH MEMORY (2294) [2-hour]

- Organization
- Erase sequence
- Program sequence
- In system programming via serial port
- On-chip bootloader

EXTERNAL MEMORY CONTROLLER [2-hour]

- Address decoding
- Chip-select registers
- Parameterizing the memory bank registers to support external burst flash

SERIAL INTERFACES [6-hour]

- I2C basics
- I2C controller
- UART controller
- SPI and SSP interfaces
- CAN protocol basics
- CAN controller (2294)
- Programming example : demonstration of drivers developed by MVD will be done at the end of the course (they are delivered in source format to attendees)

DOCUMENTATION

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