

CORTEX-A9MP SOFTWARE IMPLEMENTATION

Training Duration: 3 days

Prerequisites:

- Knowledge of ARM7/9.
- This course does not include chapters on low level programming.
- ACSYS offers a large set of tutorials to become familiar with RVDS, assembly level programming, compiler hints and tips.
- More than 12 correct answers to Cortex-A prerequisites questionnaire.

Related courses:

- Programming with RVDS IDE (reference R13)
- VFP programming (reference RC0)
- NEON programming (reference RC1)

Practical labs:

- Labs are run under RVDS 4.1

Course objectives:

- This course aims to explain all low level characteristics of the Cortex-A9 that are required to develop efficient Kernel or application code.
- MMU operation under Linux is described.
- Spin-lock implementation in a multicore system is also detailed
- Interaction between level 1 caches, level 2 cache and main memory is studied through sequences.
- The exception mechanism is explained, indicating how virtualization enables the support of several operating systems.
- An overview of the Coresight specification is provided prior to describing the debug related units.
- The operation of the Snoop Control Unit when supporting SMP is fully explained, particularly the utilization of cache tag mirrors, the advantage of connecting DMA channels to ACP and the sequences that have to be used to modify a page descriptor.

Documentation

Training manuals will be given to attendees during training. Precise and easy to use, those notes can be used as a reference afterwards.

Course Content:

First day

INTRODUCTION TO CORTEX-A9 [1-hour]

- Block diagram, 1 or 2 AXI master interfaces
- Cortex-A9 variants: single core vs multicore
- New memory-mapped registers in MPCore
- The 3 instruction sets
- Configurable options: cache size, Jazelle, NEON, FPU, PTM and IEM

ARM BASICS [1-hour]

- States and modes
- Benefit of register banking
- Exception mechanism
- Instruction sets
- Purpose of CP15

INSTRUCTION PIPELINE [2-hour]

- Superscalar pipeline operation, out-of-order operation
- Instruction cycle timing
- Branch prediction mechanism, BTAC and GHB usage
- Guidelines for optimal performance
- Return stack
- Predicted and non-predicted instructions
- Prefetch queue flush
- PMU related events

TRUSTZONE [2-hour]

- TrustZone
- Secure to non secure permitted transitions

- Related CP15 registers
- L1 and L2 secure state indicators, memory partitioning
- Interrupt management when there is a mix of secure and non-secure interrupt sources
- Boot sequence
- Entering / exiting dormant mode

OS SUPPORT – SYNCHRONIZATION OVERVIEW [2-hour]

- Inter-Processor Interrupts
- Barriers
- Cluster ID
- Exclusive access monitor, implementing Boolean semaphores
- Global monitor
- Spin-lock implementation
- Using events

Second day

THUMB-2, THUMB-2EE AND ARM INSTRUCTION SETS (V7-A) [2-hour]

- Introduction
- General points on syntax
- Data processing instructions
- Branch and control flow instructions
- Memory access instructions
- Exception generating instructions
- If...then conditional blocks

Continued ...

CORTEX-A9MP SOFTWARE IMPLEMENTATION

Training Duration: 3 days

Course Content:

... Continued

- Stack in operation
- Exclusive load and store instructions
- Accessing special registers
- Coprocessor instructions
- Interworking ARM and Thumb states
- Thumb-2EE extension for supporting interpreted languages
- Using handlers to manage NULL pointers and array index that are outside a programmable range
- Demonstration of assembly sequences aimed to understand this new instruction set

MEMORY MANAGEMENT UNIT [2-hour]

- MMU objectives
- Page sizes
- Address translation
- Page access permission, domain and page protection
- Page attributes, memory types
- Utilization of memory barrier instructions
- Format of the external page descriptor table
- Tablewalk
- TLB organization
- TLB lockdown
- Utilization of microTLBs
- Abort exception, on-demand page mechanism
- MMU maintenance operations
- Using a common page descriptor table in an SMP platform, maintaining coherency of multiple TLBs
- PMU related events
- Related CP15 registers

LEVEL 1 MEMORY SYSTEM [2-hour]

- Cache organization
- Virtual indexing, physical tagging for instruction cache ; physical indexing and tagging for data cache
- Supported maintenance operations
- Write-back write allocate cache allocation
- Memory hint instructions PLD, PLI, PLDW, data prefetching
- Describing transient cache related transactions: line fills and line eviction
- No lockdown support
- 4-entry 64-bit merging store buffer
- PMU related events

PL310 LEVEL 2 CACHE [2-hour]

- Cache configurability
- Exclusive mode operation when connected to Cortex-A9
- Understanding through sequences how cacheable information is copied from memory to level 1 and level 2 caches

- Transient operations, utilization of line buffers LFBs, LRBs, EBs and STBs
- Discarding a level 3 memory line load through merging writes into STBs
- TrustZone support
- Power management
- Cache event monitoring
- Memory mapped registers included in the cache controller
- Describing each maintenance operation
- Cache lockdown, implementation of a small memory by a boot program
- Initialization sequence
- Interrupt management

Third day

HARDWARE COHERENCY [1-hour]

- Snooping basics
 - Snoop requests
 - Snoop Control Unit: cache-to-cache transfers
 - MOESI state machine
 - Address filtering
 - Understanding through sequences how data coherency is maintained between L2 memory and L1 caches
 - Accelerator Coherency Port: connecting a DMA channel that uses this port to enforce coherency of data it is transmitting
 - Enabling coherency mode

PERFORMANCE MONITOR [1-hour]

- Event counting
- Selecting the event to be counted for the 6 counters
- Related interrupts
- Debugging a multi-core system with the assistance of the PMU

INTERRUPT CONTROLLER [2-hour]

- Cortex-A9 exception management: enforcing a particular endian mode on exception entry, configuring FIQ to be non maskable, configuring the default exception handling state: ARM vs Thumb
- Interrupt virtualization
- Integrated timer and watchdog unit in MPCore
- Interrupt groups: STI, PPI, SPI, LSPI
- Legacy mode: direct IRQ and FIQ
- Assigning a security level to each interrupt source (Secure or Non Secure)
- Prioritization of the interrupt sources
- Distribution of the interrupts to the Cortex-A9 cores
- Generation of interrupts by software
- Detailing the interrupt sequence, purpose of Interrupt Acknowledge register and End-Of-Interrupt register
- Clarifying which registers have a single instance and which registers are replicated to support MC

Continued ...

CORTEX-A9MP SOFTWARE IMPLEMENTATION

Training Duration: 3 days

Course Content:

... Continued

- Spurious interrupt

CORESIGHT DEBUG UNITS [2-hour]

- Benefits of CoreSight
- Invasive debug, non-invasive debug, taking into account the secure attribute
- APBv3 debug interface
- Connection to the Debug Access Port
- Debug facilities offered by Cortex-A9
- Process related breakpoint and watchpoint
- Program counter sampling
- Event catching
- Debug Communication Channel
- PTM interface, connection to funnel
- Debugging while the processor is in shutdown or dormant mode
- Debug registers description
- Miscellaneous debug signals
- Cross-Trigger Interface, debugging a multi-core SoC

COMPILER HINTS AND TIPS [2-hour]

- Placing code, data, stack and heap in the memory map, scatterloading
- Tailoring the C library to your target
- Reset and initialisation
- Placing a minimal vector table
- Further memory map considerations, 8-byte stack alignment in handlers
- Building and debugging an image
- Long branch veneers
- ARM compiler optimisations, tail-call optimization, inlining of functions
- Mixing C/C++ and assembly
- Coding with ARM compiler
- Measuring stack usage
- Unaligned accesses
- Local and global data issues, alignment of structures
- Further optimisations, linker feedback

In order to study these hints and tips, ACSYS has prepared a tutorial developed with RVDS4.1 HARDWARE COHERENCY [1-hour]

- Snooping basics
- Snoop requests
- Snoop Control Unit: cache-to-cache transfers
- MOESI state machine
- Address filtering
- Understanding through sequences how data coherency is maintained between L2 memory and L1 caches
- Accelerator Coherency Port: connecting a DMA channel that uses this port to enforce coherency of data it is transmitting
- Enabling coherency mode

PERFORMANCE MONITOR [1-hour]

- Event counting
- Selecting the event to be counted for the 6 counters
- Related interrupts
- Debugging a multi-core system with the assistance of the PMU

INTERRUPT CONTROLLER [2-hour]

- Cortex-A9 exception management: enforcing a particular endian mode on exception entry, configuring FIQ to be non maskable, configuring the default exception handling state: ARM vs Thumb
- Interrupt virtualization
- Integrated timer and watchdog unit in MPCore
- Interrupt groups: STI, PPI, SPI, LSPI
- Legacy mode: direct IRQ and FIQ
- Assigning a security level to each interrupt source (Secure or Non Secure)
- Prioritization of the interrupt sources
- Distribution of the interrupts to the Cortex-A9 cores
- Generation of interrupts by software
- Detailing the interrupt sequence, purpose of Interrupt Acknowledge register and End-Of-Interrupt register
- Clarifying which registers have a single instance and which registers are replicated to support MC
- Spurious interrupt

CORESIGHT DEBUG UNITS [2-hour]

- Benefits of CoreSight
- Invasive debug, non-invasive debug, taking into account the secure attribute
- APBv3 debug interface
- Connection to the Debug Access Port
- Debug facilities offered by Cortex-A9
- Process related breakpoint and watchpoint
- Program counter sampling
- Event catching
- Debug Communication Channel
- PTM interface, connection to funnel
- Debugging while the processor is in shutdown or dormant mode
- Debug registers description
- Miscellaneous debug signals
- Cross-Trigger Interface, debugging a multi-core SoC

COMPILER HINTS AND TIPS [2-hour]

- Placing code, data, stack and heap in the memory map, scatterloading
- Tailoring the C library to your target
- Reset and initialisation
- Placing a minimal vector table

Continued ...

CORTEX-A9MP SOFTWARE IMPLEMENTATION

Training Duration: 3 days

Course Content:

... *Continued*

- Further memory map considerations, 8-byte stack alignment in handlers
 - Building and debugging an image
 - Long branch veneers
 - ARM compiler optimisations, tail-call optimization, inlining of functions
 - Mixing C/C++ and assembly
 - Coding with ARM compiler
 - Measuring stack usage
 - Unaligned accesses
 - Local and global data issues, alignment of structures
 - Further optimisations, linker feedback
 - In order to study these hints and tips, ACSYS has prepared a tutorial developed with RVDS4.1
-