ARM Cortex A5 - Conception système

4 jours

OBJECTIFS

- Cette formation traite en détails les particularités des cœurs ARM, aussi bien logiciel que matériel dans le but de facilité la mise en oeuvre de cœurs Cortex-A5.
- Elle est destiné aux :
  - Ingénieurs logiciel qui veulent non seulement obtenir des détails sur la façon d'écrire un logiciel pour processeur ARM Cortex-A5, mais qui souhaitent également comprendre l'implémentation matériel des cœurs au sein d'un microcontrôleur
  - Ingénieurs matériel qui ont besoin de comprendre comment concevoir des systèmes basés sur ARM Cortex-A5 mais également être capable de comprendre les bases de la programmation logicielle sur ces plates-formes

FORMATIONS CONNEXES

- ARM1176 - Conception système
- ARM Cortex A8 - Conception système
- ARM Cortex A9/A9MP - Conception système

PARTENAIRES

PRÉREQUIS

- Une compréhension de base des microprocesseurs et microcontrôleurs est utile mais non indispensable
- Une compréhension de base de la logique numérique est utile mais non indispensable
- Une compréhension de base de la programmation en assembleur ou en langage C est utile mais non indispensable
- Des notions sur les cœurs ARM sont utiles mais non indispensables

CONFIGURATIONS

- Pour les formations sur site, les travaux pratiques peuvent être effectués sous les environnements suivants : Keil DS-5, Keil µVision, ou IAR Workbench

CHAPITRES

INTRODUCTION TO CORTEX-A9

- Block diagram
- ARMv7-A architecture
- Operating modes
- ARM instruction set
- Thumb-2 instruction set
- Thumb-2EE instruction set, replacement of Jazelle
- Program Status register
- Exceptions
- System control coprocessor
- Configurable options

THUMB-2 INSTRUCTION SET

- General points on syntax
- Data processing instructions
- Branch and control flow instructions

- Memory access instructions
- Exception generating instructions
- If...then conditional blocks
- Stack in operation
- Exclusive load and store instructions
- Accessing special registers
- Coprocessor instructions
- Memory barriers and synchronization
- Interworking ARM and Thumb states

INSTRUCTION PIPELINE

- Superscalar pipeline operation
- Studying how instructions are processed step by step
- Instruction cycle timing
- Branch prediction mechanism, BTB and GHB usage
- Guidelines for optimal performance
- Return stack
- Instruction Memory Barrier
- Prefetch queue flush

NEON TECHNOLOGY

- Overview of the NEON media coprocessor
- 10-stage NEON pipeline, processing pipelines, load/store pipeline
- Data types
- NEON instruction set
- VFPv3 architecture
- General purpose registers
- NEON vectorizing compiler support
- NEON coding examples

UNALIGNED DATA AND MIXED-ENDIAN DATA SUPPORT

- Understanding how unaligned word transfers are handled
- Setting the endian mode for data transfers through CPSR[E]
- Understanding how the bus interface unit re-orders bytes when a big endian transfer is performed

MEMORY MANAGEMENT & PROTECTION

- Introduction to page management
- V7 virtual memory architecture, 16-MB supersection support
- Understanding protection domains
- TLB reload mechanism
- TLB lockdown
- Page table in trusted and untrusted worlds
- TLB software read for debug purposes
- Abort exception management, syndrome registers
- Imprecise aborts

TRUSTZONE

- TrustZone conceptual view
- Secure to non secure permitted transitions
- Related CP15 registers
- L1 and L2 secure state indicators, memory partitioning

HARDWARE IMPLEMENTATION

- Clock domains, CLK, PCLK, ATCLK
- Using clock enable to determine the ratio between input clock and operation clock
- Reset domains, power-on reset, debug and ETM reset
- Power control, dynamic power management
- Wait For Interrupt architecture
- Debugging the processor while powered down

LEVEL 1 CACHE

- Cache organization
- Virtual indexing, physical tagging
- Hash Virtual Address Buffer
- Hardware support for virtual aliasing conditions
- Parity protection
- Write buffer
- L1 caches software read for debug purposes

LEVEL 2 CACHE

- Cache organization
- Physical indexing, physical tagging
- L2 cache transfer policy
- Parity / ECC protection
- Write buffer
- L2 Preload Engine [PLE]
- START, STOP and CLEAR commands
- L2 cache software read for debug purposes

MPCORE FEATURES

- Snoop Control Unit
- Interrupt Controller
- Timer and Watchdog
- TrustZone support

DEVELOPING FOR ARM MPCORE PROCESSORS

- Booting SMP
- Configuring an interrupt
- Synchronization
- Accelerator Coherency Port

CORESIGHT DEBUG UNIT

- Coresight specification overview
- CP14 and memory-mapped registers, utilization of an APB slave interface
- APB port access permissions
- Embedded core debug
- Invasive debug : breakpoints and watchpoints
- Vector catch
- Debug exception
- External debug interface
- Understanding how the Debug unit, the Embedded Trace Macrocell and the Cross-Triggering Interface interact

CORESIGHT EMBEDDED TRACE MACROCELL

- Overview

- Exporting the compressed trace information
- Benefits of an Embedded Trace Buffer
- Defining trace trigger conditions
- Context ID tracing
- Instrumentation instructions

THE PERFORMANCE MONITOR UNIT

- Event counting principle
- Configuring the 4 event counters
- Event selection

AXI PROTOCOL

- Topology : direct connection, multi-master, multi-layer
- PL300 AXI interconnect
- Separate address/control and data phases
- AXI channels, channel handshake
- Transaction ordering, out of order transaction completion
- Read and write burst timing diagrams
- Cortex-A8 external memory interface, ID encoding
- Exclusive ressource management

APB

- Address decoding stages
- APB interconnect
- APB in AMBA3

NOTES

- Les supports de cours seront fournis sur papier à chaque participant pendant la formation

CONTACT

Tel : 05 62 13 52 32
Fax : 05 61 06 72 60
training@mvd-training.com