

## ARM-7 / ARM-9 Conception Système

Ref : 002879A

Durée : 4 jours

### OBJECTIFS

- Cette formation traite en détails les particularités des cœurs ARM, aussi bien logiciel que matériel dans le but de faciliter la mise en oeuvre de cœurs de famille ARM7TDMI et ARM9TDMI.
- Elle est destinée aux :
  - Ingénieurs logiciel qui veulent non seulement obtenir des détails sur la façon d'écrire un logiciel pour processeur ARM, 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 mais également être capable de comprendre les bases de la programmation logicielle sur ces plates-formes

### FORMATIONS CONNEXES

- ARM7/9 Conception hardware (Ref.002579A)
- ARM Développement software avec RealView (Ref.002580A)

### PARTENAIRES

- Cette formation est approuvée par ARM

### PRE-REQUIS

- 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

### TRAVAUX PRATIQUES

- Pour les formations sur site, les travaux pratiques peuvent être effectués sous les environnements suivants : CodeWarrior/ADS/AXD, Eclipse/RVDS, Keil  $\mu$ Vision, GNU/Lauterbach simulator, or IAR Workbench
- Pour les formations inter-entreprises, les travaux pratiques se font sous Eclipse/RVDS



### Contact

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

Le contenu peut-être adapté sur site

Prochaines sessions, voir ici : <http://www.mvd-fpga.com/fr/formationsCalend.html>

### PROGRAMME

#### 1<sup>er</sup> jour

#### L'ARCHITECTURE ARM

- Vue générale d'ARM
- Modes de fonctionnement ARM
- L'ensemble des registres ARM, Program Status Registers
- Gestion d'exception, table des vecteurs, commutation automatique en mode ARM
- Ensembles d'instructions : branchement et sous-routines

#### CŒUR ARM

- Signaux cœur ARM7TDMI
- Diagramme block ARM7TDMI
- Le pipeline d'instruction du ARM7TDMI
- Interface mémoire ARM7TDMI
- Chemins de données ARM9TDMI
- Pipeline ARM9TDMI
- Vue générale de ARM9E-S, ARM10, StrongARM et Xscale

#### INTRODUCTION A REALVIEW DEVELOPPER SUITE (RVDS)

- Utilisation des outils « cœur »
- Caractéristiques clés des compilateurs C/C++
- Bibliothèques fournies
- Introduction à Codewarrior
- Débogage avec multi-ICE

#### TPs RVDS

- Compilation et lancement d'un exemple

- Création d'un fichier entête
- Création d'un nouveau projet
- Visualisation de la mémoire et des registres

#### 2<sup>ème</sup> jour

#### JEUX D'INSTRUCTIONS ARM ET THUMB

- Exécution conditionnelles et drapeaux
- Instructions de branchement
- Registre à décalage
- Constantes immédiates
- Transfert de registre simple
- Transfert de blocks de registres
- Gestion de la pile
- Instructions coprocesseur
- Accès aux registres en mode Thumb
- Nouvelle instructions de l'architecture v5TE
- TPs langage assembleur

#### ARM / THUMB INTERWORKING

- Commutation entre états
- Exemple de Branch & Exchange
- Inter-opérabilité ARM/Thumb
- Veneer ARM vers Thumb
- Veneer Thumb vers ARM

#### GESTION D'EXCEPTION

- Instructions de sortie d'exception
- Priorité des exception
- Instructions de la table des vecteurs
- Enchaînement d'exceptions
- Utilisation des registres dans un gestionnaire d'exception

- FIQ vs IRQ
- Exemple de gestionnaire d'interruption en C
- Contrôleur d'interruption
- Problème de démasquage d'interruption
- Imbrication d'interruption
- Utilisation du SWI
- Data Abort et gestion de la mémoire
- L'adresse de retour

### 3<sup>ème</sup> jour

#### INITIALISATION DE PROCESSEURS

- Les bases du cache, associativité, verrouillage de cache
- Le model programmeur
- Vidage du cache
- Tampon d'écriture, stratégie d'écriture
- Gestion de la mémoire, translation d'adresse virtuelles vers physiques
- TLB et tables de translation, descripteurs niveau 1 et niveau 2
- Génération d'adresse avec le registre ID
- Protection mémoire, configuration de la MPU
- Coprocesseur de control système
- Exemple de code d'initialisation
- Tightly Coupled Memory

#### CONSEILS DE CODAGE

- Optimisation automatique
- Re-ordonnancement des instructions
- Tail-call optimization
- Passage de paramètre
- Access à des tableaux ou structures
- Fin de boucle
- Opération de division
- Assembleur embarqué
- Utilisation de la pile
- Gestion des variables globales

#### DEVELOPPEMENT DE LOGICIEL EMBARQUE

- ROM ou RAM à 0x0 ?
- ROM/RAM remapping
- Table des vecteurs d'exceptions
- Gestionnaire de Reset

- Initialisation : pointers de pile, zone code et données
- Initialisation de la librairie C
- Fichier de configuration du linker
- Long branch veneers
- Fonctionnalités de la librairie C
- Positionnement de la pile et du tas
- Débogage d'images ROM

### 4<sup>ème</sup> jour

#### SPECIFICATION D'INTERCONNEXION AMBA 3.0

- Intérêt de la spécification
- Exemple typique de système basé sur l'interconnexion AMBA
- Différences entre AMBA 2.0 et AMBA 3.0

#### AHB – ADVANCED HIGH PERFORMANCE BUS

- Décodage d'adresse centralise
- Address gating logic
- Arbitrage de bus, parage de bus
- Transactions data simple
- Transferts séquentiels
- Réponse de type retry
- Réponse de type split
- Spécification AHB-Lite

#### APB – ADANCED PERIPHERAL BUS

- Chronogramme de lecture
- Chronogramme d'écriture
- Interconnexion AHB vers APB
- Nouvelles fonctionnalités de l'APB 3.0

#### SOLUTIONS DE DEBOGAGE ARM

- Debogage avec multiICE
- Watchpoints, points d'arrêt matériels et logiciels
- Debug Communication Channel
- Semihosting
- EmbeddedICE-RTT logic
- Traçage temps-réel
- Trace instruction et trace data
- Capture de trace

## DOCUMENTATION

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