

ARM7 ARM9 CONCEPTION SYSTEME

4 jours

Réf. : 5184

OBJECTIFS

Cette formation traite en détail les particularités des cœurs ARM, aussi bien logicielles que matérielles dans le but de faciliter la mise en œuvre de cœurs des familles 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 matérielle 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.

Les travaux pratiques du stage ARM-7 / ARM-9 Conception Système sont réalisés :

- Pour un stage intra-entreprise sur site, les travaux pratiques pourront être réalisés sur : Eclipse/RVDS, ARM Development Studio 5 (DS-5), Keil μ Vision, GNU/Lauterbach simulator, ou IAR Workbench.
- Pour les stages interentreprises, les travaux pratiques seront assurés sur l'environnement ARM Development Studio 5 (DS-5).

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.

PROGRAMME

PREMIÈRE JOURNÉE

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

Coeur 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 à RealView Developer Suite (RVDS)

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

Travaux pratiques RVDS

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

DEUXIÈME JOURNÉE

Jeu d'instructions ARM et THUMB

- Exécutions conditionnelles et drapeaux
- Instructions de branchement
- Registre à décalage
- Constantes immédiates
- Transfert de registres simples
- Transfert de blocs de registres
- Gestion de la pile
- Instructions coprocesseur
- Accès aux registres en mode Thumb
- Nouvelles instructions de l'architecture v5TE
- TP 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'exceptions

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

Gestion d'interruptions

- FIQ vs IRQ
- Exemple de gestionnaire d'interruptions 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

TROISIÈME JOURNÉE

Initialisation du processeur ARM

- 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'adresses 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
- Ré ordonnancement des instructions
- Tail-call optimisation
- Passage de paramètres
- Accès à des tableaux ou structures
- Fin de boucle
- Opération de division
- Assembleur embarqué
- Utilisation de la pile

Développement de logiciel embarqué

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

QUATRIÈME JOURNÉE

Spécification d'interconnexion AMBA 3.0

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

AHB – ADVANCED HIGH PERFORMANCE BUS

- Décodage d'adresses centralisées
- Address gating logic
- Arbitrage de bus, partage de bus
- Transactions data simple
- Transferts séquentiels
- Réponse de type retry
- Réponse de type split
- Spécification AHB-Lite

APB – ADVANCED PERIPHERAL BUS

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

Solutions de débogage ARM

- Débogage 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