

# ARM CORTEX R4

4 jours

Réf. : 5191

## OBJECTIFS

Cette formation, assurée par des formateurs habilités ARM™, aborde la problématique de la conception d'un système contenant un cœur processeur ARM Cortex-R4.

Elle permet :

- d'identifier les différents blocs qui composent le processeur ARM cortex R4,
- d'examiner en détail le fonctionnement du cœur, des différents bus et interfaces,
- d'utiliser les différentes fonctions du microcontrôleur avec des travaux pratiques sur CodeWarrior/ADS/AXD, Eclipse/RVDS, Keil μVision, simulateur GNU/Lauterbach ou IAR Workbench,
- de décrire précisément les échanges de données entre le cœur système et les périphériques.

## PRÉ-REQUIS

Des notions de base en électronique numérique, conception ASIC sont souhaitables.

Une connaissance générale de l'ARM et en particulier la connaissance du jeu d'instructions ARM V4T et V5TE, est vivement recommandée.

## PROGRAMME

### PREMIÈRE JOURNÉE

#### Introduction ARM Cortex-R4

- Les différents blocs fonctionnels
- Les nouveautés du Cortex-R4 par rapport aux autres générations de cœurs ARM
- Architecture de l'ARMv7-R
- Différents modes opérationnels
- Jeux d'instructions ARM
- Jeux d'instructions Thumb-2
- Registre d'Etat
- Les exceptions
- Système de contrôle du coprocesseur
- Options configurables

#### Jeu d'instructions THUMB-2

- Introduction
- Généralité concernant la syntaxe
- Instructions de traitement des données
- Instructions de branchement et de contrôle de flux
- Instructions d'accès à la mémoire
- Instructions liées aux exceptions
- Blocs conditionnels (If ... then ...)
- Opérations sur l'accumulateur
- Instructions exclusives : load et store
- Accès aux registres spéciaux
- Instructions du coprocesseur
- Synchronisation mémoire et contrôle d'accès
- Barrière mémoire et synchronisation
- Fonctionnement ARM et Thumb combinés
- Travaux pratiques : compilation et assemblage du nouveau jeu d'instructions

#### Unité de calcul en virgule flottante VFPv3

- Encodage des nombres à virgule flottante (normalized, tiny, zero, infinite, NaN)
- Architecture du VFPv3-D16
- Généralités et détails des registres de la FPU
- Conformité avec le standard IEEE754
- Gestion des exceptions
- Gestion des opérations arithmétiques invalides «NaN»
- Démonstration d'un calcul à virgule flottante à partir d'un exemple codé en langage C décomposé en instruction VFPv3

#### Trucs et astuces du compilateur C pour Cortex-R4

- Optimisation automatique
- Séquençement des instructions
- Optimisation de la récursivité «Tail-call»
- Passage des paramètres
- Accès aux données des tableaux et des structures
- Fin de boucle
- Codage assembleur
- Usage de la pile
- Disposition globale des données
- Travaux pratiques concernant l'optimisation «Tail-call»

### DEUXIÈME JOURNÉE

#### Développement logiciel embarqué

- Gestion de l'adressage ROM/RAM
- Table des vecteurs d'exceptions
- Gestionnaire de reset
- Initialisation : pointeur de pile, code et zone de données

- Initialisation de la librairie C
- Mécanisme de «Scatterloading»
- Règles de plan d'adressage du «Linker»
- Gestion des «branchements longs»
- Fonctions de la librairie C
- Placement de la pile et du tas (stack and heap)

#### Pipeline d'instructions Cortex-R4

- Unité de «Prefetch»
- Traitement d'instruction pas à pas
- Cycle d'exécution d'instruction
- Mécanisme de branchement prédictif : global history buffer
- Optimisation de performances
- Unité de traitement des données
- Double conditions de sortie de traitement
- Pile de retour des sous-routines
- Instruction de barrière mémoire
- Préparation du vidage des files
- Événements relatifs au PMU

#### Types de mémoires

- Restriction par rapport au chargement et au stockage
- Types de mémoires, restriction lecture / écriture multiple
- Accès périphériques et mémoire
- Ordre d'accès et barrière mémoire

#### Unité de protection mémoire (MPU)

- Protection mémoire, ARM v7 PMSA
- État des erreurs et registre d'adresses
- Cortex-M4 MPU et erreurs de bus
- Régions, sous-régions et contrôle d'accès
- Recouvrement des zones (overlapping)
- Configuration de la MPU

#### Gestions des interruptions et exceptions

- Low Interrupt Latency : abandon d'une instruction en cours de chargement/stockage
- Configuration d'état de la gestion des exceptions : jeux d'instructions «endian mode»
- Configuration de la FIQ comme non-masquable
- Primecell Vectored Interrupt Controller (VICs)
- Réduction des temps de latence via la génération automatique des vecteurs d'interruption
- Chronogramme des signaux du VIC
- Connectivité : daisy-chained VIC
- Priorité et masque d'interruption
- Abandon lié aux exceptions et gestion des erreurs
- Comment déterminer la cause de l'erreur grâce aux registres du CP15
- Erreurs connues vs erreurs non identifiées

### TROISIÈME JOURNÉE

#### Intégration matérielle de Cortex-R4

- Les horloges CLKIN, FREECLKIN et CLKDBGK
- Choix des «prescaler» entre l'horloge d'entrée et la fréquence de travail du processeur
- Reset, power-on reset, Debug et reset ETM
- Contrôle et gestion dynamique d'énergie
- Mécanisme d'attente sur interruption
- Débogage à l'extinction du processeur

#### Protocole AXI

- Topologie : connexion directe, multimaître, multicouche
- Interconnexion AXI PL300
- Séparation des accès data, control/adresses
- Canaux AXI et handshake

- Support pour les transferts de données non alignées
- Ordonnement des transactions, exécution en mode dégradé
- Chronogramme des transactions de lecture / écriture
- Interface mémoire externe Cortex-R4
- Transactions aux niveaux atomiques

#### Cache mémoire de niveau 1 du Cortex-R4

- Généralités sur le cache : organisation, algorithmes de remplacement, règles d'écritures
- Organisation du cache
- Règles d'allocations mémoires pour l'écriture
- Débogage quand les caches sont actifs
- Protection ECC / Parité
- Etude des transitions chargement / stockage de la ligne de cache : linefill buffers, eviction buffer
- Accès à la RAM du cache via l'interface esclave de l'AXI
- Entrelacement BTM initié par une connexion entre le cœur et l'AXI DMA vers l'interface esclave de l'AXI
- Store buffer, merging data
- Débogage logiciel du cache L1
- Événement lié au PMU

### QUATRIÈME JOURNÉE

#### SYSTÈME MÉMOIRE DE NIVEAU 2

- Interface maître de l'AXI, capacité d'écriture et de lecture
- Identifiants des transactions de l'AXI
- Contrôle du cache externe
- Restrictions concernant les transferts AXI
- Détermination du nombre et des types de transactions AXI en fonction des attributs mémoires et du type d'instruction
- Séparation des transactions AXI
- Interface esclave de l'AXI, capacité d'écriture et de lecture
- Activation / désactivation des accès à l'AXI esclave
- Utilisation du chip-select pour définir les accès A-TCM, B-TCM, I-Cache et D-Cache
- Utilisation de l'interface esclave de l'AXI pour faire des auto-tests intégrés
- Comprendre le mécanisme de récupération après erreur
- Accès exclusif, instructions de swap, moniteur exclusif interne
- Moniteur externe dans le cas d'une implémentation multicœur

#### APB - BUS PÉRIPHÉRIQUES AVANCÉS

- Second niveau de décodage d'adresses
- Brochage
- Chronogramme de lecture
- Chronogramme d'écriture
- Nouvelles fonctionnalités de l'APB3.0

#### Unité de débogage

- Moniteur de performance, compteur d'événements
- Interruptions, événement sur les bus
- Généralité sur la spécification Coresight
- Registres CP14 et memory-mapped, utilisation d'une l'interface APB esclave
- Permission d'accès sur le port APB
- Code de débogage embarqué
- Débogage intrusif : breakpoints et watchpoints
- Capture des vecteurs

#### Synthèse et conclusion

© CenoSYS 2011-11