RM5 - Mise en œuvre du Cortex-M33

Ce cours couvre le cœur ARMv8 Cortex-M33

Objectifs du cours

- Ce cours se divise en 3 parties :
  - Architecture du Cortex-M33
  - Mise en œuvre logicielle et débogage du Cortex-M33
  - Programmation DSP du Cortex-M33.
- Le Cortex-M33, bien qu'étant un cœur 32 bits, est l'un des premiers cœurs ARMv8-M proposés par ARM et comporte plusieurs fonctions avancées.
- La programmation bas niveau du Cortex-M33 est expliquée, en particulier le paramétrage de l'éditeur de liens ARM et certaines instructions assembleur délicates.
- Le cours indique également comment utiliser les instructions DSP et FPU pour accélérer la mise en œuvre des algorithmes de traitement du signal.
- Les différents éléments de débogage CoreSight présents dans le Cortex-M33 sont également présentés

Prérequis

- Compréhension de base des microprocesseurs et des microcontrôleurs

Environnement du cours

- Les travaux pratiques sont exécutés sur un simulateur ARMv8.
- Un support de cours imprimé est remis aux participants pendant la formation.
- Précis et facile à utiliser, il peut servir de référence par la suite.

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

Premier jour

Introduction à l'architecture ARMv8-M

- Macrocellule du processeur ARM Cortex-M33
- Modèle du programmeur’ ARMv8-M
- Pipeline d'instructions
- Plan mémoire figé
- Privilège, modes et piles
- Memory Protection Unit
- Extensions de sécurité
- Gestion des interruptions
- Nested Vectored Interrupt Controller [NVIC]
- Gestion de l'énergie
- Débogage

Cœur ARM Cortex-M33

- Registres à usage spécifique
- Chemin de données et pipeline
- Write buffer
- Bit-banding
- Timer système
- État, privilège et piles
- System control block

Architecture d'un SoC à base de Cortex-M33

- Matrice de bus interne
- Matrice de bus externe pour gérer les maîtres DMA
- Connexion des périphériques
- Partage des ressources entre le Cortex-M4 et les autres processeurs
- Connexion au Power Manager Controller

Deuxième jour

Développement de logiciel embarqué avec le Cortex-M33

- Démarrage de l'application
- Placement du code, des données, de la pile et du tas dans le plan mémoire, scatterloading
- Reset et initialisation
- Placement d'une table de vecteurs minimale
- Autres considérations sur le plan mémoire, alignement de pile sur 8 octets dans les handlers

La variante du jeu d'instructions T32 prise en charge sur ARMv8M

- Généralités sur la syntaxe
- Instructions de traitement de données
- Instructions de branchement et de contrôle de flot
- Instructions d'accès mémoire
- Instructions génératrices d'exception
- Blocs conditionnels If, then
- La pile en fonctionnement
- Registres de limite de pile
- Instructions exclusive load et store, mise en œuvre de séquences atomiques
- Memory barriers et synchronisation

Synchronisation et sémaphores

- Instructions d'accès exclusif
- Les moniteurs local, global et externe
- Interaction avec les instructions d'accès exclusif
- Utilisation et contraintes de Load Exclusive et Store Exclusive

Jeu d'instructions DSP du Cortex-M

- Instructions de multiplication
- Instructions de packing / unpacking
- Instructions ARM SIMD V6 d'addition / soustraction packées
- Instructions SIMD d'addition/soustraction combinées, mise en œuvre des opérations complexes canoniques
- Instructions multiply et multiply accumulate
- Instructions SIMD sum absolute difference

- Instruction SIMD select
- Instructions de saturation

Troisième jour

Prise en charge DSP par CMSIS

- Le cadre de la bibliothèque CMSIS
- Les fonctions intrinsèques CMSIS DSP
- Les fonctions CMSIS optimisées de traitement de données

Unité de calcul flottant

- Introduction à l'IEEE754
- Arithmétique flottante
- FPU simple précision du Cortex-M4F
- Banc de registres
- Activation de la FPU
- Performances de la FPU, MAC fusionné
- Amélioration des performances par le choix du mode flush-to-zero et du mode default NaN
- Extension de l'AAPCS pour inclure les registres FP

Astuces de compilation C/C++ pour le Cortex-M33

- Mélange de C/C++ et d'assembleur
- Programmation avec le compilateur GCC
- Mesure de l'utilisation de la pile
- Accès non alignés
- Questions liées aux données locales et globales, alignement des structures
- Autres optimisations, retour de l'éditeur de liens

Exceptions

- Comportement des exceptions, retour d'exception
- Exceptions non masquables
- Privilège, modes et piles
- Escalade de fautes
- Élévation de priorité
- Table de vecteurs

Quatrième jour

Interruptions

- Fonctionnement de base des interruptions, mécanisme d'interruption microcodé
  - Entrée / sortie d'interruption, chronogrammes
  - Pile d'interruption
  - Tail chaining
- Temps de réponse aux interruptions, préemption
- Priorisation des interruptions
- Handlers d'interruption
- Le Nested Vectored Interrupt Controller (NVIC)

L'extension de sécurité

- Security states
- Banking des registres entre security states

- Piles et security states
- Security Extension et exceptions
- Protection d'adresse en état secure
- Interactions entre états secure et non-secure
  - Transitions d'état secure
  - Appels de fonction depuis l'état non-secure vers l'état secure
  - Retour depuis l'état secure
- Exceptions et Security Extension
  - Traitement des Secure Exceptions
  - Traitement des Non-Secure Exceptions depuis l'état secure
  - Retour d'une exception non-secure vers l'état secure
- La Security Attribution Unit
- L'Implementation Defined Attribution Unit

Memory Protection Unit

- Types de mémoire
- Ordre des accès
- Memory barriers, code auto-modifiant
- Présentation de la protection mémoire, PMSA de l'ARM v8
- MPU du Cortex-M33 et bus faults
- Registres d'état et d'adresse de faute
- Présentation des régions, type de mémoire et contrôle d'accès
- Mise en place de la MPU

Fonctions de débogage

- Débogage invasif
  - Infrastructure de débogage CoreSight
  - Mode halt
  - Vector catching
  - Sources d'événements de débogage
  - Fonctions flash patch et breakpoint
  - Data watchpoint and trace
  - Spécification ARM debug interface
  - Composants CoreSight
  - AHB-Access Port
  - Mises en œuvre possibles du DP : Serial Wire JTAG Debug Port [SWJ-DP] ou SW-DP
- Débogage non invasif
  - Fonctionnement de base de l'ETM
  - Principes de la trace d'instructions
  - Instrumentation trace macrocell
  - Registres de port de stimulus de l'ITM
  - Paquets de trace DWT
  - Types d'événements matériels
  - Trace d'instructions
  - Paquets de synchronisation
  - Interface entre les données de trace sur puce issues de l'ETM et l'Instrumentation Trace Macrocell [ITM]
  - Composants du TPIU
  - Connexion Serial Wire