RM4 - Mise en œuvre du Cortex-M7

Ce cours couvre le processeur Cortex-M7 conforme à l'architecture V7E-M

Objectifs

- Ce cours se divise en 3 parties importantes :
  - Architecture du Cortex-M7
  - Mise en œuvre logicielle et débogage du Cortex-M7
  - Mise en œuvre matérielle du Cortex-M7.
- À travers un tutoriel, la programmation bas niveau du Cortex-M7 est expliquée, en particulier le paramétrage de l'éditeur de liens ARM et certaines instructions assembleur délicates.
- Notez que les participants peuvent rejouer ces travaux pratiques après la formation.
- Le cours indique également comment utiliser les caches et les TCM, unités nouvelles par rapport au Cortex-M4.
- Le cours détaille également la mise en œuvre matérielle et fournit des règles pour concevoir un SoC à base de Cortex-M7, en tirant parti des transactions AXI/AHB simultanées.
- Une présentation de la spécification CoreSight est donnée avant la description des unités liées au débogage.

Prérequis et cours associés

- Notre cours de référence cours RI0 - AXI3 / AXI4 INTERCONNECTdétaille le fonctionnement du bus AXI4.
- Le contenu peut être adapté lorsque les participants connaissent déjà le Cortex-M4.

Environnement du cours

- Cours théorique
  - Support PDF (en anglais), imprimé en présentiel ; à distance via Teams.
  - Assistance du formateur tout au long de la formation.
- Chaque session débute par un point avec les stagiaires.

Audience visée

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

Plan du cours

Premier jour - architecture

Introduction à l'arm Cortex-M7

- Interfaces mémoire : TCM, caches et port périphérique AHB
- Plan mémoire figé
- Options de configuration
- Mise en œuvre en lock-step

Cœur ARM Cortex-M7

- Mise en évidence des nouveautés de l'architecture V7E-M
- Pipeline superscalaire dans l'ordre
- Prédiction de branchement dynamique

- Bit-banding
- Timer système
- System control block
- Détail de la Data Processing Unit
- Load Store Unit, bufferisation des écritures

Caches et TCM

- Caches
  - Mise en œuvre du cache dans le Cortex-M7
  - Opérations de maintenance du cache
  - Mode dynamic read allocate, récupération après erreur
  - Fusion dans le store buffer
  - Code correcteur d'erreur des caches L1
- TCM
  - Intérêt de mettre en œuvre deux ports D-TCM distincts
  - Mise en œuvre d'un ECC externe pour les TCM

Deuxième jour - architecture

Architecture d'un SOC à base de Cortex-M7

- Matrice de bus interne
- Accès à la ROM
- Accès à la SRAM
- Connexion des périphériques
- Partage des ressources entre le Cortex-M7 et les autres processeurs
- Architecture du STM32F7

Gestion des ressources exclusives et modes basse consommation

- Atomicité dans les systèmes monoprocesseur multitâche
- Fonctionnement du moniteur local
- Wait For Interrupt
- Événements

Exceptions

- Comportement des exceptions
- Instructions reprenables après exception
- Exceptions non masquables
- Traitement des fautes MPU, fautes externes
- Élévation de priorité
- Séquence de reset, contraintes d'initialisation

Interruptions

- Entrée / sortie d'interruption, chronogrammes
- Tail chaining
- Registres du NVIC
- Priorisation des interruptions
- Handlers d'interruption
- Wake-up Interrupt Controller

Troisième jour - unités optionnelles, mise en œuvre matérielle

Memory protection unit

- Ordonnancement des mémoires device et normal
- Restrictions d'accès selon le type de mémoire
- Restrictions d'ordonnancement mémoire
- Présentation de la protection mémoire, PMSA de l'ARM v7
- Registres d'état et d'adresse de faute
- Présentation des régions, type de mémoire et contrôle d'accès, sous-régions
- Recouvrement des régions
- Mise en place de la MPU

Unité de calcul flottant

- Introduction à l'IEEE754,
- Arithmétique flottante
- FPU simple et double précision du Cortex-M7
- Prise en charge matérielle des dénormalisés et de tous les modes d'arrondi IEEE
- 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
- Sauvegarde paresseuse du contexte flottant
- Mise en évidence des nouveautés du FPv5

Mise en œuvre AXI

- Présentation de la spécification du bus AXI, explication des règles d'ordonnancement
- Attributs et transactions AXI
- Restrictions sur les transferts AXI

Mise en œuvre AHB

- Présentation du protocole AHB-Lite
- AHBP
- AHBS

Mise en œuvre matérielle du Cortex-M7

- Brochage
- Horloges et reset, gestion de l'énergie
- Utilisation d'un Wake-up Interrupt Controller (WIC) externe

Cache de niveau 2 L2C-310

- Configurabilité du cache
- Caractéristiques de l'interface AXI
- Comprendre à travers des séquences comment l'information cacheable est copiée de la mémoire vers les caches de niveau 1 et de niveau 2
- Opérations transitoires, utilisation des line buffers LFB, LRB, EB et STB
- Gestion de l'énergie
- Surveillance des événements du cache
- Verrouillage du cache
- Gestion des interruptions

Quatrième jour - débogage, conception logicielle

Débogage invasif

- Infrastructure de débogage CoreSight, DAP
- Fonctions de débogage du Cortex-M7
- Mode halt
- Vector catching
- Mode moniteur
- Sources d'événements de débogage
- Fonctions flash patch et breakpoint
- Data watchpoint and trace
- Interface de débogage AHB-lite
- Table ROM

Débogage NON invasif

- Fonctionnement de base de l'ETM
- Principes de la trace d'instructions
- Instrumentation Trace Macrocell
- Paquets de trace DWT
- Paquets d'horodatage
- Trace d'instructions, paquets de branchement, paquets de trace d'exception
- Composants du TPIU
- Embedded Trace Buffer

Cross-trigger interface

- Rôle de cette unité de débogage
- Signaux de déclenchement vers la CTI et signaux de déclenchement issus de la CTI

Développement de logiciel embarqué avec le Cortex-M7

- Processus de développement embarqué
- 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
- Construction et débogage de votre image
- Veneers de branchement long
- Règles de programmation lorsqu'un cache est utilisé

ASTUCES DE COMPILATION C/C++ POUR LE Cortex-M7

- Optimisations du compilateur ARM, optimisation des appels terminaux, inlining des fonctions
- Mélange de C/C++ et d'assembleur
- Programmation avec le compilateur ARM
- 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

Jeu d'instructions Thumb-2

- Instructions de traitement de données

- Instructions de branchement et de contrôle de flot
- Instructions génératrices d'exception
- Blocs conditionnels If…then
- La pile en fonctionnement
- Memory barriers et synchronisation

Jeu d'instructions DSP du Cortex-M7

- Instructions de multiplication
- Instructions de packing / unpacking
- Instructions ARM SIMD V6 d'addition / soustraction packées
- Instructions SIMD d'addition/soustraction combinées
- Instructions multiply et multiply accumulate
- Instructions SIMD sum absolute difference
- Instruction SIMD select
- Instructions de saturation