RR3 - Mise en œuvre et conception logicielle sur ARM Cortex-R52/R52+

Objectifs

- Comprendre l'architecture ARM v8-R et les fonctions des Cortex-R52/R52+.
- Apprendre le comportement du pipeline et l'exécution des instructions.
- Maîtriser le traitement des exceptions et les systèmes mémoire.
- Explorer la virtualisation et les fonctions de sûreté.
- Mettre en œuvre des techniques de synchronisation et de débogage.

Prérequis

- Connaissances de base de l'architecture ARM
- Familiarité avec les systèmes embarqués
- Expérience de la programmation en assembleur (optionnel)

Public visé

- Développeurs logiciels travaillant sur architecture ARM
- Architectes système
- Ingénieurs en systèmes embarqués

Environnement du cours

- Cours théorique
  - Support imprimé et PDF (en anglais).
  - 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

Présentation de l'architecture ARM v8-R

- Registres du cœur
- Modèle d'exception
- Jeux d'instructions
- Modèle mémoire
- Virtualisation

Introduction aux Cortex-R52/R52+

- Présentation
- Système mémoire
  - Mémoire TCM
  - Caches de niveau 1

    - Accès direct à la mémoire interne
    - Interface AXIM
    - Port périphérique à faible latence
    - Interface flash
    - Interface AXIS
    - Détection et traitement des erreurs
- Sûreté et options configurables
    - DCLS
    - Spin-lock

Pipeline et jeu d'instructions

- Pipeline
- Exécution des instructions
- Instructions conditionnelles
- Instructions positionnant les drapeaux
- Temps d'exécution
    - Nombre de cycles des instructions
    - Nombre de cycles des instructions de base
    - Comportement du pipeline
    - Skewing
    - Dual-issuing
    - Load / Store
    - Division et racine carrée
    - Instructions Multiply accumulate flottantes et Advanced SIMD
    - Instructions au comportement particulier
- Instructions A32 et T32

Deuxième jour

Modèle d'exceptions

- État d'exception
- Niveaux d'exception
- État de reset dans l'ARMv8-R
- Interruption
    - Contrôleur
    - Traitement
- Virtualisation

Système mémoire de niveau 1 (cache et TCM)

- Cache
    - Bases du cache : organisation, algorithme de remplacement, politiques d'écriture
    - Organisation du cache
    - Politique write with allocate
    - Comprendre le chargement / rangement transitoire des lignes de cache : linefill buffers, eviction buffer
    - Opérations de maintenance du cache
- TCM
    - Tightly Coupled Memories, décodage d'adresse
    - Configuration des ITCM et DTCM
    - Accès aux TCM depuis l'interface esclave AXI
    - Protection ECC, détection et correction d'erreur interne aux TCM
    - Préchargement des TCM avec ECC
    - Utilisation des TCM dès le reset
- Ordonnancement mémoire
- Memory Barriers

- Gestion des ressources partagées

Memory Protection Unit (MPU)

- Présentation de la protection mémoire
- Régions de la MPU
- Prise en charge de la virtualisation

Troisième jour

GICv3

- Introduction
- Principes fondamentaux
- Configuration du GIC (SPI, PPI, SGI …)
- Traitement des interruptions
- Configuration des LPI
- Virtualisation

Virtualisation ARMv8-R et considérations de sûreté

- Introduction à la virtualisation
  - Concepts de base
  - Interruptions virtuelles
  - Interception des exceptions
  - Interception des instructions et des accès aux registres
  - PMSA (MPU)
  - Timers virtuels
  - Identifiants de machine virtuelle
  - Registres de sauvegarde
- Exemples de virtualisation
  - Commutateur d'OS invités
  - Moniteur d'OS
  - Interruptions virtuelles

Présentation de la synchronisation

- Interruptions inter-processeurs
- Identifiant de cluster
- Moniteur d'accès exclusif, mise en œuvre de sémaphores booléens
- Moniteur global
- Mise en œuvre du spin-lock
- Utilisation des événements

Performance Monitoring Unit (PMU)

- Interface d'événements
- Compteurs
- Signaux d'authentification et comportement de la PMU

Débogage

- Présentation du débogage (débogage externe, self-hosted debug …)
- Cross trigger
- Embedded Trace Macrocell (ETM)