FCC4 - Mise en œuvre du e6500

Ce cours couvre le cœur e6500 présent dans les SoC NXP T2 et T4

Objectifs

- *Ce cours poursuit 6 objectifs principaux :*
  - *Présenter le niveau de privilège hypervisor, la capacité à supporter plusieurs systèmes d'exploitation invités, la répartition de charge et la virtualisation.*
  - *Clarifier les mécanismes de base nécessaires dans un système multicœur : séquences atomiques, interruptions doorbell, messagerie.*
  - *Apprendre le mécanisme d'exception, en se concentrant sur l'unité proxy et en donnant des règles pour mettre en œuvre l'imbrication.*
  - *Expliquer le fonctionnement et l'initialisation de la MMU et des caches.*
  - *Mettre en évidence les problèmes de cohérence de cache et de TLB et expliquer le snooping.*
  - *Décrire les unités de débogage.*

Prérequis

- *Une expérience d'un processeur ou d'un DSP 32 bits est indispensable.*

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

PRÉSENTATION DU CŒUR e6500

- *Nouveau concept de cluster*
- *Compatibilité logicielle avec les e500mc et e5500*

Multi-threading, mode 64 bits et hypervisor

- *Dual-Threading*
- *Arithmétique en virgule fixe 64 bits*
- *Niveau de privilège hypervisor du e6500*
- *Collaboration entre l'OS invité et l'hypervisor pour recharger les TLB*
- *Messagerie au sein d'un domaine de cohérence*

Pipeline d'instructions

- *Double émission, exécution dans le désordre*
- *Modèle d'exécution*

- *Rôle de l'instruction isync*
- *Temps d'exécution des instructions, latence et cadence de répétition*
- *Gestion des branchements*
- *Mémoire guarded*
- *Règles de codage*

Unité de calcul en virgule flottante

- *Bases de la norme IEEE754*
- *Fonctionnement de la FPU*
- *FPU entièrement pipelinée*

Unités altivec

- *Fonctionnement vectoriel et scalaire*
- *Registres AltiVec, initialisation du VSCR*
- *Utilisation de VRSAVE pour éviter de sauvegarder et restaurer tous les registres vectoriels volatils*
- *Extension ANSI C pour supporter les opérateurs vectoriels*
- *Mise en œuvre d'AltiVec sur le e6500 : les unités d'exécution VALU et VPU*
- *Extension de l'EABI pour supporter AltiVec*

Le mecanisme d'exception

- *Gestion des exceptions : construction de la table de handlers via les registres IVPR et IVOR*
- *Identification de la cause exacte de l'exception via les registres de syndrome*
- *Conditions requises pour supporter l'imbrication des exceptions*
- *Interrupt proxy*
- *Exceptions multicœur, doorbells et messages*
- *Timers intégrés, time base, decrementer, FIT et WDT*
- *Routine de service du watchdog*

Chemins de données et d'instructions

- *Gestion des ressources exclusives*
- *Mise en œuvre d'une routine de spin lock*
- *Instruction wait*
- *Decorated storage facility*
- *Barrières mémoire*

L'unité de gestion mémoire

- *Traduction d'adresses, comprendre l'adresse virtuelle intermédiaire sur 86 bits*
- *Protection des processus par le TID*
- *Architecture de MMU à deux niveaux*
- *Organisation des TLB*
- *Rechargement logiciel des TLB*
- *Instruction TLB invalidate local*
- *Tirer parti des pages de grande taille*
- *" Virtualization fault*
- *" Rechargement matériel des TLB pour les pages de 4 Ko*
- *" Protection de parité des TLB*
- *" Cache de traduction d'adresse logique vers adresse réelle*

Caches L1 et L2, snooping

- *Bases des caches*
- *Cache L1 du e6500 : fonctionnement Write-Through*
- *Cache L2 partagé au sein du cluster*

- *Instructions liées aux caches*
- *Détection et correction d'erreurs des caches L1 et L2, injection d'erreurs*

Gestion de l'energie du cœur et du cluster

- *Connexion à l'unité PM de la plateforme*
- *Etats d'alimentation : thread, cœur et cluster*
- *Instruction wait for interrupt*
- *Etat drowsy d'AltiVec*

Débogage

- *Performance monitor*
- *Interruption de débogage*
- *Points d'arrêt sur instruction et sur données, programmation des plages d'adresses*
- *Message d'acquisition de données de débogage*
- *Instruction Debug Notify Halt*
- *Trace Nexus*
- *Activation et désactivation de la trace de propriété*