

V4 - Optimisation des FPGA

Architecture matérielle

Objectifs

- Concevoir et optimiser des systèmes à base de FPGA en performances, surface et consommation
- Comprendre les problématiques de timing, de pipeline et de franchissement de domaines d'horloge
- Appliquer des stratégies efficaces d'utilisation de la mémoire et des ressources
- Développer des implémentations matérielles d'algorithmes DSP et mathématiques
- Maîtriser les techniques de codage RTL pour la synthèse et la fiabilité du matériel
- Réaliser simulation, vérification et analyse temporelle
- Comprendre la conception physique : floorplanning, placement-routage et contraintes
- Intégrer les designs FPGA aux systèmes logiciels embarqués
- Explorer la co-conception matériel/logiciel et les techniques d'accélération
- Utiliser ARM NEON pour l'optimisation des performances côté logiciel

Prérequis

- Notions de VHDL ou Verilog
- Connaissance du langage C (voir par exemple notre cours L2)
- Familiarité avec les concepts FPGA

Environnement du cours

- Cours théorique
 - Support de cours au format PDF (en anglais)
 - Le formateur répond aux questions des stagiaires' durant la formation et apporte une assistance technique et pédagogique
- Travaux pratiques
 - Les travaux pratiques représentent de 40% à 50% de la durée du cours
 - Exemples de code, travaux pratiques et corrigés
 - Vivado ou Libero pour la conception, la synthèse et l'analyse temporelle ; ModelSim ou Vivado pour la simulation

Audience visée

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

Plan du cours

Premier jour

Concevoir pour la vitesse

- Débit élevé
- Faible latence
- Timing
 - Ajouter des étages de registres
 - Structures parallèles
 - Aplatir les structures logiques
 - Équilibrage des registres

- Réordonner les chemins

Concevoir pour la surface

- Replier le pipeline
- Réutilisation de logique pilotée par le contrôle
- Partage de ressources
 - Impact du reset sur la surface
 - Ressources sans reset
 - Ressources sans set
 - Ressources sans reset asynchrone
 - Remettre à zéro une RAM
 - Utiliser les broches Set/Reset des bascules

Concevoir pour la consommation

- Contrôle de l'horloge
 - Clock Skew
 - Maîtriser le skew
- Contrôle des entrées
- Réduire la tension d'alimentation
- Bascules déclenchées sur les deux fronts
- Modifier les terminaisons

Exemple de design : l'Advanced Encryption Standard

- Architectures AES
 - Un étage pour Sub-bytes
 - Aucun étage pour Shift Rows
 - Deux étages de pipeline pour Mix-Column
 - Un étage pour Add Round Key
 - Architecture compacte
 - Architecture partiellement pipelinée
 - Architecture entièrement pipelinée
- Performances contre surface
- Autres optimisations

Deuxième jour

Conception de haut niveau

- Techniques de conception abstraite
- Machines d'état graphiques
- Conception DSP
- Fondamentaux des threads en co-conception logiciel/matériel

Domaines d'horloge

- Franchir les domaines d'horloge
 - Métastabilité
 - Solution 1 : contrôle de phase
 - Solution 2 : double bascule
 - Solution 3 : structure FIFO
 - Partitionner les blocs de synchronisation
- Horloges gâtées dans les prototypes ASIC
- Module d'horloges
- Statistiques d'exécution de la suppression du gating

Implémenter les fonctions mathématiques

- Division matérielle
 - Multiplication et décalage
 - Division itérative
 - La méthode de Goldschmidt
- Développement en séries de Taylor et Maclaurin
- L'algorithme CORDIC

Troisième jour

Circuits de reset

- Asynchrone ou synchrone
 - Problèmes des resets entièrement asynchrones
 - Resets entièrement synchronisés
 - Assertion asynchrone, désassertion synchrone
- Mélanger les types de reset
 - Bascules non remises à zéro
 - Resets générés en interne
- Domaines d'horloge multiples

Simulation avancée

- Architecture du banc de test
 - Composants du banc de test
 - Déroulement du banc de test
- Thread principal
- Horloges et resets
- Cas de test
- Stimuli système
 - MATLAB
 - Modèles fonctionnels de bus
- Couverture de code
- Simulations au niveau portes
- Couverture de basculement
- Pièges à l'exécution
 - Timescale
 - Rejet des glitches
 - Modélisation des délais combinatoires

Floorplanning

- Partitionnement du design
- Floorplanning du chemin critique
- Dangers du floorplanning
- Floorplanning optimal
 - Chemin de données
 - Fort fan-out
 - Structure du composant
 - Réutilisabilité
- Réduire la dissipation de puissance

Quatrième jour

Analyse temporelle statique

- Analyse standard
- Verrous
- Circuits asynchrones
 - Rebouclage combinatoire

Problématiques de PCB

- Alimentation
 - Besoins en alimentation
 - Régulation
- Condensateurs de découplage
 - Principe
 - Calcul des valeurs
 - Placement des condensateurs

Exemple de design : microprocesseur (RISCV NEORV32, IBX)

- Architecture SRC
- Optimisations de synthèse
 - Vitesse contre surface
 - Pipeline
 - Synthèse physique
- Optimisations de floorplan
 - Floorplan partitionné
 - Floorplan du chemin critique

Annexes

Optimisation de la mémoire dans les designs FPGA

- Types de mémoire dans un FPGA
 - Bascules (FF), LUT RAM, Block RAM (BRAM) et UltraRAM
- Quand NE PAS utiliser de bascules
 - Explosion des ressources et impact sur le routage
- Mapping mémoire efficace
 - Utiliser la BRAM pour les tampons et les FIFO
 - Inférer une RAM en HDL
- Stratégies d'utilisation de la RAM distribuée

Implémentation DSP et FFT

- Blocs DSP dans un FPGA
 - Multiplieurs et unités MAC
- Architectures de FFT
 - Bases du Radix-2 / Radix-4
 - FFT pipelinée ou itérative
 - Compromis virgule fixe et virgule flottante
 - Compromis débit et ressources