

Résumé

Les travaux présentés dans cette thèse s'inscrivent dans le cadre des recherches menées sur la conception et l'implémentation des outils de génération et de parallélisation automatique de code à hautes performances. Ces recherches visent à accélérer et à optimiser la conception des systèmes sur puce (SoCs), en particulier des systèmes multiprocesseurs sur puce (MPSoCs). Il s'agit de proposer une approche de parallélisation permettant un chemin automatique commençant du code séquentiel jusqu'à l'exécution du code parallèle sur les systèmes multiprocesseurs sur puce.

Suite à l'étude bibliographique portant sur les modèles de programmation parallèle et les outils de parallélisation automatiques existants, nous avons établi un modèle de mesure qui permet de calculer les consommations et les performances de ces outils à travers des métriques et des benchmarks parallèles. L'un des principaux résultats de notre étude théorique et expérimentale est la nécessité de choisir la méthodologie de parallélisation la plus convenable pour un tel modèle de programmation afin de garantir une efficacité maximale. L'objectif principal de cette thèse est d'augmenter le niveau d'efficacité du système de parallélisation de code tout en minimisant le coût et la complexité d'un tel système. Tout cela doit être construit selon une méthode de conception intégrant le paralléliseur automatique le plus adéquat et automatisant l'exécution des codes parallèles. Une approche de conception d'un paralléliseur automatique du code C à base du compilateur PLUTO est alors proposée. Cette conception se base sur l'implémentation d'un analyseur syntaxique du code parallèle openMP (C avec directives openMP) et des générateurs automatiques de code qui s'appellent ACAGen et MppSoCGEN.

L'approche proposée est validée à travers des tests de génération et des expérimentations sur FPGA en parallélisant des applications de traitement d'images.

Mots clés: parallélisation automatique, MPSoC, PLUTO, Métriques, openMP, ACAGen, MppSoCGEN ;