

Version abrégée

La forte croissance du marché des systèmes embarqués durant ces dernières années engendre un important besoin de processeurs performants, sujets à de sévères contraintes de puissance consommée et de coût. Ces processeurs embarqués équipent en effet des téléphones mobiles, des agendas électroniques (PDAs), des imprimantes ou encore des équipements de réseaux informatiques.

Le parallélisme au niveau d'instruction est l'une des principales techniques permettant l'augmentation des performances des processeurs équipant les stations de travail. Ces circuits sont toutefois de haute complexité et leur puissance dissipée s'avère très importante. Le parallélisme constitue également une technique permettant la réduction de la consommation d'énergie d'un circuit. Cette double caractéristique rend le parallélisme au niveau d'instruction très attrayant pour ces processeurs embarqués requérant un haut niveau de performance et une faible consommation de puissance.

Cette thèse se concentre sur la conception d'architectures parallèles de basse consommation d'énergie offrant une synergie entre le compilateur et le matériel, comme par exemple les processeurs EPIC et VLIW. Cette interaction compilateur-processeur permet le déplacement d'une grande partie de la complexité des architectures superscalaires vers le compilateur, tout en conservant le même niveau de performance. Cependant, l'introduction du parallélisme dans la *datapath* d'un processeur modifie fortement son architecture. Afin de comprendre et de quantifier les répercussions de ces modifications, nous avons développé un nouveau processeur de type EPIC, appelé *DEVIL*, dont l'implémentation a fait l'objet d'une analyse détaillée.

DEVIL intègre une unité de *fetch* particulière permettant d'encoder le parallélisme d'un paquet d'instructions et supportant des instructions à taille variable. Ce mécanisme permet l'obtention d'un code 50% plus compact que celui d'un processeur VLIW standard, tout en maintenant le même niveau de performance.

DEVIL peut exécuter jusqu'à deux instructions en parallèle à chaque coup d'horloge. Cette caractéristique permet d'augmenter d'un facteur moyen de 1.5 les performances par rapport à un processeur scalaire. Ce gain compense les pertes de vitesse d'exécution induites par les modes de fonctionnement à basse fréquence et basse tension requis pour la faible consommation. Nous montrons que *DEVIL* exécute des tâches à la même vitesse qu'un processeur scalaire, tout en consommant 38% d'énergie en moins.

Les processeurs exploitant le parallélisme au niveau d'instruction souffrent généralement d'une augmentation de la taille du code. Cet effet indésirable est restreint par le mécanisme de *fetch* inclus dans *DEVIL*. La taille de code de *DEVIL* demeure cependant supérieure à celle d'un processeur scalaire. L'exécution à prédicats (ou exécution conditionnelle) constitue une solution à ce problème. Les résultats de nos travaux établissent que, moyennant l'exécution à prédicats et une unité de *fetch* adéquate, il est possible de générer du code 12% plus rapide et 25% plus petit.