

1 IRISA-WP

1.1 Work Package 1 : Un compilateur recible pour MIPS étendu

Délivrable : software

Implémentation d'un back-end de compilation ciblant une version étendue du processeur MIPS, et opérant à partir de la représentation intermédiaire commune définie en ?? et issue de GCC. Ce *back-end* intégrera en particulier des passes d'extraction de motifs de calculs (sous-graphes), ainsi une passe de sélection d'instructions basée sur des techniques de couvertures de graphes, permettant d'exploiter au mieux les motifs d'instruction "spécialisés" spécifiés par l'utilisateur et/ou extraits à partir de l'application. Ici, il faut voir si la RI proposée ne permettrait pas de régénérer un code C dans lequel l'utilisation d'instruction spécialisée se fait au travers de directives de type `asm{ ... }`. une telle approche offrirait une flexibilité accrue, sans impacter la qualité/performance des résultats obtenus.

1.2 Work Package 2 : Définition d'un modèle simplifié de micro-architecture MIPS extensible

Délivrable : software

Définition d'un modèle extensible de micro-architecture basée sur un processeur de type MIPS pipeliné à 5 étage (incluant cache de données et instruction), offrant à l'utilisateur la possibilité de définir ses propres extensions architecturales, au travers d'un Domain Specific Language (on exploitera les technologies d'IDM XText-EMF).

On mettra également en œuvre un outil de génération de description matérielle synthétisable (VHDL) de la micro-architecture à partir de ce modèle, en utilisant des technologies d'Ingénierie dirigée par les modèles (EMF-XPAND)

Deux versions de cet outil sont envisagées, dans la première (qui est l'objet de ce WP), on restreindra les possibilités de communication entre le processeur et ses extensions à des communications passant par la file de registre du processeur (en permettant éventuellement un plus grand nombre d'accès en lecture/écriture par cycle).

1.3 Work Package 3 : Définition d'un modèle complexe de micro-architecture MIPS extensible

Délivrable : rapport/software ?

Dans la seconde version (plus orientée exploratoire) on souhaite pouvoir lever la limitation portant sur les communications et permettre un couplage plus fin entre les extensions et le cœur de la micro-architecture, par exemple,

en proposant un accès direct au cache de données et/ou en donnant la possibilité aux extensions de réutiliser les opérateurs (par exemple le multiplieur 32×32 bits) mis en oeuvre dans le chemin de donnée natif du processeur.

Ici on pourrait également envisager des connections directes avec d'autres composants au travers de structures similaire aux *FSL* disponibles sur les processeurs softcore Microblaze de la société Xilinx.

1.4 Work Package 4 : Définition d'un modèle complexe de micro-architecture MIPS extensible

Délivrable : rapport

Le dernier *package* a également un caractère exploratoire, et portera sur l'intégration de ce type d'extension architecturales au sein d'un compilateur. En particulier, il s'agira d'étudier comment il est possible d'intégrer ces instructions complexes dans la passe de selection de code, tout en s'assurant que leur contraintes d'utilisation soient respectées.