

МЕТОД ВЫДЕЛЕНИЯ ОБЛАСТЕЙ ПЛАНИРОВАНИЯ

к.т.н., доц. С.Ю. Гавриленко, студент М.С. Романько,
Национальный технический университет "ХПИ", г. Харьков

Подходы, используемые при оптимизации кода, могут существенно зависеть от критериев оптимизации. Обычно рассматривают три критерия или их комбинации с некоторыми приоритетами:

- минимизация времени выполнения программы;
- минимизация размера кода;
- минимизация энергопотребления.

Компилятор для *Instruction Level Parallelism (ILP)* процессора объединяет в себе стандартные механизмы компиляции. На первом этапе проводится лексический, синтаксический и семантический анализ программы на входном языке и строится промежуточное представление. Затем проводится оптимизация в терминах промежуточного представления. Как пример стандартных оптимизаций можно привести удаление избыточного кода, свертка константных вычислений из циклов, понижение мощности операции. В *ILP*-компиляции делается акцент на методе усилении программного параллелизма:

- выделение областей планирования;
- реорганизация кода, направленная на удлинение линейных участков и расширение областей планирования;
- усиление параллелизма в пределах выделенных областей;
- планирование команд в пределах выделенных областей.

Рассмотрим более подробно метод выделения областей планирования.

Область планирования – это фрагмент или множество фрагментов программы, в пределах которых применяется алгоритм планирования. *ILP*-процессоры реализуют планирование в рамках более широких областей кода, объединяющих несколько линейных участков, так что инструкции могут в результате перемещаться из одного участка в другие. При этом обычно стремятся максимально ускорить выполнение вдоль наиболее часто исполняемых ветвей программы. Для того чтобы избежать некорректного перемещения инструкций между линейными участками, используются определенные приемы, ограничения и аппаратные средства. Одной из идей данного метода является то, что код можно реорганизовать таким образом, чтобы сократить время выполнения одних путей за счет замедления других. Но такой подход может быть не приемлем для приложений реального времени. Метод не требует аппаратной поддержки и основан на добавлении минимального необходимого числа дополнительных линейных участков, содержащих зондирующий код.