

Исследование кодов программ на предмет избыточных команд*А. М. Ключников¹*¹АО «Интел А/О»

При исследовании кодов программ на языке ассемблер, можно встретить инструкции, исполнение которых можно упростить. Примером может служить команда *MOV EAX, EBX*, которая значение из одного архитектурного регистра переписывает в другой. В современных высокопроизводительных процессорах, при исполнении команды под каждый архитектурный регистр будет выделен физический регистр [1]. В результате исполнения команды *MOV*, значение из одного физического регистра переписывается в другой. Таким образом, при обработке команды *MOV* будет занято исполнительное устройство, а два физических регистра будут хранить одно и то же значение. Такую команду *MOV* можно назвать избыточной, а её исполнение может быть упрощено. Упрощение будет заключаться в назначении одно и того же физического регистра для двух архитектурных (*Move Elimination*) [2]. Также в статье [2] приводятся варианты упрощения для арифметических команд, у которых один из операндов является константой (*Constant Folding*).

Помимо команд, исполнение которых можно упростить, в кодах программ есть команды, вырабатывающие результат, который не используется другими командами и перезаписывается. Исполнение такой команды можно отменить. Отмена исполнения одной избыточной команды, может повлечь отмену исполнения цепочки зависимых друг от друга команд.

Для исследования, проводимого в данной статье, можно сформулировать определение избыточной команды: команда считается избыточной, если значение, которое она вырабатывает (пишет в регистр), ни разу не использовалось и было перезаписано последующими командами.

При расчете числа таких команд был использован программный потактовый симулятор процессора и набор трасс, для расчета производительности процессора. В потактовый симулятор добавлен механизм обнаружения избыточных команд, который за один прогон трассы, находит первую избыточную команду в цепочке. Для получения максимального числа избыточных команд, необходимо каждую трассу перезапустить несколько раз.

Результаты расчета числа команд показали (рис. 1), что исполнение 3% команд может быть отменено. При этом, отмена исполнения команд позволяет повысить производительность процессора на 0.4% (рис. 2).

Стоит заметить, что исполнение не всех команд может быть отменено. Нельзя отменять исполнение команд, которые могут вызвать прерывание: команды обращения в память, некоторые арифметические команды. Если убрать команды обращения в память из рассмотрения (рис. 1 и рис. 2, столбец 8), то исполнение 2% команд может быть отменено, что позволит увеличить производительность процессора на 0.2%.

Литература

1. *Hennessy J.L., Patterson D.A.* Computer Architecture, Fifth Edition: A Quantitative Approach. - Morgan Kaufmann, 2011. – 856 p.
2. *Petric V., Sha T., Roth A.* RENO - A Rename-Based Instruction Optimizer. ISCA 2005: 98-109

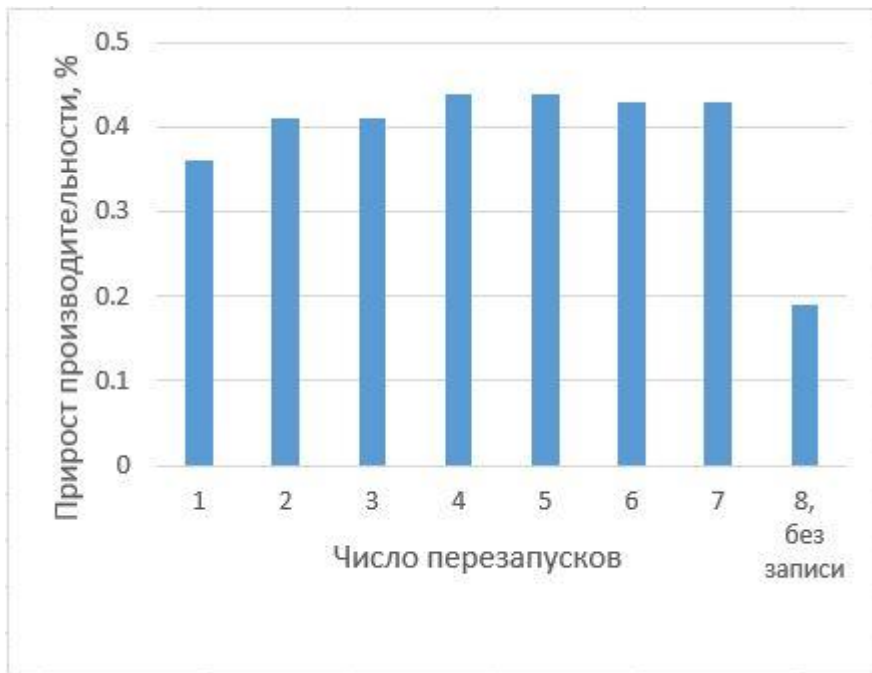


Рис. 1. Доля избыточных команд в зависимости от числа перезапусков

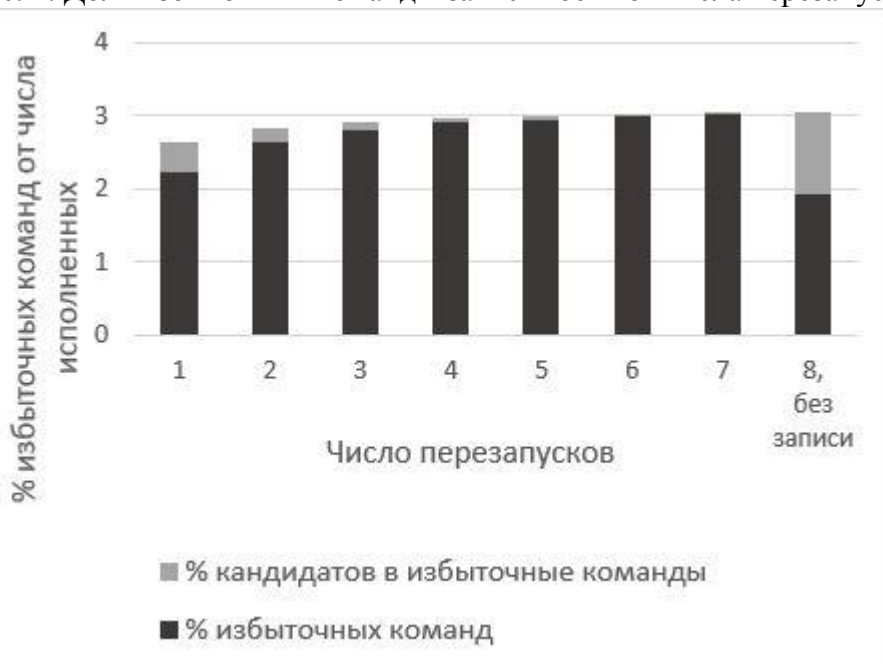


Рис. 2. Прирост производительности, полученный после отмены исполнения избыточных команд