

## ИЗМЕРЕНИЕ ЭНЕРГЕТИЧЕСКОЙ СТОИМОСТИ ИНСТРУКЦИЙ НА АРХИТЕКТУРЕ ARM

И.И.Каретин, В.А.Макаров

## ESTIMATING THE INSTRUCTION ENERGY CONSUMPTION ON THE ARM ARCHITECTURE

I.I.Karetin, V.A.Makarov

Институт электронных и информационных систем НовГУ,  
Ilya.Karetin@astrosoft-development.com, Vladimir.Makarov@novsu.ru

Описывается разработанный метод анализа энергетической стоимости индивидуальных инструкций из набора команд ARM Instruction Set. В экспериментах использовался энергетический симулятор хеему.

**Ключевые слова:** энергопотребление инструкций, ARM Instruction, эмулятор, энергосберегающая компиляция

This paper contains a description of the method of a total energy consumption estimation of the ARM Instruction Set. The power simulators XEEMU was used for accurate simulating the power consumption.

**Keywords:** energy consumption of instructions, ARM Instruction, emulator, energy saving compilation

В данной работе анализируется энергетическая стоимость индивидуальных инструкций из набора команд ARM Instruction Set.

Для запуска тестового кода использовался хеему-эмулятор микропроцессорного ядра XScale [1]. Данный эмулятор был разработан как часть научной работы по максимально точному моделированию функционирования процессора, описанной в статье [2].

Перечень инструкций для измерений был извлечен из документации по архитектуре ARM [3,4].

Тестовый код, используемый в ходе измерений, во всех запусках был совершенно одинаковым и представлял собой простой цикл, выполняющий целевую инструкцию 1024000 раз. Изменяемые регистры подбирались таким образом, чтобы они больше нигде не использовались на протяжении всей остальной программы, либо сразу же перезаписывались бы новыми значениями. Метка для инструкций перехода располагалась непосредственно следом за инструкцией.

Код был написан на ассемблере с целью устранения какого-либо влияния со стороны компилятора. Генерация исполняемого файла производилась с помощью утилиты arm-elf-gcc версии 4.3.3. Версия операционной системы на тестовой машине: ubuntu 12.04, ядро 3.2.0-53-generic-pae.

За счет использования эмулятора хеему была достигнута 100% воспроизводимость результата — различия в нагрузке на CPU, равно как и все прочие возможные изменения тестовой системы, не оказывали никакого влияния на результаты, выдаваемые эмулятором в каждом из опытов.

В нижеследующей таблице приведены результаты измерений.

Результаты измерений

Инструкции	Время выполнения (сек)	Затраты энергии на выполнение всей программы (Дж)	Средние затраты энергии на целевую инструкцию (наноДж)
—	0,0188	0,0089	—
nop	0,0205	0,0098	0,8789
add r0, r0, r0	0,0205	0,0098	0,8789
mul r0, r1, r1	0,0205	0,0100	1,0742
and r0, r0, r0	0,0205	0,0098	0,8789
lsl r0, r1	0,0222	0,0105	1,5625
lsl r0, #1	0,0205	0,0098	0,8789
lsr r0, r1	0,0222	0,0105	1,5625
lsr r0, #1	0,0205	0,0098	0,8789
asr r0, r1	0,0222	0,0105	1,5625
asr r0, #1	0,0205	0,0098	0,8789
lsls r0, r1	0,0222	0,0105	1,5625
lsls r0, #1	0,0205	0,0098	0,8789
lsrs r0, r1	0,0222	0,0105	1,5625
lsrs r0, #1	0,0205	0,0098	0,8789
asrs r0, r1	0,0222	0,0105	1,5625
asrs r0, #1	0,0205	0,0098	0,8789
mvn r0, r0	0,0205	0,0098	0,8789
add r0, r0, r0	0,0205	0,0098	0,8789
adc r0, r0, r0	0,0205	0,0098	0,8789
qadd r0, r0, r0	0,0222	0,0105	1,5625
qdadd r0, r0, r0	0,0222	0,0105	1,5625
sub r0, r0, r0	0,0205	0,0098	0,8789
sub r0, r0, r0	0,0205	0,0098	0,8789
rsb r0, r0, r0	0,0205	0,0098	0,8789
rsc r0, r0, r0	0,0205	0,0098	0,8789
qsub r0, r0, r0	0,0222	0,0105	1,5625
qdsb r0, r0, r0	0,0222	0,0105	1,5625
mul r0, r1, r1	0,0205	0,0100	1,0742
mla r0, r1, r1, r2	0,0205	0,0100	1,0742

Окончание табл.

umull r0, r5, r2, r2	0,0205	0,0100	1,0742
umlal r0, r5, r2, r2	0,0205	0,0100	1,0742
smull r0, r5, r2, r2	0,0205	0,0100	1,0742
smlal r0, r5, r2, r2	0,0205	0,0100	1,0742
smulbb r0, r1, r1	0,0205	0,0098	0,8789
smulbt r0, r1, r1	0,0205	0,0098	0,8789
smultb r0, r1, r1	0,0205	0,0098	0,8789
smultt r0, r1, r1	0,0205	0,0098	0,8789
smulWb r0, r1, r1	0,0205	0,0098	0,8789
smulWt r0, r1, r1	0,0205	0,0098	0,8789
smlabb r0, r1, r1, r4	0,0205	0,0098	0,8789
smlabt r0, r1, r1, r4	0,0205	0,0098	0,8789
smlatb r0, r1, r1, r4	0,0205	0,0098	0,8789
smlatt r0, r1, r1, r4	0,0205	0,0098	0,8789
smlaWb r0, r1, r1, r4	0,0205	0,0098	0,8789
smlaWt r0, r1, r1, r4	0,0205	0,0098	0,8789
smlal r4, r5, r1, r2	0,0205	0,0100	1,0742
smlalbb r4, r5, r1, r2	0,0205	0,0098	0,8789
smlalbt r4, r5, r1, r2	0,0205	0,0098	0,8789
smlalbtb r4, r5, r1, r2	0,0205	0,0098	0,8789
smlalatt r4, r5, r1, r2	0,0205	0,0098	0,8789
tst r0, r1	0,0205	0,0098	0,8789
teq r0, r1	0,0205	0,0098	0,8789
and r0, r1, r2	0,0205	0,0098	0,8789
eor r0, r1, r2	0,0205	0,0098	0,8789
orr r0, r1, r2	0,0205	0,0098	0,8789
bic r0, r1, r2	0,0205	0,0098	0,8789
cmpcq r0, r1	0,0205	0,0097	0,78125
cmneq r0, r1	0,0205	0,0097	0,78125
beq testlabel	0,0205	0,0097	0,78125
bleq testlabel	0,0205	0,0097	0,78125
b testlabel	0,0205	0,0098	0,8789
bl testlabel	0,0205	0,0098	0,8789
bne testlabel	0,0205	0,0098	0,8789
blne testlabel	0,0205	0,0098	0,8789
ldr r0, =Data1	0,0205	0,0100	1,0742
ldrb r0, =Data1	0,0205	0,0100	1,0742
ldrbsb r0, =Data1	0,0205	0,0100	1,0742
ldrhr r0, =Data1	0,0205	0,0100	1,0742
ldrsh r0, =Data1	0,0205	0,0100	1,0742
ldm r0, {r4-r10}	0,0341	0,0167	7,6172
mov r0, #Data1	0,0205	0,0098	0,8789
stm r0, {r4-r10}	0,0341	0,0163	7,2266
str r0, [sp, #100]	0,0205	0,0099	0,9766
strb r0, [sp, #100]	0,0205	0,0099	0,9766
strh r0, [sp, #100]	0,0205	0,0099	0,9766
ldr r4, r5, [r0, #100]	0,0205	0,0098	0,8789
strd r4, r5, [r0, #100]	0,0205	0,0098	0,8789
swp r4, r5, [r0]	0,0205	0,0100	1,0742

Первая строчка таблицы показывает затраты времени и энергии на выполнение тестовой программы без целевой инструкции (пустой цикл). Таким образом, можно измерить, какие именно затраты происходят при добавлении ровно одной требуемой инструкции.

На основании проведенных измерений можно сделать следующие выводы. Как и следовало ожи-

дать, работа с длинными (64-битными) целочисленными переменными требует затраты большей энергии по сравнению с остальными целыми типами, даже в том случае, если время выполнения остается таким же.

Удивителен тот факт, что операция `por` оказалась не самой энергоэффективной. Инструкции проверки условий могут потратить меньше энергии, а значит, в ряде случаев возможно их использование вместо `por` для пропуска такта, если это требуется алгоритмом. Так, например, использование операции условного перехода при ложном условии потратило меньше энергии.

Затраты времени и энергии на загрузку или сохранение группы регистров в память (`ldm`, `stm`) аналогичны использованию индивидуальных операций чтения или записи каждого регистра. Таким образом, даже при использовании этих групповых операций нужно минимизировать количество пересылаемых регистров.

Сохранение регистров в память оказалось энергетически менее затратной операцией по сравнению с их загрузкой из памяти, при этом время выполнения осталось неизменным.

Операции переходов (`beq`, `bleq`) оказались на удивление малозатратными, по сравнению с арифметическими операциями и командами пересылок. В то же время обычно именно они рассматриваются как основной потребитель времени исполнения и энергии. Возможно, данный факт связан с тем, что в тестовом коде очень хорошо отработал механизм предсказания переходов (`branch prediction`) [5,6]. Это наблюдение должно быть проверено в последующих работах.

Использование операций битового сдвига (`lsl`, `lsr`) с регистром в качестве второго параметра является крайне дорогим и по времени исполнения, и по энергозатратам. Размер сдвига должен быть предварительно вычислен на этапе компиляции всегда, когда это возможно — в таком случае становится возможным использование константного параметра, что, согласно полученным результатам, значительно эффективнее.

Большая часть сделанных выводов может быть использована в дальнейшем при разработке энерго-сберегающих компиляторных оптимизаций для архитектуры ARM.

1. XEEMU Homepage, <http://www.inf.u-szeged.hu/xeemu/index.html>
2. Herczeg Z., Kiss A., Schmidt D., et al. XEEMU: An Improved XScale Power Simulator // Proc. 17th International Workshop PATMOS 2007. Gothenburg, Sweden, 3-5 September 2007. Gothenburg, 2007. P.300-309.
3. ARM Infocenter, <http://infocenter.arm.com>
4. Intel® XScale™ Microarchitecture Assembly Language Quick Reference Card
5. Kumar N. and Johari R. Power Profiling and Analysis of MI-Benchmarks Using Xscale Power Simulator (XEEMU) // Global Journal of Researches in Engineering, Electrical and Electronics Engineering. 2012. V.12. Issue 7. P.39-42.
6. Intel XScale Core Developer's Manual. Ch.10.2 Branch Prediction. 2004. P.164.