

## # ADD.i vs ADD comparision

Бенчмарк направлен на сравнение производительности инструкций "add.i dst, src, 0" и "add dst, src, zero".

Для выполнения этой задачи был написан код на языке C с использованием ассемблерных вставок.

Цикл для add с ассемблерными вставками:

```
```c
void measure_add(int iterations, int dst, int src) {
    for (uint32_t i = 0; i < iterations; ++i) {
        asm volatile (
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            " add %0, %1, x0;"
            : "=r" (dst)
            : "r" (src)
        );
    }
}
```
```

Цикл для add.i с ассемблерными вставками :

```
```c
void measure_add(int iterations, int dst, int src) {
    for (uint32_t i = 0; i < iterations; ++i) {
        asm volatile (
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            " addi %0, %1, 0;"
            : "=r" (dst)
            : "r" (src)
        );
    }
}
```
```

Для увеличения времени выполнения исследуемых команд их количество в цикле было увеличено до 190 штук, а количество циклов равно 1,000,000,000.

Программа была запущена на плате Banana PI.

Для инструкции add были получены следующие результаты:

Профилировка: (perf record -e cpu-clock ./название файла >> perf report)

| Percent |                        |
|---------|------------------------|
|         | for (uint32_t i = 0; i |
|         | sw zero, -20(s0)       |
|         | ↓ j 32c                |
|         | asm volatile (         |
| 3.05    | lw a5, -44(s0)         |
|         | add a5, a5, zero       |
| 0.50    | add a5, a5, zero       |
| 0.45    | add a5, a5, zero       |
| 0.49    | add a5, a5, zero       |
| 0.47    | add a5, a5, zero       |
| 0.93    | add a5, a5, zero       |
| 0.49    | add a5, a5, zero       |
| 0.53    | add a5, a5, zero       |
| 0.44    | add a5, a5, zero       |
| 0.45    | add a5, a5, zero       |
| 0.45    | add a5, a5, zero       |
| 0.47    | add a5, a5, zero       |
| 0.56    | add a5, a5, zero       |
| 0.49    | add a5, a5, zero       |
| 0.46    | add a5, a5, zero       |
| 0.49    | add a5, a5, zero       |
| 0.55    | add a5, a5, zero       |
| 0.45    | add a5, a5, zero       |
| 0.43    | add a5, a5, zero       |
| 0.50    | add a5, a5, zero       |
| 0.49    | add a5, a5, zero       |
| 0.48    | add a5, a5, zero       |
| 0.53    | add a5, a5, zero       |
| 0.53    | add a5, a5, zero       |
| 0.53    | add a5, a5, zero       |
| 0.55    | add a5, a5, zero       |
| 0.51    | add a5, a5, zero       |
| 0.46    | add a5, a5, zero       |
| 0.47    | add a5, a5, zero       |

Время работы – 10 прогонов (perf stat -r 10 ./название файла):

|   |                    |   |                       |              |
|---|--------------------|---|-----------------------|--------------|
| Performance counter stats for './watermark_add2.out' (10 runs): |                    |   |                       |              |
| 12,601.67 msec  | task-clock:u       | # | 0.999 CPUs utilized   | ( +- 0.03% ) |
| 0   | context-switches:u | # | 0.000 /sec            |              |
| 0   | cpu-migrations:u   | # | 0.000 /sec            |              |
| 39  | page-faults:u      | # | 3.095 /sec            | ( +- 0.71% ) |
| 20,161,287,964  | cycles:u           | # | 1.600 GHz             | ( +- 0.02% ) |
| 19,921,456,580  | instructions:u     | # | 0.99 insn per cycle   | ( +- 0.00% ) |
| 100,018,507   | branches:u         | # | 7.937 M/sec           | ( +- 0.00% ) |
| 1,972   | branch-misses:u    | # | 0.00% of all branches | ( +- 3.28% ) |
| 12.61417 +- 0.00486 seconds time elapsed ( +- 0.04% )           |                    |   |                       |              |

Для инструкции add.i были получены следующие результаты:

Профилировка:

| Percent |     | for (uint32_t i = 0; i < |
|---------|-----|--------------------------|
|         |     | sw zero,-20(s0)          |
|         | ↓ j | lb0                      |
|         |     | asm volatile (           |
|         |     | " add.i %0, %1, 0;"      |
| 2.97    | lw  | a5,-44(s0)               |
| 0.00    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 1.00    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.52    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.49    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.52    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.51    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |
| 0.50    | mv  | a5,a5                    |

Время работы – 10 прогонов

|  |                    |   |                       |               |
|--|--------------------|---|-----------------------|---------------|
| Performance counter stats for './watermark_addi2.out' (10 runs): |                    |   |                       |               |
| 94,845.06 msec   | task-clock:u       | # | 1.000 CPUs utilized   | ( +- 17.67% ) |
| 0  | context-switches:u | # | 0.000 /sec            |               |
| 0  | cpu-migrations:u   | # | 0.000 /sec            |               |
| 38   | page-faults:u      | # | 0.401 /sec            | ( +- 0.98% )  |
| 151,748,260,813  | cycles:u           | # | 1.600 GHz             | ( +- 17.67% ) |
| 150,088,330,055  | instructions:u     | # | 0.99 insn per cycle   | ( +- 17.67% ) |
| 753,448,927  | branches:u         | # | 7.944 M/sec           | ( +- 17.67% ) |
| 3,051  | branch-misses:u    | # | 0.00% of all branches | ( +- 8.68% )  |
| 94.89 +- 16.77 seconds time elapsed ( +- 17.67% )                |                    |   |                       |               |

Исходя из полученных результатов можно сделать вывод что инструкция add выполняется быстрее чем add.i в 8 раз. Скорее всего это происходит из-за интерпретации команды add.i в mv.