Выполнил: студент группы А-13-18

Маренков Михаил Андреевич

Приняла: Шамаева О.Ю.

**Практическое задание №4**

Фрагмент программы:

**For i := 1 to n**

**C[i] := A[i] \* b[i]**

1. Скалярный процессор с простым конвейером без планирования (с учетом задержек)

|  |  |  |
| --- | --- | --- |
| 1 | Loop: LD F1, 0(R1) | Загрузка значения из вектора A в F1 |
| 2 | LD F2, 0(R2) | Загрузка значения из вектора B в F2 |
| 3 | приостановка | (1 такт) Загрузка > АЛУ |
| 4 | MULTD F3, F1, F2 | Умножение значений векторов A и B. Результат в F3 |
| 5 | приостановка | (2 такта) АЛУ > Запись |
| 6 | приостановка |  |
| 7 | SD 0(R3), F3 | Запись результата |
| 8 | SUBI R1, R1, #8 | Пересчет указателя для вектора A |
| 9 | SUBI R2, R2, #8 | Пересчет указателя для вектора B |
| 10 | SUBI R3, R3, #8 | Пересчет указателя для вектора C |
| 11 | BNEZ R1, Loop | Переход к началу цикла, если R1 не равно нулю |
| 12 | приостановка | (1 такт) Переход |

Для выполнения цикла требуется 12 тактов на итерацию; одна приостановка для команды **LD**, две для команды **MULTD** и одна для задержанного перехода.

Число машинных тактов для этого случая составляет: **12\*n**

1. Скалярный процессор с простым конвейером + Разворачивание цикла без оптимизации последовательности машинных команд

**4 шага цикла преобразованы в один**

|  |  |  |
| --- | --- | --- |
| 1 | Loop: LD F1, 0(R1) | Загрузка значения из вектора A в F1 |
| 2 | LD F2, 0(R2) | Загрузка значения из вектора B в F2 |
| 3 | приостановка | (1 такт) Загрузка > АЛУ |
| 4 | MULTD F3, F1, F2 | Умножение значений векторов A и B. Результат в F3 |
| 5 | приостановка | (2 такта) АЛУ > Запись |
| 6 | приостановка |  |
| 7 | SD 0(R3), F3 | Запись результата |
| 8 | LD F4, -8(R1) | Загрузка значения из вектора A в F4 |
| 9 | LD F5, -8(R2) | Загрузка значения из вектора B в F5 |
| 10 | приостановка | (1 такт) Загрузка > АЛУ |
| 11 | MULTD F6, F4, F2 | Умножение значений векторов A и B. Результат в F6 |
| 12 | приостановка | (2 такта) АЛУ > Запись |
| 13 | приостановка |  |
| 14 | SD -8(R3), F6 | Запись результата |
| 15 | LD F7, -16(R1) | Загрузка значения из вектора A в F7 |
| 16 | LD F8, -16(R2) | Загрузка значения из вектора B в F8 |
| 17 | приостановка | (1 такт) Загрузка > АЛУ |
| 18 | MULTD F9, F7, F8 | Умножение значений векторов A и B. Результат в F9 |
| 19 | приостановка | (2 такта) АЛУ > Запись |
| 20 | приостановка |  |
| 21 | SD -16(R3), F9 | Запись результата |
| 22 | LD F10, -24(R1) | Загрузка значения из вектора A в F10 |
| 23 | LD F11, -24(R2) | Загрузка значения из вектора B в F11 |
| 24 | приостановка | (1 такт) Загрузка > АЛУ |
| 25 | MULTD F12, F11, F10 | Умножение значений векторов A и B. Результат в F12 |
| 26 | приостановка | (2 такта) АЛУ > Запись |
| 27 | приостановка |  |
| 28 | SD -24(R3), F12 | Запись результата |
| 29 | SUBI R1, R1, #32 | Пересчет указателя для вектора A |
| 30 | SUBI R2, R2, #32 | Пересчет указателя для вектора B |
| 31 | SUBI R3, R3, #32 | Пересчет указателя для вектора C |
| 32 | BNEZ R1, Loop | Переход к началу цикла, если R1 не равно нулю |
| 33 | приостановка | (1 такт) Переход |

Данный цикл выполняется за 33 такта или по 8,25 тактов на каждую операцию.

Число машинных тактов для этого случая составляет: **8,25\*n**

**6 шагов цикла преобразованы в один**

|  |  |  |
| --- | --- | --- |
| 1 | Loop: LD F1, 0(R1) | Загрузка значения из вектора A в F1 |
| 2 | LD F2, 0(R2) | Загрузка значения из вектора B в F2 |
| 3 | приостановка | (1 такт) Загрузка > АЛУ |
| 4 | MULTD F3, F1, F2 | Умножение значений векторов A и B. Результат в F3 |
| 5 | приостановка | (2 такта) АЛУ > Запись |
| 6 | приостановка |  |
| 7 | SD 0(R3), F3 | Запись результата |
| 8 | LD F4, -8(R1) | Загрузка значения из вектора A в F4 |
| 9 | LD F5, -8(R2) | Загрузка значения из вектора B в F5 |
| 10 | приостановка | (1 такт) Загрузка > АЛУ |
| 11 | MULTD F6, F4, F2 | Умножение значений векторов A и B. Результат в F6 |
| 12 | приостановка | (2 такта) АЛУ > Запись |
| 13 | приостановка |  |
| 14 | SD -8(R3), F6 | Запись результата |
| 15 | LD F7, -16(R1) | Загрузка значения из вектора A в F7 |
| 16 | LD F8, -16(R2) | Загрузка значения из вектора B в F8 |
| 17 | приостановка | (1 такт) Загрузка > АЛУ |
| 18 | MULTD F9, F7, F8 | Умножение значений векторов A и B. Результат в F9 |
| 19 | приостановка | (2 такта) АЛУ > Запись |
| 20 | приостановка |  |
| 21 | SD -16(R3), F9 | Запись результата |
| 22 | LD F10, -24(R1) | Загрузка значения из вектора A в F10 |
| 23 | LD F11, -24(R2) | Загрузка значения из вектора B в F11 |
| 24 | приостановка | (1 такт) Загрузка > АЛУ |
| 25 | MULTD F12, F10, F11 | Умножение значений векторов A и B. Результат в F12 |
| 26 | приостановка | (2 такта) АЛУ > Запись |
| 27 | приостановка |  |
| 28 | SD -24(R3), F12 | Запись результата |
| 29 | LD F13, -32(R1) | Загрузка значения из вектора A в F13 |
| 30 | LD F14, -32(R2) | Загрузка значения из вектора B в F14 |
| 31 | приостановка | (1 такт) Загрузка > АЛУ |
| 32 | MULTD F15, F13, F14 | Умножение значений векторов A и B. Результат в F15 |
| 33 | приостановка | (2 такта) АЛУ > Запись |
| 34 | приостановка |  |
| 35 | SD -32(R3), F15 | Запись результата |
| 36 | LD F16, -40(R1) | Загрузка значения из вектора A в F16 |
| 37 | LD F17, -40(R2) | Загрузка значения из вектора B в F17 |
| 38 | приостановка | (1 такт) Загрузка > АЛУ |
| 39 | MULTD F18, F16, F17 | Умножение значений векторов A и B. Результат в F18 |
| 40 | приостановка | (2 такта) АЛУ > Запись |
| 41 | приостановка |  |
| 42 | SD -40(R3), F18 | Запись результата |
| 43 | SUBI R1, R1, #48 | Пересчет указателя для вектора A |
| 44 | SUBI R2, R2, #48 | Пересчет указателя для вектора B |
| 45 | SUBI R3, R3, #48 | Пересчет указателя для вектора C |
| 46 | BNEZ R1, Loop | Переход к началу цикла, если R1 не равно нулю |
| 47 | приостановка | (1 такт) Переход |

Данный цикл выполняется за 47 тактов или по 7.83 тактов на каждую операцию.

Число машинных тактов для этого случая составляет: **7.83\*n**

1. Скалярный процессор с простым конвейером + Разворачивание цикла с оптимизацией последовательности машинных команд

**4 шага цикла преобразованы в один**

|  |  |  |
| --- | --- | --- |
| 1 | Loop: LD F1, 0(R1) | Загрузка значения из вектора A в F1 |
| 2 | LD F2, 0(R2) | Загрузка значения из вектора B в F2 |
| 3 | LD F4, -8(R1) | Загрузка значения из вектора A в F4 |
| 4 | LD F5, -8(R2) | Загрузка значения из вектора B в F5 |
| 5 | LD F7, -16(R1) | Загрузка значения из вектора A в F7 |
| 6 | LD F8, -16(R2) | Загрузка значения из вектора B в F8 |
| 7 | LD F10, -24(R1) | Загрузка значения из вектора A в F10 |
| 8 | LD F11, -24(R2) | Загрузка значения из вектора B в F11 |
| 9 | MULTD F3, F1, F2 | Умножение значений векторов A и B. Результат в F3 |
| 10 | MULTD F6, F4, F2 | Умножение значений векторов A и B. Результат в F6 |
| 11 | MULTD F9, F7, F8 | Умножение значений векторов A и B. Результат в F9 |
| 12 | MULTD F12, F10, F11 | Умножение значений векторов A и B. Результат в F12 |
| 13 | SD 0(R3), F3 | Запись результата |
| 14 | SD -8(R3), F6 | Запись результата |
| 15 | SD -16(R3), F9 | Запись результата |
| 16 | SUBI R1, R1, #24 | Пересчет указателя для вектора A |
| 17 | SUBI R2, R2, #24 | Пересчет указателя для вектора B |
| 18 | SUBI R3, R3, #24 | Пересчет указателя для вектора C |
| 19 | BNEZ R1, Loop | Переход к началу цикла, если R1 не равно нулю |
| 20 | SD 8(R3), F12 | Запись результата |

Данный цикл выполняется за 20 тактов или по 5 тактов на каждую операцию.

Число машинных тактов для этого случая составляет: **5\*n**

**6 шагов цикла преобразованы в один**

|  |  |  |
| --- | --- | --- |
| 1 | Loop: LD F1, 0(R1) | Загрузка значения из вектора A в F1 |
| 2 | LD F2, 0(R2) | Загрузка значения из вектора B в F2 |
| 3 | LD F4, -8(R1) | Загрузка значения из вектора A в F4 |
| 4 | LD F5, -8(R2) | Загрузка значения из вектора B в F5 |
| 5 | LD F7, -16(R1) | Загрузка значения из вектора A в F7 |
| 6 | LD F8, -16(R2) | Загрузка значения из вектора B в F8 |
| 7 | LD F10, -24(R1) | Загрузка значения из вектора A в F10 |
| 8 | LD F11, -24(R2) | Загрузка значения из вектора B в F11 |
| 9 | LD F13, -32(R1) | Загрузка значения из вектора A в F13 |
| 10 | LD F14, -32(R2) | Загрузка значения из вектора B в F14 |
| 11 | LD F16, -40(R1) | Загрузка значения из вектора A в F16 |
| 12 | LD F17, -40(R2) | Загрузка значения из вектора B в F17 |
| 13 | MULTD F3, F1, F2 | Умножение значений векторов A и B. Результат в F3 |
| 14 | MULTD F6, F4, F2 | Умножение значений векторов A и B. Результат в F6 |
| 15 | MULTD F9, F7, F8 | Умножение значений векторов A и B. Результат в F9 |
| 16 | MULTD F12, F10, F11 | Умножение значений векторов A и B. Результат в F12 |
| 17 | MULTD F15, F13, F14 | Умножение значений векторов A и B. Результат в F15 |
| 18 | MULTD F18, F16, F17 | Умножение значений векторов A и B. Результат в F18 |
| 19 | SD 0(R3), F3 | Запись результата |
| 20 | SD -8(R3), F6 | Запись результата |
| 21 | SD -16(R3), F9 | Запись результата |
| 22 | SD -24(R3), F12 | Запись результата |
| 23 | SD -32(R3), F15 | Запись результата |
| 24 | SUBI R1, R1, #48 | Пересчет указателя для вектора A |
| 25 | SUBI R2, R2, #48 | Пересчет указателя для вектора B |
| 26 | SUBI R3, R3, #48 | Пересчет указателя для вектора C |
| 27 | BNEZ R1, Loop | Переход к началу цикла, если R1 не равно нулю |
| 28 | SD 8(R3), F18 | Запись результата |

Данный цикл выполняется за 28 тактов или по 4,67 тактов на каждую операцию.

Число машинных тактов для этого случая составляет: **4,67\*n**

Сравнение реализаций

* Скалярный процессор с простым конвейером без планирования (с учетом задержек): 12 МТ на элемент.
* Скалярный процессор с простым конвейером + разворачивание цикла без оптимизации последовательности машинных команд:  
  4 шага на цикл: 8.25 МТ на элемент.

6 шагов на цикл: 7.83 МТ на элемент.

* Скалярный процессор с простым конвейером + разворачивание цикла с оптимизацией последовательности машинных команд:

4 шага на цикл: 5 МТ на элемент.

6 шагов на цикл: 4.67 МТ на элемент.

Вывод: наиболее эффективным оказывается разворачивание цикла с оптимизацией последовательности команд (разворачивание цикла выявляет больше вычислений, которые могут быть оптимизированы для минимизации приостановок конвейера), причем чем больше шагов на цикл, тем меньше приходится МТ на элемент (предполагаем, что количество шагов кратно длинам векторов).

Вывод:

Чем больше шагов цикла развернуты в один, те более эффективна оптимизация. Разворачивание циклов представляет собой простой, но полезный метод увеличения размера линейного кодового фрагмента, который может быть эффективно оптимизирован.