**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

**ВАРИАНТ 12**

**Цель работы:** проверить работу различных алгоритмов при различных режимах записи.

**1 Принципы построения кэш-памяти**

|  |  |
| --- | --- |
| Режимы записи | Алгоритм замещения |
| Сквозная | О, без учета бита записи |
| Обратная | СЗ, с учетом бита записи |

Текст программы:

jmp 006

rd #76

wr 14

wr r2

push r2

ret

call 001

Таблица 1 – Состояние кэш-памяти при размере 4 и сквозном режиме записи

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Команда | № ячейки | Тэг | Данные | Z | U | W |
| Jmp 006 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 000 | 000000 | 0 | 0 | 0 |
| 2 | 000 | 000000 | 0 | 0 | 0 |
| 3 | 000 | 000000 | 0 | 0 | 0 |
| Call 0001 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 1 | 0 |
| 3 | 000 | 000000 | 0 | 0 | 0 |
| Rd #76 | 0 | 000 | 100006 | 1 | 0 | 0 |
| 1 | 006 | 190001 | 1 | 0 | 0 |
| 2 | 999 | 000007 | 1 | 0 | 0 |
| 3 | 001 | 211076 | 1 | 0 | 0 |
| Wr 14 | 0 | 014 | 000076 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 0 | 0 |
| 2 | 999 | 000007 | 1 | 0 | 0 |
| 3 | 002 | 220014 | 1 | 1 | 0 |
| Wr r2 | 0 | 014 | 000076 | 1 | 1 | 0 |
| 1 | 003 | 320002 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 0 | 0 |
| 3 | 002 | 220014 | 1 | 0 | 0 |
| Push r2 | 0 | 998 | 000076 | 1 | 0 | 0 |
| 1 | 003 | 320002 | 1 | 0 | 0 |
| 2 | 004 | 060002 | 1 | 0 | 0 |
| 3 | 998 | 000076 | 1 | 1 | 0 |
| ret | 0 | 005 | 080000 | 1 | 1 | 0 |
| 1 | 003 | 320002 | 1 | 0 | 0 |
| 2 | 004 | 060002 | 1 | 0 | 0 |
| 3 | 998 | 000076 | 1 | 0 | 0 |

Команда записи WR R2 в режиме микрокоманд:

00 MAR := PC

01 MRd

02 CR := MDR – произошло изменение кэш-памяти

03 PC := PC+1

04 RAR := CR5

05 RDR := Acc

06 RWr

07 END\_COMMAND

Таблица 2 – Состояние кэш-памяти при размере 8 и обратном режиме записи

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Команда | № ячейки | Тэг | Данные | Z | U | W |
| Jmp 006 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 000 |  | 0 | 0 | 0 |
| 2 | 000 |  | 0 | 0 | 0 |
| 3 | 000 |  | 0 | 0 | 0 |
| 4 | 000 |  | 0 | 0 | 0 |
| 5 | 000 |  | 0 | 0 | 0 |
| 6 | 000 |  | 0 | 0 | 0 |
| 7 | 000 |  | 0 | 0 | 0 |
| Call 001 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 1 | 0 |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| Rd #076 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 1 | 0 |
| 3 | 001 | 211076 | 1 | 1 | 0 |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| Wr 14 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 1 | 0 |
| 3 | 001 | 211076 | 1 | 1 | 0 |
| 4 | 002 | 220014 | 1 | 1 | 0 |
| 5 | 014 | 000076 | 1 | 1 | 0 |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| Wr r2 | 0 | 000 | 100006 | 1 | 1 | 0 |
| 1 | 006 | 190001 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 1 | 0 |
| 3 | 001 | 211076 | 1 | 1 | 0 |
| 4 | 002 | 220014 | 1 | 1 | 0 |
| 5 | 014 | 000076 | 1 | 1 | 0 |
| 6 | 003 | 320002 | 1 | 1 | 0 |
| **7** |  |  |  |  |  |
| Push r2 | 0 | 000 | 100006 | 1 | 0 | 0 |
| 1 | 998 | 000076 | 1 | 1 | 0 |
| 2 | 999 | 000007 | 1 | 0 | 0 |
| 3 | 001 | 211076 | 1 | 0 | 0 |
| 4 | 002 | 220014 | 1 | 0 | 0 |
| 5 | 014 | 000076 | 1 | 0 | 0 |
| 6 | 003 | 320002 | 1 | 0 | 0 |
| **7** | 004 | 060022 | 1 | 0 | 0 |
| ret | 0 | 000 | 100006 | 1 | 0 | 0 |
| 1 | 998 | 000076 | 1 | 1 | 0 |
| 2 | 005 | 080000 | 1 | 1 | 0 |
| 3 | 001 | 211076 | 1 | 0 | 0 |
| 4 | 002 | 220014 | 1 | 0 | 0 |
| 5 | 014 | 000076 | 1 | 0 | 0 |
| 6 | 003 | 320002 | 1 | 0 | 0 |
| **7** | 004 | 060022 | 1 | 0 | 0 |

**ЧАСТЬ 2. АЛГОРИТМЫ ЗАМЕЩЕНИЯ СТРОК КЭШ-ПАМЯТИ**

*Задание*

В данной части расчетно-графической работы все варианты задания одинаковы: исследовать эффективность работы кэш-памяти при выполнении двух разнотипных программ, написанных и отлаженных вами при выполнении лабораторной работы № 2 и 3.

Таблица 2.1 – Результаты моделирования лабораторной работы №2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Способ | Сквозная запись | | | | | |
| Алгоритм | Случайное замещение | | Очередь | | Бит использования | |
| Размер | Без W | c W | Без W | c W | Без W | c W |
| 4 | 0.025 | 0.075 | 0.075 | 0.025 | 0.025 | 0.025 |
| 8 | 0.125 | 0.175 | 0.175 | 0.1 | 0.15 | 0.125 |
| 16 | 0.2 | 0.175 | 0.3 | 0.125 | 0.175 | 0.2 |
| 32 | 0.2 | 0.2 | 0.5 | 0.25 | 0.2 | 0.2 |
| Способ | Обратная запись | | | | | |
| Алгоритм | Случайное замещение | | Очередь | | Бит использования | |
| Размер | Без W | c W | Без W | c W | Без W | c W |
| 4 | 0.05 | 0.075 | 0.05 | 0.075 | 0.025 | 0.025 |
| 8 | 0.1 | 0.1 | 0.125 | 0.1 | 0.15 | 0.15 |
| 16 | 0.1 | 0.175 | 0.1 | 0.2 | 0.175 | 0.175 |
| 32 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |

Таблица 2.2 – Результаты моделирования лабораторной работы №3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Способ | Сквозная запись | | | | | |
| Алгоритм | Случайное замещение | | Очередь | | Бит использования | |
| Размер | Без W | c W | Без W | c W | Без W | c W |
| 4 | 0.076 | 0.092 | 0.11 | 0.11 | 0.125 | 0.092 |
| 8 | 0.289 | 0.3 | 0.278 | 0.14 | 0.169 | 0.169 |
| 16 | 0.57 | 0.66 | 0.7 | 0.68 | 0.81 | 0.786 |
| 32 | 0.834 | 0.81 | 0.82 | 0.82 | 0.82 | 0.82 |
| Способ | Обратная запись | | | | | |
| Алгоритм | Случайное замещение | | Очередь | | Бит использования | |
| Размер | Без W | c W | Без W | c W | Без W | c W |
| 4 | 0.009 | 0.075 | 0 | 0.19 | 0.0 | 0.004 |
| 8 | 0.17 | 0.31 | 0.027 | 0.27 | 0.077 | 0.13 |
| 16 | 0.65 | 0.59 | 0.59 | 0.63 | 0. 65 | 0.64 |
| 32 | 0.82 | 0.87 | 0.84 | 0.83 | 0.83 | 0.79 |

**ВЫВОДЫ**

В ходе выполнения расчётно-графической работы были протестированы различные алгоритмы замещения и режимы записи кэш-памяти.

Кэш-память повышает эффективность при работе с повторяющимися участками программы. Эффективность работы кэш-памяти оценивается числом кэш-попаданий по отношению к общему числу обращений к памяти. Так же эффективность работы системы с кэш-памятью зависит от класса решаемой задачи.

Для выявления этих особенностей кэш-памяти была рассчитана эффективность ее работы в лабораторных работах №2 и №3.

В третьей лабораторной работе основная часть программ состоит из циклов и вызовов процедур, из-за чего коэффициент эффективности выше, чем во второй лабораторной. Из полученных данных видно, что программы использующие большое количество циклов и вызовов подпрограмм лучше работают с алгоритмами случайного замещения, а линейные программы имеют большую производительность с алгоритмами типа очереди. Также программы, имеющие большое число повторяющихся участков (часто вызываемых подпрограмм и/или циклов) при прочих равных условиях обеспечат более высокую эффективность применения кэш-памяти, чем линейные программы.

Также весьма существенно на производительность кэш-памяти влияет ее размер, из проведенных тестов видно, что при большом кэше программа и все ее данные помещаются в кэше и нет необходимости обращаться к ОЗУ, что существенно увеличивает производительность.