|  |
| --- |
| *Министерство образование и науки*  *Российской федерации*  Федеральное агентство по образованию  Курганский государственный университет |
| Кафедра «Безопасность информационных и автоматизированных систем» |
| Лабораторная работа № 3:  **«Измерение времени выполнения команд (RDTSC)»** |
| Выполнил \_\_\_\_ Бутенко А.Ю.  Принял \_\_\_\_\_\_ Стукало В.А. |
|  |
| Курган 2016 |

Цель работы:

- углубление знаний программно-аппаратной организации IBM PC и

ее службы времени;

- измерение времени выполнения команды;

- закрепление навыков программирования на языках Assembler.

**Ход работы:**

Основная задача состоит в том, чтобы измерить время исполнения операции. Для этого необходимо исключить издержки на поддержание цикла и повысить точно измерения (увеличить количество итераций).

Для измерения времени воспользуемся функцией rdtsc, которая возвращает количество тактов с последнего сброса процессора.

1. **RDTSC**
2. **mov** **dword[bp**-200**]**,**edx** *; записываем текущие значения количества тиков из rdtsc*
3. **mov** **dword[bp**-204**]**,**eax**
4. clr **ecx**
5. floop:
6. **inc** **ecx**
7. *; add ebx, ebx*
8. **cmp** **ecx**, 0x5f5e100
9. **jb** floop
11. **RDTSC**
12. **sub** **eax**,**dword[bp**-204**]** *; вычитаем старые значения из новых*
13. **sbb** **edx**,**dword[bp**-200**]**
14. **mov** **dword[bp**-200**]**,**edx** *; записываем их, для дальнейшего использования*
15. **mov** **dword[bp**-2

Программа состоит из двух одинаковых частей, в одной из них измеряется количество тиков для пустого цикла, в другой для цикла с командой.

Вторая часть выглядит вот так:

1. **RDTSC**
2. **mov** **dword[bp**-208**]**,**edx** *; записываем текущие значения количества тиков из rdtsc для второго цикла*
3. **mov** **dword[bp**-212**]**,**eax**
4. clr **ecx**
5. sloop:
6. **inc** **ecx**
7. **add** **ebx**, **ebx**
8. **cmp** **ecx**, 0x5f5e100
9. **jb** sloop
11. **RDTSC**
12. **sub** **eax**,**dword[bp**-212**]** *; вычитаем старые значения из новых*
13. **sbb** **edx**,**dword[bp**-208**]**

И последняя часть программы заключается в вычислении одного результата из другого, что требуется для исключения из замера издержек цикла.

1. **sub** **eax**,**dword[bp**-204**]** *; вычитаем результаты второго замера из результатов первого*
2. **sbb** **edx**,**dword[bp**-200**]**
3. **push** **eax**
4. **mov** **eax**,**edx**
5. print\_eax *; выводим на экран*
6. **pop** **eax**
7. print\_eax

Для чистоты эксперимента запустим программу, реализующую вышеописанный функционал 5 раз.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| CNT (n) | 100002448 | 100002448 | 100002448 | 100002448 | 100002448 |

Из измерения видно, что время работы одной команды сложения занимает 100002448 тиков.

Для того, чтобы сравнить с результатами прошлой лабораторной, в которой мы также замеряли время работы команды, необходимо перевести количество тиков в время.

В таблице записаны исходные данные для 100 миллионов итераций. Для измерения времени потребуется вычислить период выполнения одного тика в процессоре с заданной частотой (75 MHz).

Этот период равен с

Время выполнения 100 миллионов операций с

Теперь для сравнений времени выполнения одной операции из прошлой лабораторной (13 нс) необходимо вычислить время выполнения этой операции здесь, и перевести число в наносекунды.

Погрешность измерения связана с особенностями вычисления в первой лабораторной.

**Вывод:** мной было измерено время работы одной команды. Результат – 13.33 наносекунд, что равно результату предыдущей лабораторной с измерением времени работы команды другим способом.