

Изучение системы выполнения инструкций процессоров Intel

Выполнил:

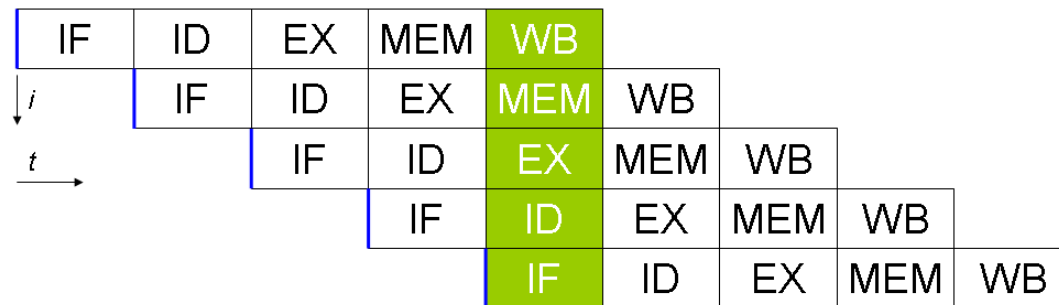
Студент 243 гр. Чернявский А.А.

Научный руководитель:

Баклановский М.В.

Эволюция Intel

- Конвейерная обработка
- Многоуровневое кэширование
- Внеочередное выполнение
- Механизмы предугадывания процессов
- Результат – прогнозируемое ВВИ



Задача – изучение системы выполнения инструкций.

Инструментарий

T32.exe

Аргументы: t32 [SER] [SV] N [C1] [J C2]

SER – барьер: n|m|x|mx; n – none, m – mfence, x = cpuid

SV – хранение rdtsc: m|p|x; m – memory, p – stack, x – mmx-registers

N – количество измерений

C1, C2 – мнемоники ассемблерных команд

J – количество повторений команды

Действия: ...SER, rdtsc, SC, SER Code SER, rdtsc, result... - N times

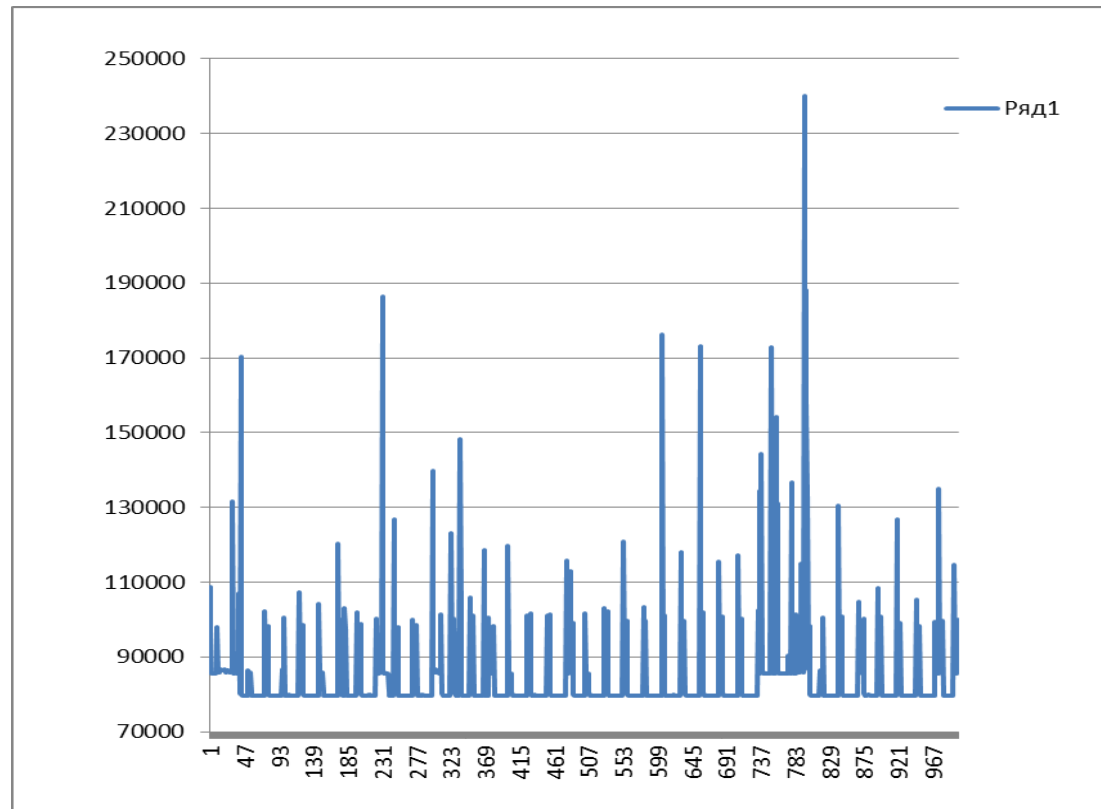
А также: Batch files, Perl, Excel

Проверка неустойчивости

1000 замеров

100 тыс. выполнений

Inc eax



Противоречивые результаты:

- 1) Неустойчивость, разброс.
- 2) Период, «опора»

Способы получения ВВИ

Идеи:

1) ввести понятие **среднего ВВИ**

$$t = \text{result}/J$$

2) статистическое распределение внутри цикла измерений:

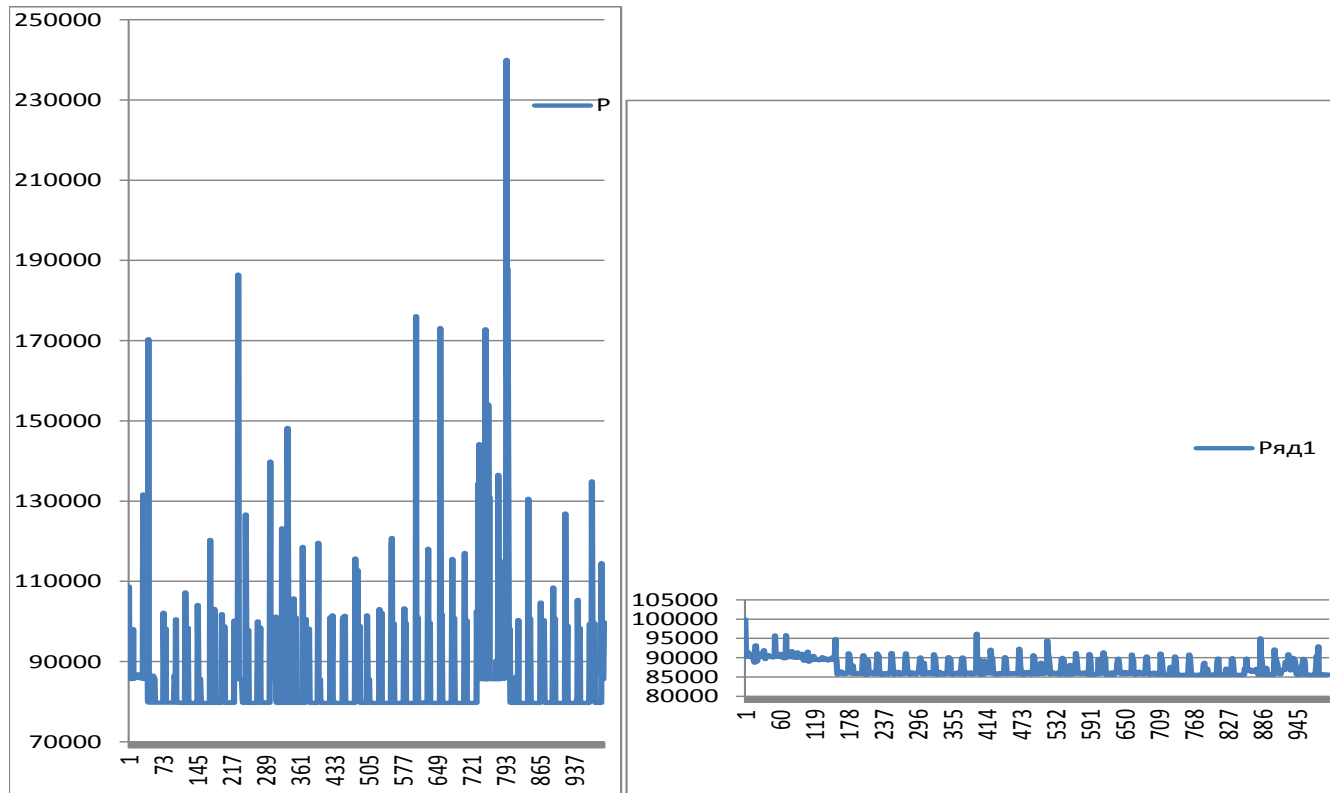
Принять за ВВИ наиболее часто встречаемое значение (диапазон)

Вывод: способы не подошли из-за нестабильности ответов.

Гипотеза: на результаты измерений влияет состояние ПК:

Проверка гипотезы

Выполнение одной и той же последовательности замеров:



Обычный и «облегчённый» режим работы ПК.

Вывод – необходимость учёта действующего состояния ПК.

Способ измерения с учётом действующего состояния ПК:

Идеи:

- 1) нейтрализация влияния внешних факторов
- 2) выбор минимума среди измерений

«Прогон барьеров»

Формула:

\bar{t} – время «прогона барьеров» (без инструкций)

t – результат t_{32} (с инструкциями)

n – количество выполнений в памяти (между барьерами)

$$t_i = (t - \bar{t}) / n$$

t, \bar{t} – берутся минимальными на основе N измерений

$N := 1000$

Результаты:

Неадекватность:

>2; 0; -0.21

Нестабильность.

Предположение:

А минимальны ли «минимумы»? Достаточно ли 1000 измерений для выявления минимума?

Проверка: минимальное ВВИ для барьера μ на основе разного количества измерений.

Кол-во замеров	1 000	10 000	100 000	500 000	1 000 000
Результат	452	396	377	376	376

Вывод – оптимальнее считать μ на 100 тыс. измерениях.

Результаты с учётом изменения:

Стабильно:

Instruction\Source	official manual	i5 2410M	i7 2630QM
inc eax	0.33	0.69	0.69
dec eax	0.33	0.69	0.69
add eax 1	0.33	0.69	0.69
sub eax 1	0.33	0.69	0.69
xor ax ax	0.33	0.69	0.69
nop	0.25	0.28	0.28

Выводы: реальные ВВИ действительно отличаются от заявленных в сторону возрастания.

Перспективы дальнейшей деятельности:

Всё, что было ранее – «предельные» случаи.

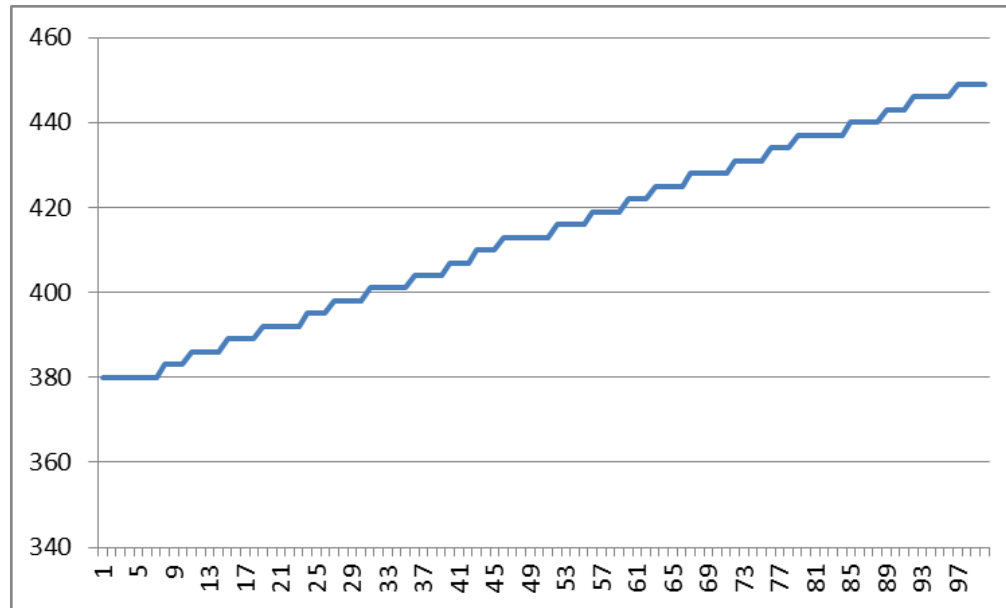
Практика – случаи «одиночных» взаимодействий.

Востребованность и перспектива.

Почему перспектива там?

Пример 1

Зависимость минимального значения от количества повторов в памяти:



Ступенчатость с шагом 3 такта. Переменный период.

Пример 2

Многократный замер сочетаний произвольных команд:

43 - inc eax; 40 – inc ebx;

43 n 40: inc eax, inc ebx, ..., inc ebx <- n раз

opcode	<empty>	43	40	43 1 40	43 2 40	43 3 40	43 10 40	43 100 40
result	377	380	570	567	496	469	502	579

Объяснение: использование различных оптимизирующих технологий.

Таким образом, дальнейшее направление работы следует вести в случаях «одиночных» взаимодействий

Выводы

- 1) Показатели ВВИ напрямую зависят от «загруженности» ПК, что следует учитывать при измерении.
- 2) Один из способов сделать это – работать с минимальными значениями и измерять «прогон барьеров»
- 3) Реальные ВВИ не соответствуют заявленным Intel
- 4) В реальных условиях поведение команд в большей степени определяется особенностями их взаимодействия