

Лабораторна робота № 3

Дослідження продуктивності та пропускну здатності підсистеми пам'яті за допомогою програми Cache Burst 32

Мета роботи: ознайомитись з програмою Cache Burst 32, дослідити з її допомогою основні параметри оперативної пам'яті та кеш-пам'яті, навчитися визначати тип процесора за результатами роботи тестової програми.

Загальна інформація. Відомо, що всі комірки пам'яті DDR, DDR II, DDR III мають одну і ту ж будову. Всі вони є представниками динамічної пам'яті (DRAM або Dynamic RAM). RAM - Random Access Memory – пам'ять з довільним доступом. Всі комірки DRAM складаються з одного транзистора і одного конденсатора. Це найдешевший спосіб виробництва комірок пам'яті. Стан конденсатора визначає, "0" чи "1" зберігається у комірці, але сама наявність конденсатора є причиною деяких обмежень DRAM. Заряджений конденсатор еквівалентний логічній "1", розряджений – логічному "0". Однак з часом конденсатор втрачає заряд, і тому необхідно час від часу його оновлювати. Потрібний для цього струм дуже малий, тому необхідно небагато часу, щоб конденсатор малої ємності (0,12-0,18 мікрофарад) був заряджений знову. Але під час цього процесу до комірки пам'яті звертатися не можна. Виробники DRAM говорять, що подібне оновлення повинно проводитися кожні 64 мс, тобто комірка пам'яті недоступна менш, ніж 1% всього часу. Але найбільша проблема із DRAM в тому, що при операції зчитування з комірки конденсатор втрачає свій заряд, тобто читання є деструктивним, і комірка після зчитування інформації повинна бути поновлена. Таким чином, кожен раз при зчитуванні повинен проводитись і запис. Іншими словами комірка пам'яті оновлюється кожні 64 мс, коли з неї здійснюється зчитування. В результаті збільшується час циклічного доступу, збільшується латентність.

Поняття латентності пам'яті: як видно з вищесказаного, пам'ять працює з меншою частотою, ніж CPU (внаслідок оновлення пам'яті кожні 64 мс, перезарядки конденсаторів та інших технологічних особливостей). Внаслідок цього відбуваються затримки в передачі даних від пам'яті до процесора. Процесор

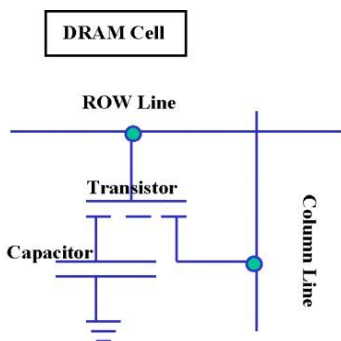
«змушений» деякий час простоювати. Наявність кеш-пам'яті L_1 та L_2 в деякій мірі згладжує простої в роботі процесора і збільшує швидкість передачі даних від оперативної пам'яті до CPU (оскільки частота роботи кешей більша, ніж частота роботи оперативної пам'яті). Однак розмір кешей обмежений і тому процесор все одно простоює декілька циклів своєї роботи в очікуванні даних. Таким чином, іншими словами латентність – це простій в роботі процесора (вимірюється в циклах). Отже, оптимальна оперативна пам'ять – недорога пам'ять з низькою латентністю та високою пропускнуою здатністю.

В модулях статичної пам'яті (SRAM або Static RAM) така проблема відсутня. Одна комірка SRAM складається з 4 транзисторів і 2 резисторів. Комірки SRAM зберігають дані не шляхом ємнісної зарядки (як комірки DRAM), а шляхом перемикання транзисторів в необхідний стан, подібно транзисторам в CPU. Зчитування інформації з комірки SRAM не деструктивно, отже, оновлення комірки SRAM не потребує.

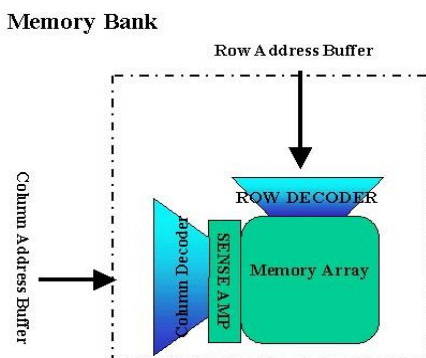
SRAM може працювати на більш високих частотах, ніж DRAM, але найважливішою її перевагою є більш низька латентність при отриманні перших 8 байт (машинне слово) даних. SRAM необхідно біля 2-3 циклів для отримання необхідних даних, в той час як DRAM потребує від 3 до 9 циклів для накопичення необхідних даних у вихідних буферах. SRAM, яка потребує в 4 рази більшої кількості транзисторів, ніж DRAM, має значно дорожчу собівартість при виробництві. Таким чином з одного боку модулі SRAM мають дуже низьку латентність і можуть працювати при високих частотах, а з іншої – вони приблизно у 8 разів дорожчі за DRAM.

Чому латентність DRAM при отриманні першого машинного слова може коливатися від 3 до 9 циклів? Щоб зрозуміти це, необхідно розглянути внутрішню будову комірки пам'яті DRAM.

Структура і функціонування комірок пам'яті. Найменша складова мікросхеми пам'яті – це комірка пам'яті. Кожна комірка представляє один біт і має унікальну адресу, утворену перетином строки і стовпця. Вісім біт представляють собою один байт, що дозволяє скласти 256 можливих комбінацій. 28 байт – найменша адресовувана частина пам'яті. Комірка пам'яті не може адресуватися індивідуально, занадто велика кількість адрес була б задіяна. Тому адресація ведеться за номерами строки або стовпчика, тобто комірки об'єднані в матричну структуру. А ці структури в свою чергу об'єднуються в банки пам'яті.



Більшість чипсетів підтримують до 4 таких банків, і в кожному SDRAM DIMM (Dual Inline Memory Module) може бути 8 або 16 чипів. Кожний SDRAM DIMM має 14 адресних рядків та 64-х бітну адресу в рядку.



Розглянемо банк пам'яті. Як видно з рисунку, показаному вище, в кожному банку пам'яті є чутливий підсилювач (sense amp) та перетворювач рядків і стовпців (row decoder, column decoder). Для розуміння роботи пам'яті "зсередини" необхідно розглянути, що відбувається при запиті даних процесором в ситуації відсутності цих даних в кеші.

- Процесор запитує 32 байта і надсилає запит чипсету. На це витрачається 1 цикл.
- Чипсет надсилає запит через 14 адресних ліній. Тобто запит надсилається всім чипам на даному DIMM. Всі рядки, що мають необхідну адресу, називають сторінками. Іншими словами, коли чипсет надсилає адресу рядка в мікросхемі DIMM, він "відкриває" сторінку в DIMM.

- В кожному банку пам'яті є sense amps (amps = amplifiers, підсилювач), який підтримує заряд в комірках (біти), який повинен бути прочитаний або записаний в комірки. Sense amp прочитає рядок у відповідності з переданою адресою. Цей процес займає деякий час, що називається RAS to CAS delay. В залежності від якості SDRAM, RAS to CAS delay може потребувати 2-3 циклів.

- Таким чином, у нас є необхідний рядок в sense amps, але до сих пір немає необхідних комірок. Латентність CAS – це час для отримання необхідного стовпця в даному рядку (запитувану комірку з інформацією), рівний 2 або 3 циклам.

- Уміст комірок пам'яті надсилається на буфер виводу та чипсет, і нарешті його можна прочитати.

- Тепер, коли є перше слово (8 байт) і необхідний рядок знаходиться в sense amps, отримання наступних 24 байт виглядає просто. Внутрішній лічильник збільшується кожен раз, коли чип пам'яті надсилає ряд комірок пам'яті із наступного стовпця у зовнішній буфер. Кожен такт 8 байт (одне слово) пересилається у зовнішній буфер. Це так званий "монопольний режим" (burst mode).

- Як правило латентність пам'яті в цілому (від FSB до DRAM) дорівнює сумі:

- Латентності між FSB і чипсетом (+/- 1 цикл)
- Латентності між чипсетом і DRAM (+/- 1 цикл)
- Латентності RAS to CAS (2-3 цикла, в залежності від рядка)
- Латентності CAS (2-3 цикла, в залежності від стовпця)
- 1 цикла на перенос даних
- Латентності, необхідної для отримання даних із зовнішнього буфера DRAM і передачі процесору (через чипсет) (+/- 2 цикла).

Все вищезазначене дозволить отримати перше слово – 8 байт. Модулі PC100 SDRAM CAS 2 мають латентність біля 9 циклів, наступні 3 цикли витрачаються на отримання наступних 24 байт. В цьому випадку PC100 SDRAM отримує 32 байта за 12 циклів. Якщо необхідно отримати латентність процесора, то слід помножити латентність пам'яті (загальну) на множник шини. Так для 500 MHz (5 x 100 MHz) процесора отримаємо латентність 5 x 9 циклів - 45 циклів. 45 циклів пройде до моменту, поки не будуть знайдені та прочитані дані, відсутні в кеші L2.

ОПИС ПРОГРАМИ CACHE BURST 32.

Опис кнопок:

Start Запускає відмічені флажками тести на виконання. У вихідному положенні - вибрані тести MMX Read/ Write та SSE Read/ Write. Якщо процесор не підтримує технології MMX і SSE, відповідні опції вибору будуть недоступні. Тест Memory Walk не може бути вибраний одночасно з будь-яким іншим тестом.

Report Створює звіт у вигляді HTML файлу і відкриває його у вашому Інтернет – браузері. Для тесту Memory Walk створюється окремий звіт. Одночасно із створенням звіту, значення графіків експортується у файл в форматі CSV (роздільники – крапка з комою) для подальшої обробки результатів у Microsoft Excel.

Exit Завершує роботу програми. Ви не можете відмінити або зупинити виконання вибраних тестів для повного їх завершення. Однак, якщо це необхідно - зніміть задачу засобами операційної системи.

Команди меню:

File

Команди цього меню повністю дублюють призначення кнопок.

Settings

Команда *Colors* дозволяє перевизначити кольори графіків. *Help*

Містить команди виклику довідки англійською або російською мовами, а також вікна *About* (про програму).

Настройки:

Start/End Block Size – дозволяє визначити початковий і кінцевий розмір блоку даних для тестування.

Read Latency Step – дозволяє задати крок тесту латентності зчитування інформації.

Memory Walk Start Step – дозволяє задати початковий розмір кроку для тестування латентності пам'яті.

Copy to Itself – стає активною у випадку вибору тестів копіювання (Copy/ MMX Copy/ SSE Copy). Із встановленим прапорцем копіювання блоку даних здійснюється в те саме місце, де знаходиться сам блок, тобто фактично вміст пам'яті не змінюється. У вихідному стані – ця опція не вибрана і копіювання здійснюється із зміщенням, рівним довжині переміщеного блоку даних. У зв'язку із майже 100 % потраплянням операцій запису в кеш в Copy to itself тестується в основному здатність

пам'яті до зчитування після запису (read around write). В цьому випадку значно краща утилізація кеш-пам'яті і тест є значно "легшим" для підсистеми пам'яті, що обумовлює більш високі результати і менший час тестування.

Log Y Axis – із встановленим прапорцем робить шкалу Y логарифмічною.

4x Iteration – із встановленим прапорцем збільшує кількість проходів у 4 рази. Це підвищує стабільність результатів, але у стільки ж разів збільшує час виконання тестів.

Опис тестів:

Визначення пропускну здатності:

Пропускна здатність визначається за тестами Read/ Write/ Copy (Зчитування/ Запис/ Копіювання), MMX Read/ MMX Write/ MMX Copy и SSE Read/ SSE Write/ SSE Copy. Тести Read/ Write/ Copy визначають пропускну здатність кеш-пам'яті та оперативної пам'яті в режимі 32 бітної передачі даних. Тести MMX Read/ MMX Write/ MMX Copy визначають пропускну здатність кеш-пам'яті та оперативної пам'яті в режимі 64 бітної передачі даних із застосуванням MMX інструкцій, а тести SSE Read/ SSE Write/ SSE Copy – в режимі 128 бітної передачі даних із застосуванням SSE інструкцій.

Алгоритми тестів пропускну здатності кеш-пам'яті та оперативної пам'яті в Cache Burst 32 оптимізовані для роботи з малими блоками даних. Це зроблено для отримання адекватних результатів в кеш-пам'яті процесора. При застосуванні інших алгоритмів можливе значне покращення показників пропускну здатності пам'яті на блоках даних, що перевищують розміри кешпам'яті.

Визначення латентності:

Латентність кеш-пам'яті та оперативної пам'яті визначається за тестами Read Latency (Латентність зчитування) та Memory Walk (прохід по пам'яті). Тест Read Latency найбільш ефективно показує латентність кеш-пам'яті і відображує латентність зчитування кешпам'яті та оперативної пам'яті для обраного кроку. Коли крок менший за розмір рядка кеш-пам'яті, ми вимірюємо час отримання критичного слова, тому для вимірювання латентності кеш-пам'яті L₁, L₂ або оперативної пам'яті необхідно вибрати крок, що дорівнює або перевищує розмір рядка кеш-пам'яті. Тест Memory

Walk слугує для визначення латентності пам'яті, а також як і тест Read Latency, визначає латентність зчитування. Memory Walk вимірює латентність при всіх кроках для блока пам'яті розміром 32 мегабайти. Побудова графіка Memory Walk займає кілька хвилин і отримується з деякими паузами, і це - нормально.

Значення латентності кеш-пам'яті та оперативної пам'яті визначаються в циклах роботи процесора.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Записати тему і мету роботи.
2. Ознайомитись і зробити короткий конспект методичних вказівок.
3. Закрити всі працюючі програми.
4. Запустити програму Cache Burst 32 і ознайомитись з роботою програми.
5. Дослідити пропускну здатність і латентності кеш-пам'яті та оперативної пам'яті.

Для цього:

а) Увімкнути опції Read, Write та Copy для 32 бітної шини (Select 32 bit Transfer) –виставити прапорці напроти опцій Read, Write та Copy. При цьому всі прапорці напроти Select 64 bit Transfer, Select 128 bit Transfer, Select Latency Test, Log Y Axis, Copy to itself, 4x Iterations зняти. Start/End Block Size (початковий і кінцевий розмір блоку даних для тестування) виставити рівними 4 kB та 16 MB відповідно. Запустити виконання програми (кнопка Start). Отримати графік залежностей пропускну здатності для зчитування, запису і копіювання даних із кеш-пам'яті та оперативної пам'яті в процесор та зобразити його в зошиті.

б) В опції Select Latency Test увімкнути параметр Read Latency, Block Size = 32. При цьому всі прапорці напроти Select 32 bit Transfer, Select 64 bit Transfer, Select 128 bit Transfer, Log Y Axis зняти. Отримати графік для латентності і занести його до звіту з лабораторної роботи.

6. Проаналізувати отримані графіки і заповнити таблицю:

Розмір кешу L_1 , кБ	
Розмір кешу L_2 , кБ	
Пікова пропускна здатність кешу L_1 , МБ/с	
Пікова пропускна здатність кешу L_2 , МБ/с	
Пікова пропускна здатність оперативної пам'яті, МБ/с	
Латентність кешу L_1 , циклів	
Латентність кешу L_2 , циклів	
Латентність оперативної пам'яті, циклів	

7. Зробити висновки по роботі.

ЗМІСТ ЗВІТУ

1. Номер, назва та мета роботи.
2. Відповіді на контрольні запитання.
3. Графік для латентності.
4. Таблиця з результатами тестування.
5. Висновки по роботі.

КОНТРОЛЬНІ ПИТАННЯ

1. Улаштування та принцип дії комірки динамічної пам'яті DRAM.
2. Улаштування та принцип дії статичної пам'яті SRAM. Причина відносно невеликого об'єму кеш-пам'яті.
3. Улаштування банку пам'яті.
4. Принцип роботи оперативної пам'яті.
5. Дати визначення латентності пам'яті.
6. Призначення налаштувань програми Cache Burst 32.
Призначення і вплив кеш-пам'яті L_1 та L_2 на швидкодію комп'ютера.