

Report 3

Ravil Zakirianov

- 1) $IPC = \text{instr} / \text{cycle}$ (Среднее количество исполняемых инструкций за один такт процессора)
 $CPI = 1 / IPC$ (Cycles per instr, обратная величина IPC)
 $\text{Performance} = 1 / \text{Time} = 1 / (N_{\{\text{instr}\}} * CPI * T_{\text{cycle}}) = IPC * f / N_{\{\text{instr}\}}$
(Производительность процессора. Среднее количество задач выполняемое за определенное время)
Dynamic power: $P = C * V^{(2)} * f$ (мощность, которую использует процессор во время исполнения задач)

2) SuperScalar processor - процессор, который поддерживает скорость выполнения, превышающую одну команду за один такт. Но простая одновременная обработка нескольких инструкций не делает архитектуру суперскалярной, поскольку конвейерные, многопроцессорные или многоядерные архитектуры также обеспечивают это, но другими методами. Параллелизм должен достигаться за счет нескольких одинаковых одновременно обрабатывающих устройств

- 3) Типы зависимости по данным:

RAW(Read after write) - одна инструкция пишет в регистр, а следующая читает. В качестве оптимизации можно использовать bypass или OOO execution

```
%r0 = add %r1, %r2
```

```
%r3 = mul %r0, %r0
```

WAR(Write after read) - одна инструкция выполняет запись в регистр, а предыдущая считывает. Возможные оптимизации: register renaming

```
%r0 = add %r1, %r2
```

```
%r1 = move C
```

WAW(Write after write) - две инструкции записывают в один и тот же регистр

```
%r0 = add %r1, %r1
```

```
%r0 = add %r2, %r2
```

Register renaming or SW optimizations

- 4) Разделение инструкции store на STA and STD:

STA – Store address - микроинструкция, вычисляющая адрес, STD – store data - микроинструкция, вычисляющая данные store. Так как между

операциями чтения и записи могут возникнуть зависимости, когда они используют одни и те же адреса, то благодаря разделению на микроинструкции снижается количество зависимостей.

5) Reordered Buffer(ROB) - Хранит инструкции в ООО порядке. Удерживает порядок инструкций при исключениях/прерываниях, возвращая процессор к верному состоянию.

Sheduler Queue - Планировщик инструкций. Планирует и отправляет инструкции на исполнения своевременно и отслеживает зависимости, чтобы сохранять порядок.

RAT(Register alias table) - таблица для мэтча между физическими и логическими регистрами, используемая при оптимизации register renaming. Таким образом, RAT решает проблемы с зависимостями и переименовывает регистры.

PRF(Physical register file) - файл с данными для переименованных регистров. Сохраняет состояния для регистров во время register renaming

Load/Store Buffer - буффер для операции load/store. Отслеживает все зависимости по данным между этими инструкциями и разрешает их.

6) $p = 20\%$ операций - branch. Accuracy_bp = 90%

Процент неверных предсказаний на операцию: $p * (1 - \text{accuracy_bp}) = 0.2 * 0.1 = 2\%$, тогда в среднем одно неверное предсказание будет на 50 инструкции -> размер ROB'a = 50 инструкций

7) Memory Disambiguation - это набор методов, используемых процессорами с неупорядоченным исполнением, которые выполняют команды доступа к памяти (загружают и сохраняют) не в программном порядке. Механизмы устранения неоднозначности в памяти обнаруживают истинные зависимости между операциями с памятью во время выполнения и позволяют процессору восстанавливать состояние, когда зависимость была нарушена. Они также устраняют ложные зависимости от памяти и обеспечивают больший параллелизм на уровне инструкций(ILP), позволяя безопасно выполнять загрузку и сохранение данных не по порядку.

8) Проблема со спекулятивным исполнением store инструкций.

Процессор может записывать данные в память до того, как будет проверено условие ветвления. Это может привести к тому, что данные будут записаны в неправильную ячейку памяти. В случае операции store вернуться к исходному состоянию невозможно, так как данные в памяти могут быть перезаписаны и утеряны

9) Store forwarding and Load Speculation in OOO

Store forwarding - оптимизация, позволяющая получать следующей операции load получать напрямую из store, избегая записи в память.

Load speculation - оптимизация, при котором процессор выполняет до инструкции load до того момента, пока адреса не были вычислены.

10) SMT

SMT - метод, позволяющий одному ядру исполнять несколько потоков с помощью разделения одного физического ядра на несколько логических. У каждого логического потока есть соответственно собственное состояние.