

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)  
ФИЗТЕХ-ШКОЛА РАДИОТЕХНИКИ И КИБЕРНЕТИКИ

## Homework 3: Q/A

Работу выполнил:  
Шурыгин Антон Алексеевич, группа М01-306

Долгопрудный, 2024

## Содержание

1	Вопрос	3
2	Вопрос	4
3	Вопрос	5
4	Вопрос	6
5	Вопрос	7
6	Вопрос	8
7	Вопрос	9
8	Вопрос	10
9	Вопрос	11
10	Вопрос	12

# 1 Вопрос

Q: Объясните смысл и напишите формулы для следующих метрик про производительности процессора IPC, CPI, Performance, Dynamic Power  
A:

1. IPC – среднее число инструкций за такт.
2. CPI – среднее число тактов на выполнение одной инструкции.
3. Performance – скорость исполнения инструкций.
4. Dynamic power – затраты энергии на перезарядку паразитных емкостей при переключении транзисторов в процессоре.

$$IPC = \frac{\text{instructions count}}{\text{clock cycles}} \quad (1)$$

$$CPI = \frac{1}{IPC} \quad (2)$$

$$\text{Performance} = \frac{1}{\text{Time}} = \frac{1}{N_{\text{instrs}} \cdot CPI \cdot T_{\text{cycle}}} = \frac{1}{N_{\text{instrs}}} \cdot IPC \cdot f \quad (3)$$

$$\text{Dynamic Power} = \frac{C_{\text{eff}} \cdot (V_{\text{dd}})^2 \cdot f \cdot \alpha}{2} \quad (4)$$

, где  $\alpha$  – частота переключения транзистора,  $f$  – тактовая частота.

## 2 Вопрос

Q: Что такое суперскалярный (superscalar) процессор?

А: Суперскалярный (superscalar) процессор - процессор, способный выполнять две и более инструкций за такт за счет нескольких одинаковых вычислительных устройств, независимо обрабатывающих инструкции одного потока.

### 3 Вопрос

**Q:** Какие типы зависимостей по данным существуют? Приведите примеры аппаратных оптимизаций, которые позволяют сократить связанные задержки или разрешить каждый тип зависимостей

**A:**

1. **True Dependencies** – истинная зависимость потока данных, при которой нынешняя инструкция зависит от результата предыдущей. Нарушение зависимости приводит к Read-After-Write hazard. Возможная оптимизация **store forwarding** – пробрасывание результата STORE напрямую в LOAD через bypass.
2. **False Dependencies** – ложная зависимость потока данных, при которой нынешняя инструкция на самом деле не нужны данные с предыдущей. Однако проблемы возникают в результате записи данных в destination регистр, который может конфликтовать и пересекаться с другими инструкциями. Нарушение приводит к Write-After-Write, Write-After-Read hazard. Данная проблема решается техникой **Register renaming**, так как физических регистров гораздо больше, чем архитектурных (логических). Соответствие физических и архитектурных регистров осуществляется с помощью Register Aliases Table.

## 4 Вопрос

Q: С какой целью инструкцию Store разделяют на микро-операции STA (Store address calculation) и STD (Store data calculation) ?

A:

1. STA – микрооперация, позволяющая вычислить адрес STORE инструкции
2. STD – микрооперация, позволяющая вычислить данные STORE инструкции

В идеальном процессе исполнения программы зависимости между STORE и LOAD минимальны. Откуда вообще берутся зависимости? Например, зависимость между STORE и LOAD существуют, если адреса обращения в память инструкций пересекаются. В обычной реализации исполнения STORE адрес, по которому происходит загрузка в память, не известен до тех пор, пока не будут готовы данные для записи. Микрооперация STA позволяет вычислить адрес обращения в память, пока еще не вычислены данные. Таким образом, получается уменьшить число STORE и LOAD зависимостей.

## 5 Вопрос

Q: Объясните назначение и функции следующих аппаратных структур: ROB, Scheduler Queue (Issue Queue, Reservation Station), RAT, PRF, Load Buffer, Store Buffer

A:

1. ROB, re-order buffer – буфер для хранения последовательного порядка инструкции при Out Of Order исполнении. При ООО инструкции исполняются непоследовательно, однако порядок исполнения важен и нужен, чтобы знать архитектурное состояние. Таким образом, re-order buffer обеспечивает спекулятивное исполнение инструкций. Инструкция вытесняется из ROB, обновляя архитектурное состояние, если она была исполнена и последняя в буфере.
2. Scheduler Queue – очередь неготовых инструкций к исполнению, для которой производятся все необходимые проверки (аллокация регистров, отправка на исполнение).
3. RAT – таблица соответствия физических и логических регистров, используемых при алгоритме переименования регистров.
4. PRF – Physical Register File, регистровый файл, в котором описаны все физические регистры процессора.
5. Load Buffer, Store Buffer – буферы для хранения STORE и LOAD инструкций, которые обеспечивают разрешение зависимостей по данным между STORE и LOAD.

## 6 Вопрос

Q: Пусть каждая 5-ая инструкция в процессоре это Branch. Предсказатель переходов имеет точность 90%. Оцените, ROB какого максимального размера имеет смысл для такого процессора.

A:

Оцениваем вероятность того, что для ROB на N инструкций, все N инструкций будут исполнены.  $P_{\text{branch}} = \frac{1}{5}$  – вероятность branch инструкции,  $P_{\text{predict}} = \frac{9}{10}$  – точность предсказателя переходов.

Таким образом, в ROB размера N находится:

$$N_{\text{branch}} = N \cdot P_{\text{branch}}$$

– число бранчей в ROB.

$$P_{N,\text{predict}} = P_{\text{predict}}^{N_{\text{branch}}} = P_{\text{predict}}^{N \cdot P_{\text{branch}}}$$

– вероятность, что все бранчи будут предсказаны верно.

$$\Rightarrow N = \frac{\ln(P_{N,\text{predict}})}{\ln(P_{\text{predict}}) \cdot P_{\text{branch}}} = \frac{1}{P_{\text{branch}}} \cdot \frac{\ln(P_{N,\text{predict}})}{\ln(P_{\text{predict}})}$$

Положим, что все бранчи будут верно предсказаны с вероятностью  $P_{N,\text{predict}} = 0.8$

$$N = 5 \cdot \log_{P_{\text{predict}}}(P_{N,\text{predict}}) = 5 \log_{0.9}(0.8) \approx 10$$



## 7 Вопрос

Q: Что такое Memory Disambiguation?

A:

Это набор техник, применяемых в Out Of Order процессорах, для уменьшения количества зависимостей по памяти и улучшения параллелизма на уровне команд (ILP). К Memory Disambiguation техникам относятся такие, как Load buffer, Store buffer, Register renaming и т.д..

## 8 Вопрос

Q: В чем заключается проблема со спекулятивным исполнением Store инструкций?

A:

STORE инструкции не могут быть исполнены спекулятивно, потому что нет способа откатить их.

## 9 Вопрос

Q: Что такое Store forwarding и Load speculation в ООО процессоре?

A:

1. STORE инструкции не могут быть исполнены спекулятивно, переупорядочивать их исполнение нельзя. Однако если возникает такой случай, при котором STORE и LOAD инструкции обращаются к одинаковому адресу в памяти, можно применить оптимизацию Store Forwarding. При данной оптимизации LOAD может не дожидаться записи в память, а взять данные напрямую из STORE инструкции, минуя кэши.
2. Load speculation – техника, позволяющая инструкции LOAD исполняться спекулятивно, не дожидаясь исполнения всех STA микроопераций.

## 10 Вопрос

Q: Что такое Simultaneous Multithreading?

A:

Simultaneous Multithreading – технология, позволяющая одному физическому ядру исполнять несколько логических потоков. Логический процессор сохраняет своё собственное архитектурное состояние, каждая стадия в пайплайне занята всегда одним из потоков.