**Draft**

16

L1 CACHE

Description

Кэш-память первого уровня в процессорной системе на кристалле Selen: руководство и описание

**https://github.com/Zhikharev/selen**

Оглавление

Основные характеристики 2

Общее описание 2

Место в системе 3

Интерфейс взаимодействия с процессорным ядром 3

Описание поддерживаемых команд 5

Описание регистров 5

Микроархитектура 5

Поддержка верификации 5

Описание внутренних проверок 5

Описание функциональных групп 6

***Список изменений***

|  |  |
| --- | --- |
| **22.05.16** | **Начальная версия** |
|  |  |

# ****Основные характеристики****

Кэш-память инструкций и данных с характеристиками:

* Ассоциативность – 4 канала;
* Объём – 32 KB;
* Длина строки (размер блока данных) – 32 B;
* Общее количество строк – 32 KB / 32B = 1K;
* Алгоритм вытеснения: LRUm;
* Write-through, No-Write-Allocate.

# Общее описание

Общая формула расчёт размера выглядит следующим образом: , где, *N* – объём кэш-памяти, *l* – длина кэш-строки, *m* – ассоциативность, *s* – количество сетов, *b* – количество банков.

При *N* = 32 KB, *m* = 4, *l* = 32 B и *b* = 1, s получается равным 256.

При такой структуре памяти, физический адрес состоит из 3 элементов:

* tag – тэг адреса, хранится вместе с кэш-строкой;
* idx – номер сета;
* offset – смещение внутри кэш-строки.

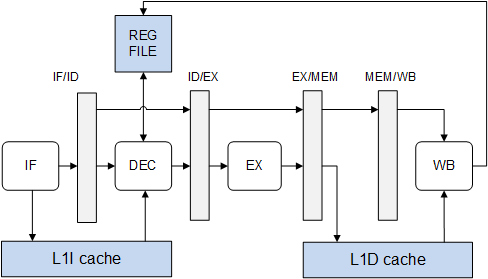
Для предлагаемых параметров и длине физического адреса 32 бита получаем:

* tag = 19 бит;
* idx = 8 бит;
* offset = 5 бита.

Таблица 1. Разбиение физического адреса в L1

|  |  |  |
| --- | --- | --- |
| Тэг | Индекс | Номер байта |
| [31:13] | [12:5] | [4:0] |

# Место в системе



## Интерфейс взаимодействия с процессорным ядром

Взаимодействие с подсистемой памяти реализовано согласно протоколу, представленному на рисунке 4.

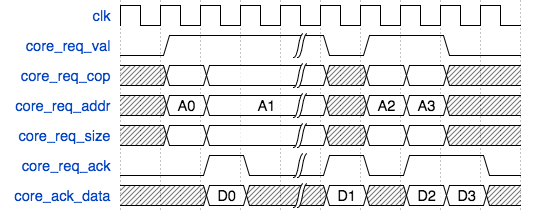


Рисунок 4. Протокол взаимодействия ядра с подсистемой памяти

Ядро выставляет core\_req\_val для подтверждения значимости запроса на шине и вместе с этим код операции core\_req\_cop, адрес обращения core\_req\_addr и размер запроса core\_req\_size. Со стороны подсистемы памяти выставляется core\_req\_ack для подтверждения значимости данных и вместе с этим данные core\_ack\_data.

**Таблица 4. Интерфейс взаимодействия ядра с кэш-памятью инструкций**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Разрядность** | **Направление** | **Описание** |
| **i\_req\_val** | **[0:0]** | **Output** | **Значимость запроса** |
| **i\_req\_addr** | **[31:0]** | **Output** | **Адрес запроса** |
| **i\_req\_ack** | **[0:0]** | **Input** | **Значимость ответа** |
| **i\_ack\_rdata** | **[31:0]** | **Input** | **Данные ответа** |

**R1** Сигнал i\_req\_val должен быть установлен на всём протяжении функционирования. Если он упадёт в 1’b0, то это приведёт к зависанию

**Таблица 5. Интерфейс взаимодействия ядра с кэш-памятью данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Разрядность** | **Направление** | **Описание** |
| **d\_req\_val** | **[0:0]** | **Output** | **Значимость запроса** |
| **d\_req\_addr** | **[31:0]** | **Output** | **Адрес запроса** |
| **d\_req\_cop** | **[2:0]** | **Output** | **Типа запроса:**  **3’b000 – RD**  **3’b001 – WR**  **3’b010 – RDNC**  **3’b011 - WRNC**  **Остальные – зарезервировано** |
| **d\_req\_wdata** | **[31:0]** | **Output** | **Данные для записи** |
| **d\_req\_size** | **[2:0]** | **Output** | **Размер запроса (1, 2 или 4 байта)** |
| **d\_req\_ack** | **[0:0]** | **Input** | **Значимость ответа** |
| **d\_ack\_rdata** | **[31:0]** | **Input** | **Данные ответа** |

## Описание поддерживаемых команд

Кэш-памяти данных поддерживает следующие типы операций:

* RD – кэшируемое чтение. Если требуемые данные находятся в кэш-памяти, то они возвращаются ядру. Если данных в кэш-памяти нет, то из внешней памяти запрашивается целая кэш-строка и требуемое слово данных возвращается ядру;
* WR – кэшируемая запись. Если адресуемая кэш-строка присутствует в кэш-памяти, то происходит обновление данных и формируется запрос на запись во внешнюю память. По окончанию операции актуальные данные есть и в кэш-памяти и во внешней памяти. Если кэш-строки нет в кэш-памяти, то просто формируется запрос записи во внешнюю память;
* RDNC – некэшируемое чтение. Формируется запрос чтения во внешнюю память. По окончанию операции кэш-строка не заводится;
* WRNC – некэшируемая запись. Формируется запрос записи во внешнюю память.

**R2**  В контроллер не должны поступать кэшируемые и некэшируемые запросы с одинаковым адресами

# ****Описание регистров****

STAT\_L1I 'h00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Разрядность** | **Тип доступа** | **Описание** | **Значение по сбросу** |
| req\_num | 31:16 | RO | Количество полученных запросов | 16'h0000 |
| hit\_num | 15:0 | RO | Количество попаданий | 16'h0000 |

STAT\_L1D 'h04

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Разрядность** | **Тип доступа** | **Описание** | **Значение по сбросу** |
| req\_num | 31:16 | RO | Количество полученных запросов | 16’h0000 |
| hit\_num | 15:0 | RO | Количество попаданий | 16'h0000 |

STAT\_L1D 'h04

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Разрядность** | **Тип доступа** | **Описание** | **Значение по сбросу** |
| req\_num | 31:16 | RO | Количество полученных запросов | 16’h0000 |
| hit\_num | 15:0 | RO | Количество попаданий | 16'h0000 |

STAT\_CTRL\_STATUS ‘h08

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Разряд-ность** | **Тип доступа** | **Описание** | **Значение по сбросу** |
| reserved | 31:7 | RO | Зарезервированные биты |  |
| l1i\_req\_num\_ov | 6 | RO |  | 1'b0 |
| l1i\_hit\_num\_ov | 5 | RO |  | 1'b0 |
| l1d\_req\_num\_ov | 4 | RO |  | 1'b0 |
| l1d\_hit\_num\_ov | 3 | RO |  | 1'b0 |
| l1d\_nc\_count | 2 | RW | Учитывать некэшируемые запросы при подсчёте общего числа запросов для STAT\_L1D  1’b0 – некэшируемые запросы не учитываются  1’b1 – некэшируемые запросы учитываются | 1'b0 |
| l1d\_stat\_rst | 1 | W1C | Обнулить значения счётчиков в регистре статистики кэш-памяти данных, а так же сбросить статус переполнения | 1'b0 |
| l1i\_stat\_rst | 0 | W1C | Обнулить значения счётчиков в регистре статистики кэш-памяти инструкций, а так же сбросить статус переполнения | 1’b0 |

# Микроархитектура

# Поддержка верификации

## Описание внутренних проверок

Таблица 2. Внутренние проверки

|  |  |
| --- | --- |
| Свойство | Описание |
| offset\_allign\_p |  |
| tag\_cmp\_with\_val\_vect\_p | one hot |
| lru\_way\_vect\_p | one hot |

## Описание функциональных групп