Национальный исследовательский ядерный университет «МИФИ»

Институт Интеллектуальных Кибернетических Систем



|  |  |
| --- | --- |
|  | Кафедра  «Компьютерные системы и технологии» |

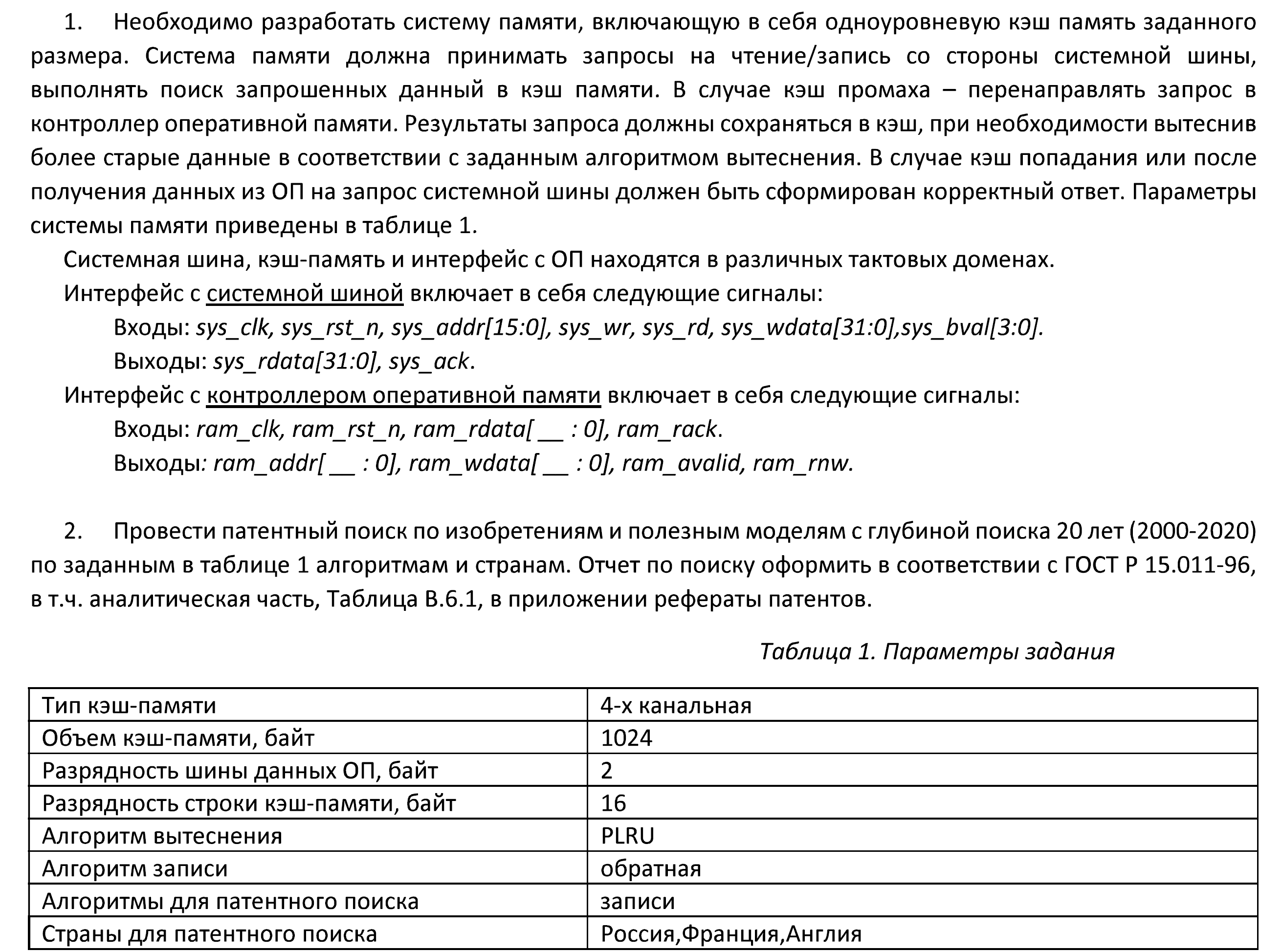
Пояснительная записка к курсовому проекту

на тему «Разработка системы памяти»

Студент гр. М22-512 Григорьев М.П. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

Руководитель Скитев А.А. / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

Техническое задание



ВВЕДЕНИЕ

С2истема памяти – одно из основных и важных узлов любой вычислительной системы. Кэш-память – промежуточный буфер между процессором и оперативной памятью, к котором обеспечен быстрый доступ. В кэш-памяти содержится информация, которая может быть запрошена процессором с наибольшей вероятностью. Это позволяет уменьшить количество обращений к более медленной памяти – оперативной памяти, и уменьшить время простоя ЦПУ.

Целью данной работы является изучение алгоритмов работы кэш-памяти путем разработки одноуровневой однопортовой кэш-памяти. В первом разделе проведен обзор алгоритмов работы, заявленных в техническом задании. Во втором разделе рассмотрены результаты разработки и временные диаграммы работы разработанной системы памяти.

В результате выполняемой работы была реализована система памяти, соответствующая техническому заданию.

# 1 Обзор кэш-памяти

В данной работе рассматриваются алгоритмы, необходимые для реализации системы памяти. Среди этих алгоритмов будут рассмотрены:

* алгоритм работы разрабатываемой системы памяти;
* принцип работы множественно-ассоциативного кэша;
* алгоритм вытеснения PLRU;

## Обзор алгоритмов

Кэш-память относится к классу ассоциативно запоминающих устройств. В ней для ускорения доступа к данным происходит их буферизация из оперативной памяти.

По способу отображения кэш память классифицируется по следующим типам: память с прямым отображением (у каждой строки из ОП есть только одна строка кэш-памяти, в которой эта информация может располагаться), полностью ассоциативную память (любая строка из ОП может находиться в любой строке кэша) и множественно-ассоциативную, которая рассмотрена подробнее ниже и после была реализована.

### Множественно-ассоциативный кэш

Множественно-ассоциативный кэш состоит из нескольких независимых сегментов(каналов), каждый из которых представляет собой кэш с прямым отображением, а сами каналы являются полностью ассоциативными по отношению к оперативной памяти. Любой элемент оперативной памяти может быть размещен в любом канале кэш-памяти, но внутри канала ему соответствует строго определенная кэш-строка.

Множественно-ассоциативная кэш-память сочетает в себе простоту кэш-памяти с прямым отображением и большую вероятность кэш-попадания полностью ассоциативной кэш-памяти.

Для поиска тега в памяти нужно обратиться к строке, номер которой задан в адресе (биты, отвечающие за индекс). Сравнивается тег, поданный на вход и хранимый в памяти тегов. В случае равенства тегов необходимые данные расположены по смещению, также определяемому адресом.

### Вытеснение PLRU

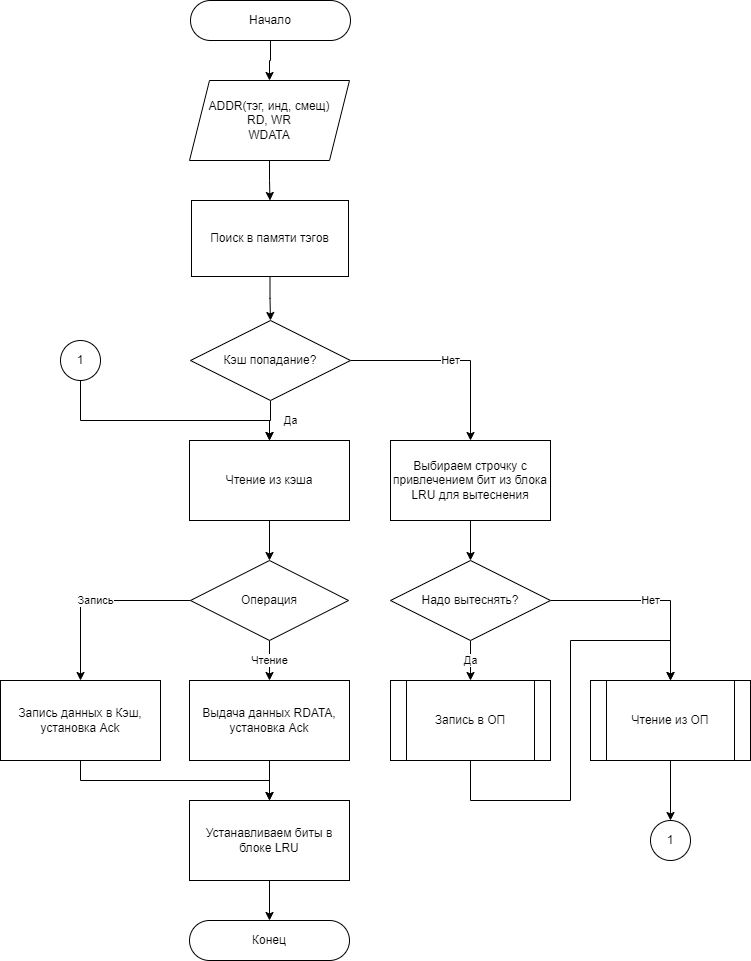
Разработано множество алгоритмов, которые разными способами находят строку, которую нужно вытеснить из кэш-памяти в оперативную память для записи новой строки. Основные из них – Least Recently Used (LRU), Least Frequently Used (LFU), First In First Out (FIFO), Pseudo LRU (PLRU). В техническом задании указан последний.

Суть данного метода состоит в том, что в ситуации, когда необходимо вытеснить тег из памяти, вытесняется тот тег, на который указывает текущая ветка дерева-LRU. Вытесняемый тег необязательно наиболее последний по использованию.

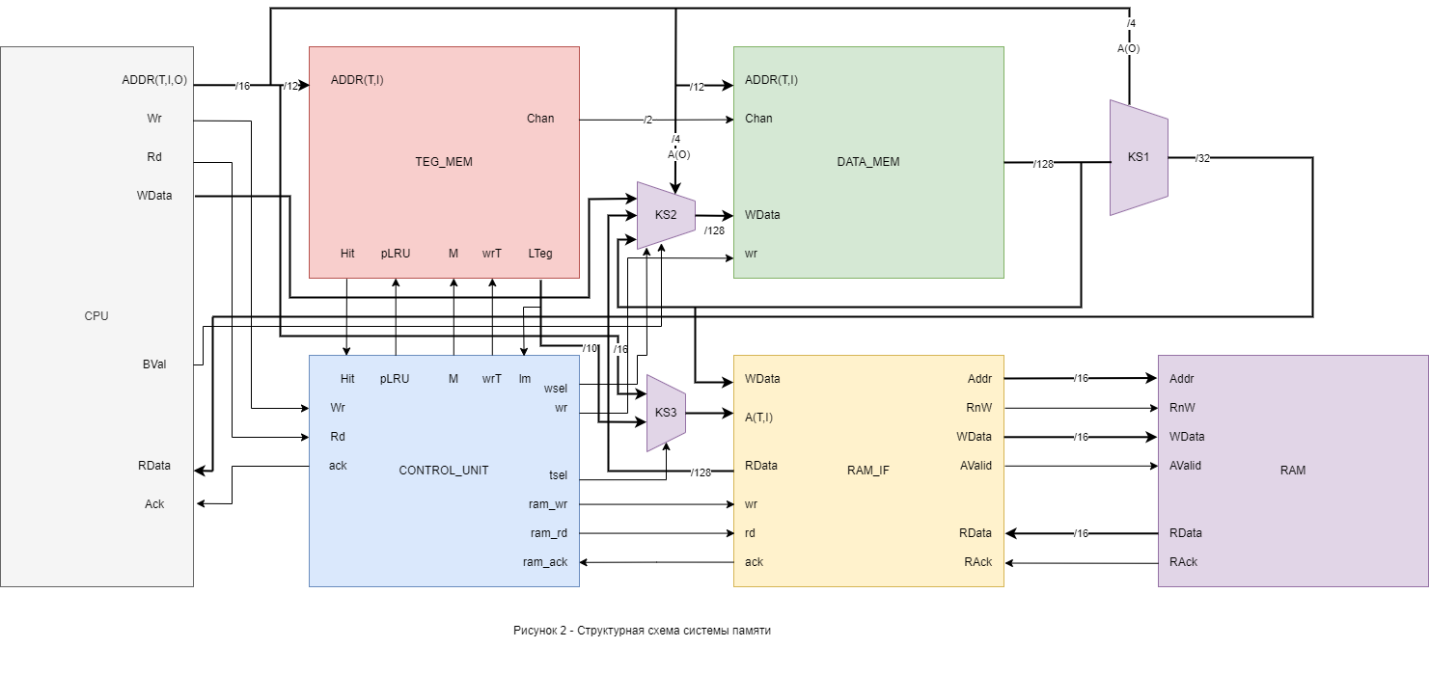
# Обзор кэш-памяти

## Алгоритм работы

Ниже рассмотрен алгоритм работы разрабатываемой системы. На вход подаются адрес ячейки памяти, запрос на запись или чтение и данные. После поиска тега в памяти тегов происходит переход к ветке попадания или промаха. По заданию реализовывается кэш с обратной записью. Была выбрана реализация простой обратной записи – запись в оперативную память происходит после вытеснения из кэша. Таким образом, если кэш не заполнен, то в случае промаха новые данные из оперативной памяти записываются в канал, на который указывает PLRU. После записи данный указатель сдвигается. После чего алгоритм переходит к ветке кэш-попадания.



## Структурная схема

Ниже представлена структурная схема системы памяти, разработанная в соответствие с техническим заданием. 

На схеме представлены 4 основных узла: память тегов, память данных, устройство управления и интерфейс оперативной памяти. Устройство этих модулей будет рассмотрено в следующих подразделах.

Адрес обращения к ОП используется кэшем следующим образом:

* младшие 4 бита кодируют смещение внутри канала
* следующие 4 бита используются для выборки нужной строки внутри каналов памяти;
* старшие 8 бит отведены на тег.

|  |  |  |
| --- | --- | --- |
| Тэг | Индекс | Смещение |
| 8 бит | 4 бит | 4 бит |

Разбиение адреса ячейки ОП

* 1. Адрес обращения к оперативной памяти

Адрес обращения к оперативной памяти (16-битный) используется кэшем следующим образом:

* младшие 4 бита кодируют смещение внутри слова;
* следующие 4 бита используются для выборки строки внутри каналов памяти;
* старшие 8 бит отведены на тэг.

## Память данных

По техническому заданию разрабатывается 4-ёх канальная память объемом 1КБ. Так как разрядность строки каждого канала составляет 16 байт, то иметь для каждого канала 16 строк.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 16 \* 8 | 16 \* 8 | 16 \* 8 | 16 \* 8 |
| 15 |  |  |  |  |
| 14 |  |  |  |  |
| 13 |  |  |  |  |
| 12 |  |  |  |  |
|  | … | | | |
|  |
| 2 |  |  |  |  |
| 1 |  |  |  |  |
| 0 |  |  |  |  |
|  | Канал 3 | Канал 2 | Канал 1 | Канал 0 |

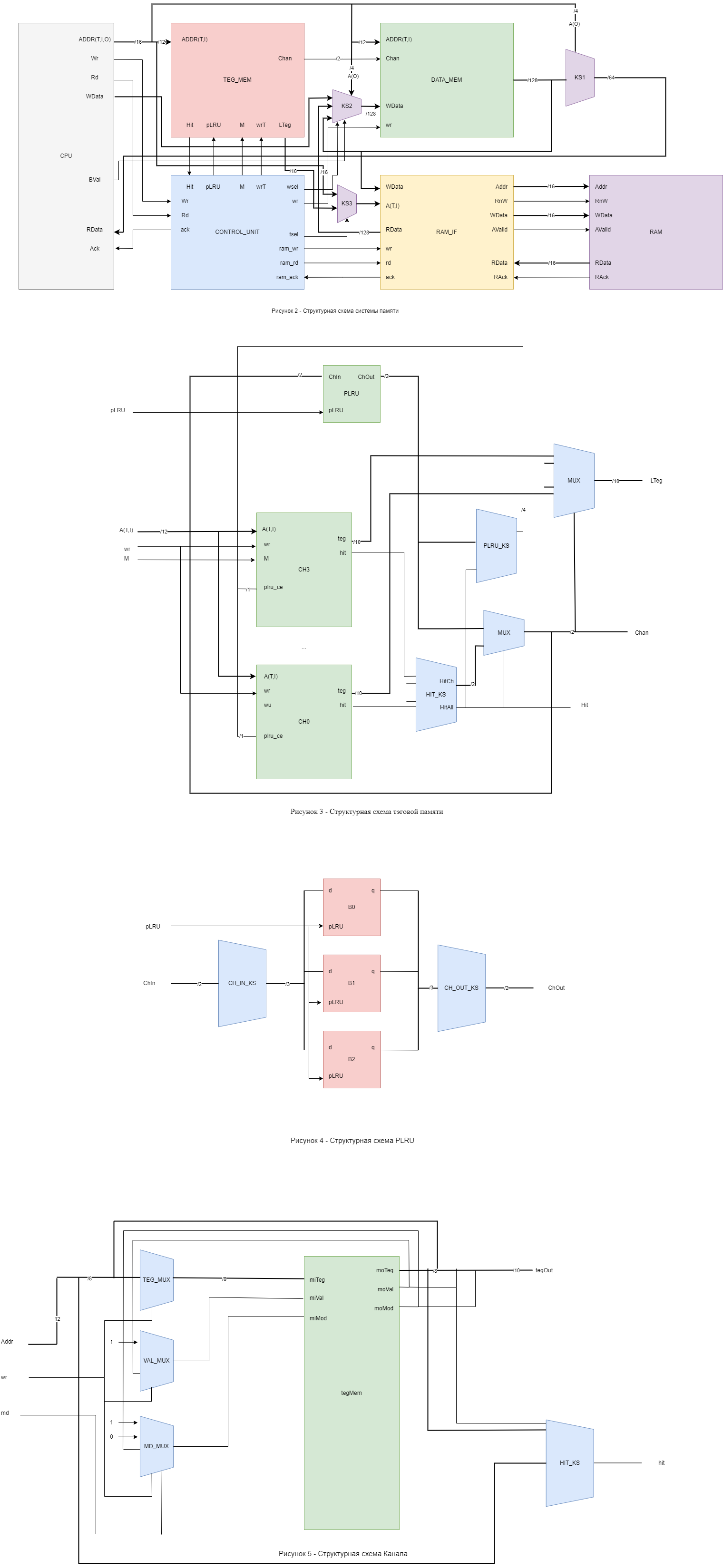
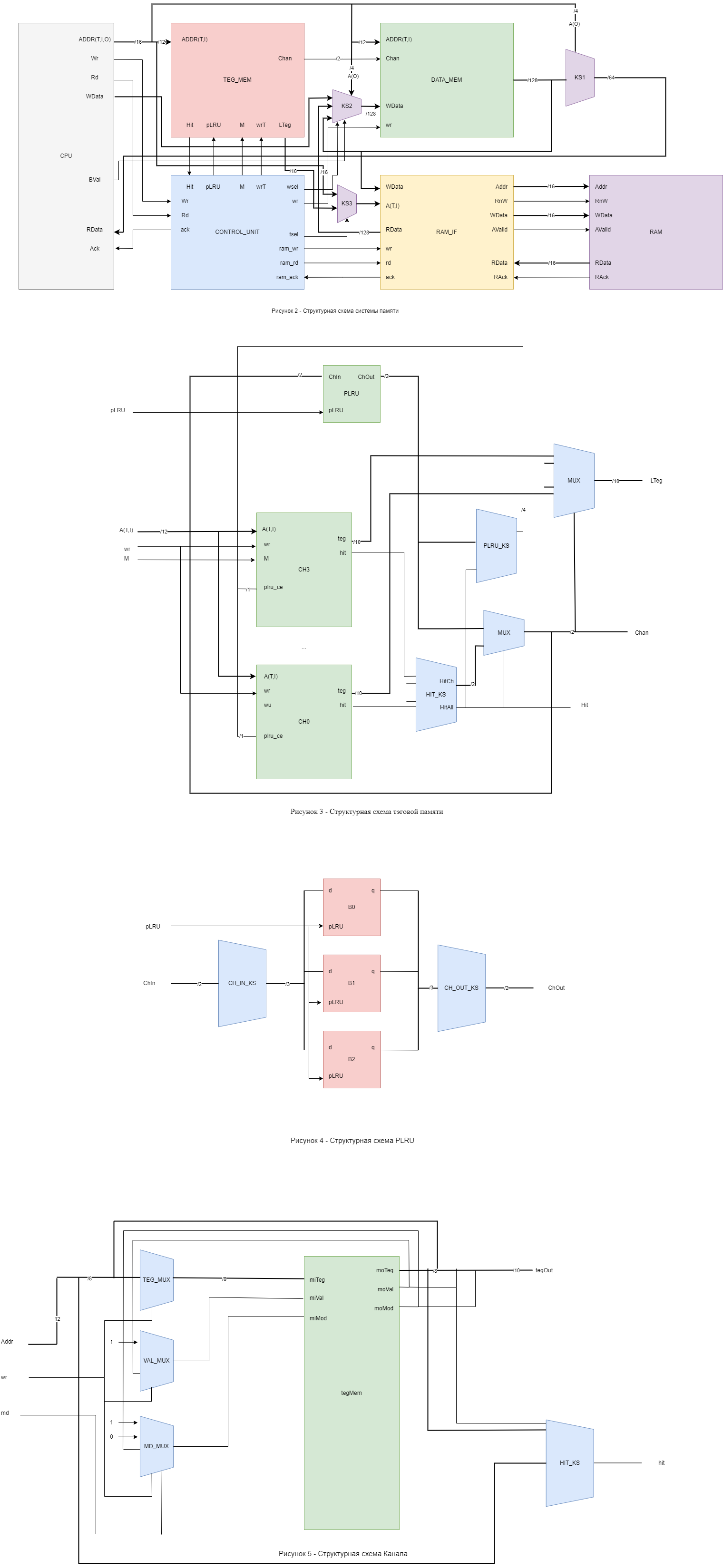
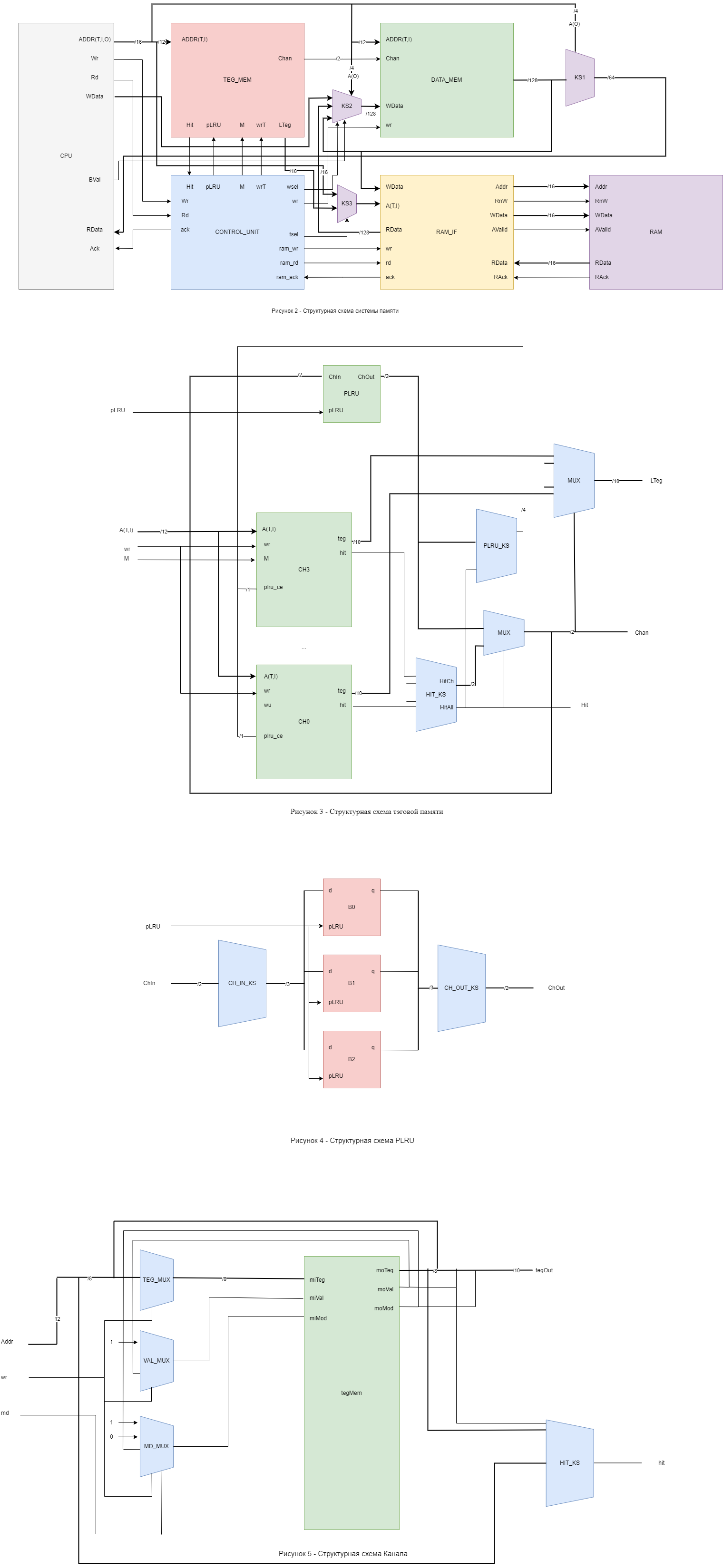
## Память тегов

Память тэгов состоит из четырёх каналов, ширина каждого из которых состоит из 12 бит: 8 отводится на хранение тега, 1 бит занимает бит валидности, 1 бит занимает бит модификации.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 |  |  |  |  |
| 14 |  |  |  |  |
| 13 |  |  |  |  |
| 12 |  |  |  |  |
|  | … | | | |
|  |
| 2 |  |  |  |  |
| 1 |  |  |  |  |
| 0 |  |  |  |  |
|  | Канал 3 | Канал 2 | Канал 1 | Канал 0 |

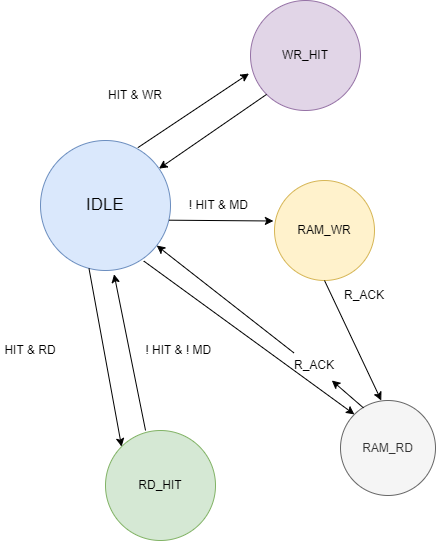
|  |  |  |
| --- | --- | --- |
| Тэг | Val | M |
| 8 | 1 | 1 |

Четыре канала связаны двумя комбинационными схемами, одна из которых - PLRU\_KS - формирует номер канала для записи при вытеснении, а другая - HIT\_KS - подает управляющие сигналы в канал попадания.



## Устройство управления

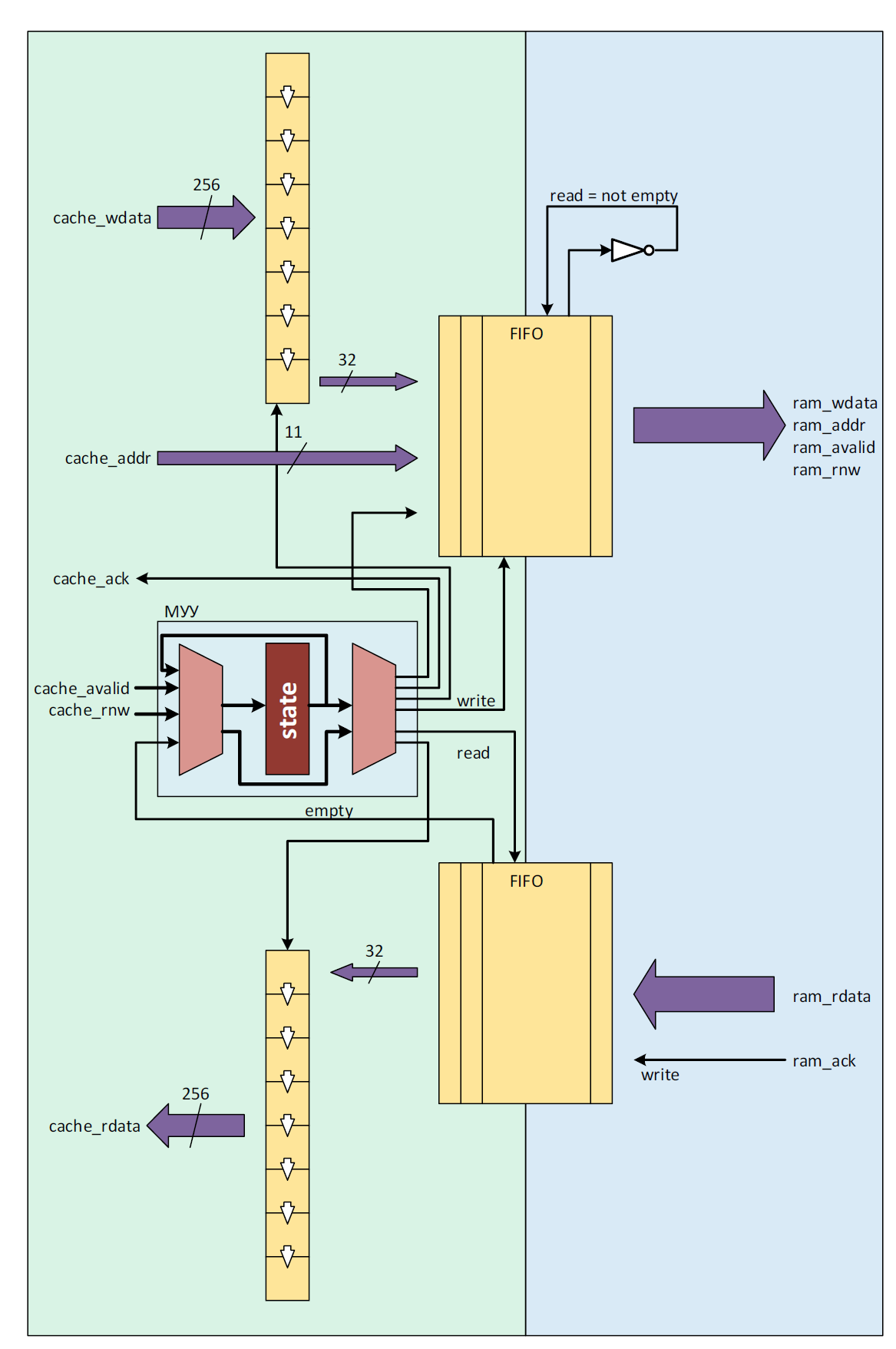
Логика работы устройства управления отражена в графе состояний, представленном ниже и повторяет разработанный алгоритм работы системы памяти.



## Интерфейсы CPU и RAM

Ввиду того, что предполагается, что RAM и CPU будут работать на отличающейся от кэша тактовой частоте, необходимо разработать систему для перехода информации через домен частот.

Объем оперативной памяти составляет 64 КБ. Разрядность строки - 16 байт. Соответственно, количество строк в ОП - 4096. Ширина шины данных RAM - 2 байта. Это значит, что 16 байт будут переданы на 8 циклов. Переход через домен частот кэш-ОП реализован с помощью FIFO. Буферами управляет автомат состояний.



# 3 Реализация кэш-памяти

## 3.1 Временные диаграммы

Для отладки и тестирования были разработаны программные тесты, моделирующие работу системы памяти в ответ на задаваемые входные воздействия. Приведенные ниже временные диаграммы работы системы соответствуют следующему набору тестов:

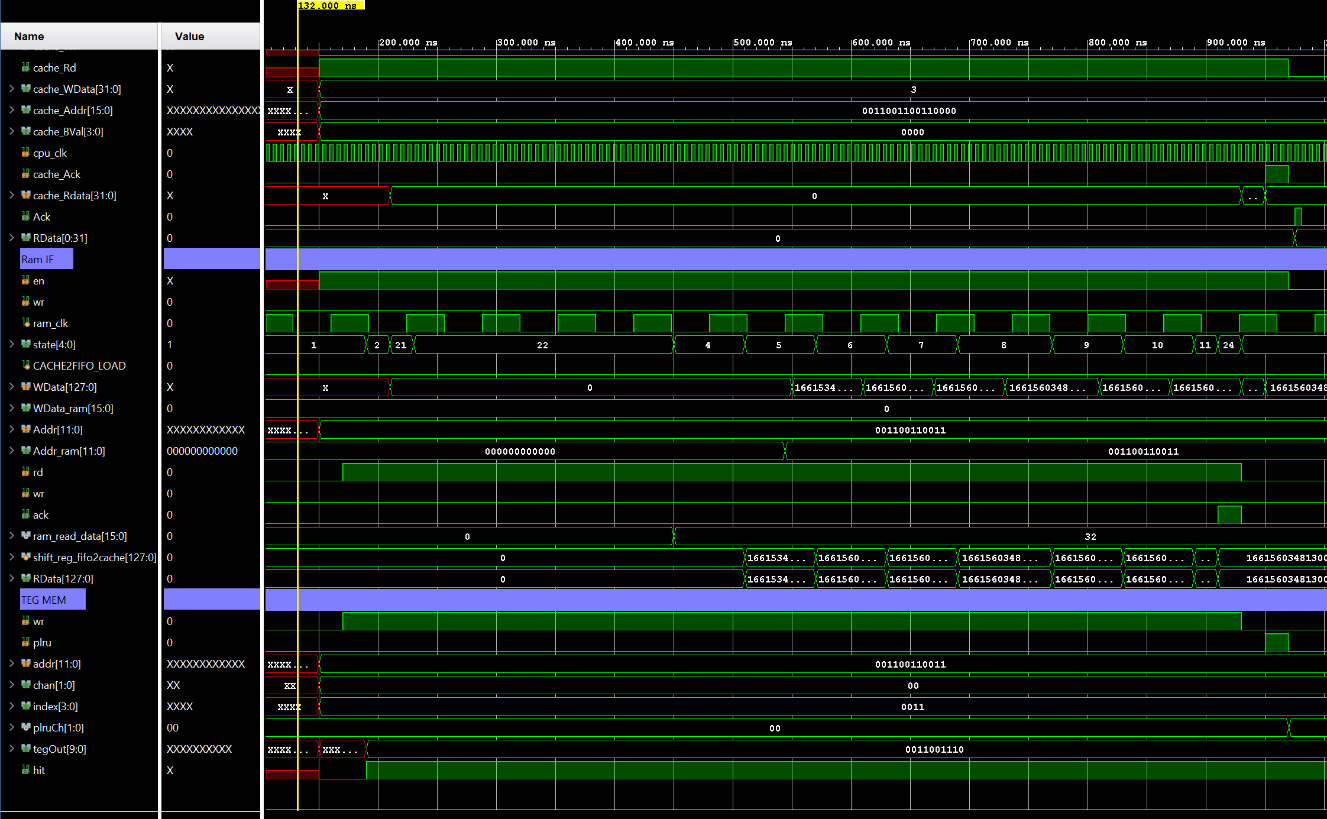
1. Установка и сброс сигнала системного сброса для сброса системы памяти;
2. Запись в данных (запись с промахом);
3. Запись в данных по тому же адресу (запись с попаданием);
4. Чтение данных по тому же адресу (чтение с попаданием);
5. Чтение данных по другому адресу (чтение с промахом);
6. Последовательная запись данных в 8 каналов кэша по одному индексу (заполнение кэша для демонстрации вытеснения);
7. Чтение данных по последнему адресу записи (чтение с попаданием);
8. Запись данных в другой адрес по тому же индексу (запись с промахом, вытеснение)

Временные диаграммы работы системы памяти при соответствующих случаях приведены на рисунках 8-12.

## Чтение промах

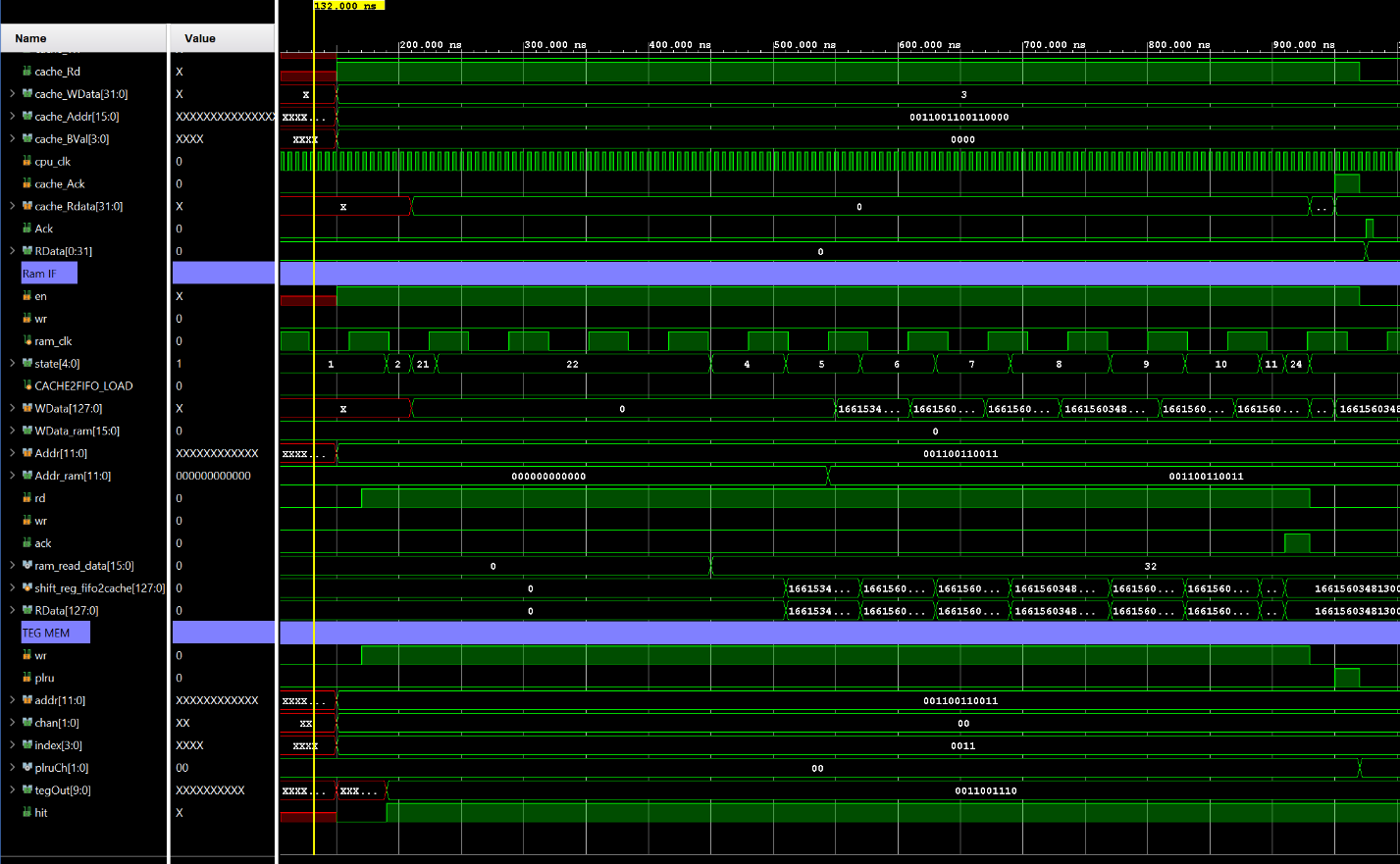
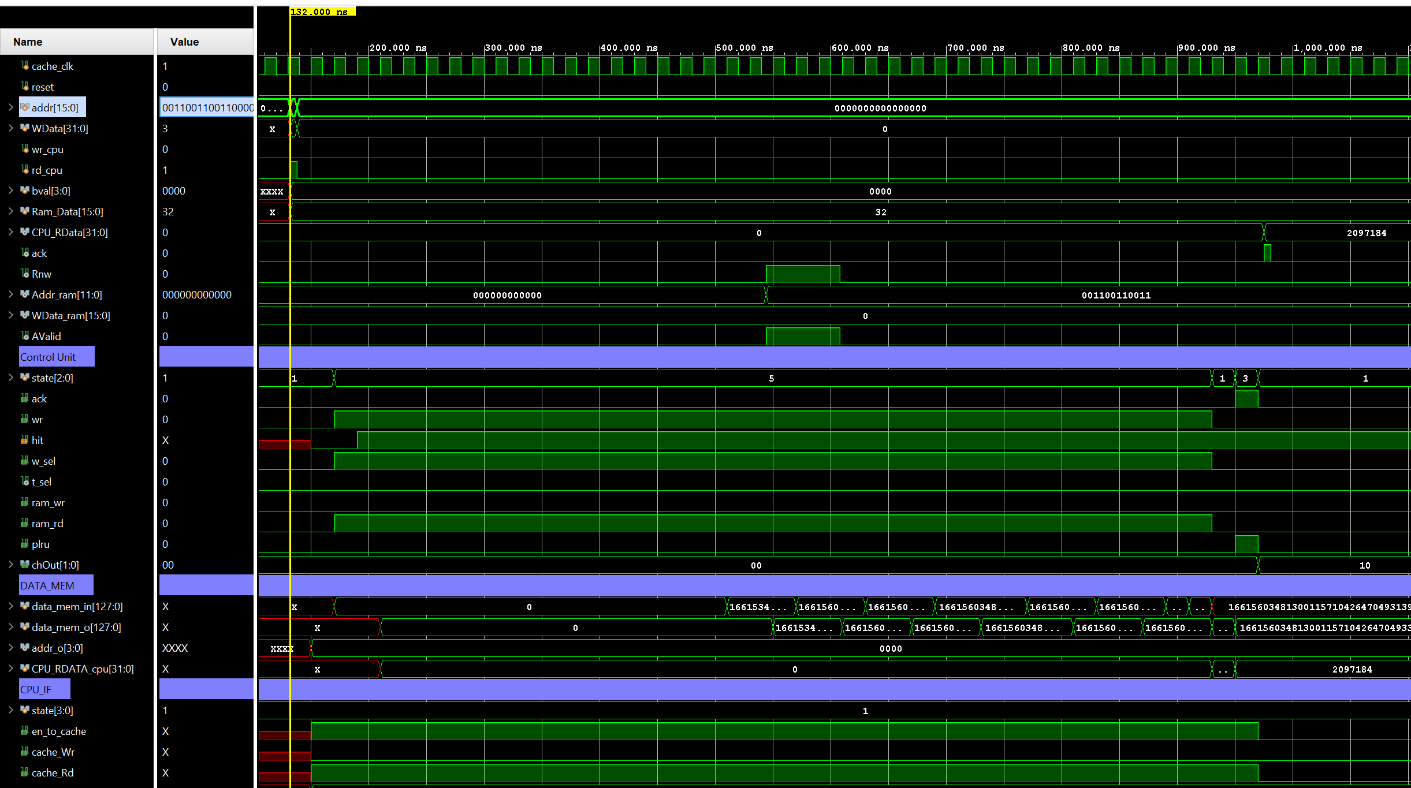
Фрагмент диаграммы ниже показывает случай чтения с промахом. Можно видеть передачу запроса на чтение через домен CPU. Автомат состояний УУ переходит в состояние S\_RAM\_RD (5), после чего начинается процесс работы автомата состояний RAMIF и переноса данных через домен RAM. После завершения приема данных из оп УУ переходит в состояние S\_RD\_HIT (3), выполняет чтение CpuIF.





## Чтение попадание

Диаграммы ниже показывают ситуацию, когда при запросе чтения искомый тег уже находился в памяти. Очевидно, нет обмена с оперативной памятью, т. к. не надо читать и выгружать изменения в RAM. Происходит изменение канала для вытеснения по алгоритму PLRU.



* 1. Запись промах

На следующей диаграмме представлена ситуация записи и промаха. Запись происходит в первый канал. Так как произошел промах, начинается считывание из RAM новой строки данных. После чего данные считываются.

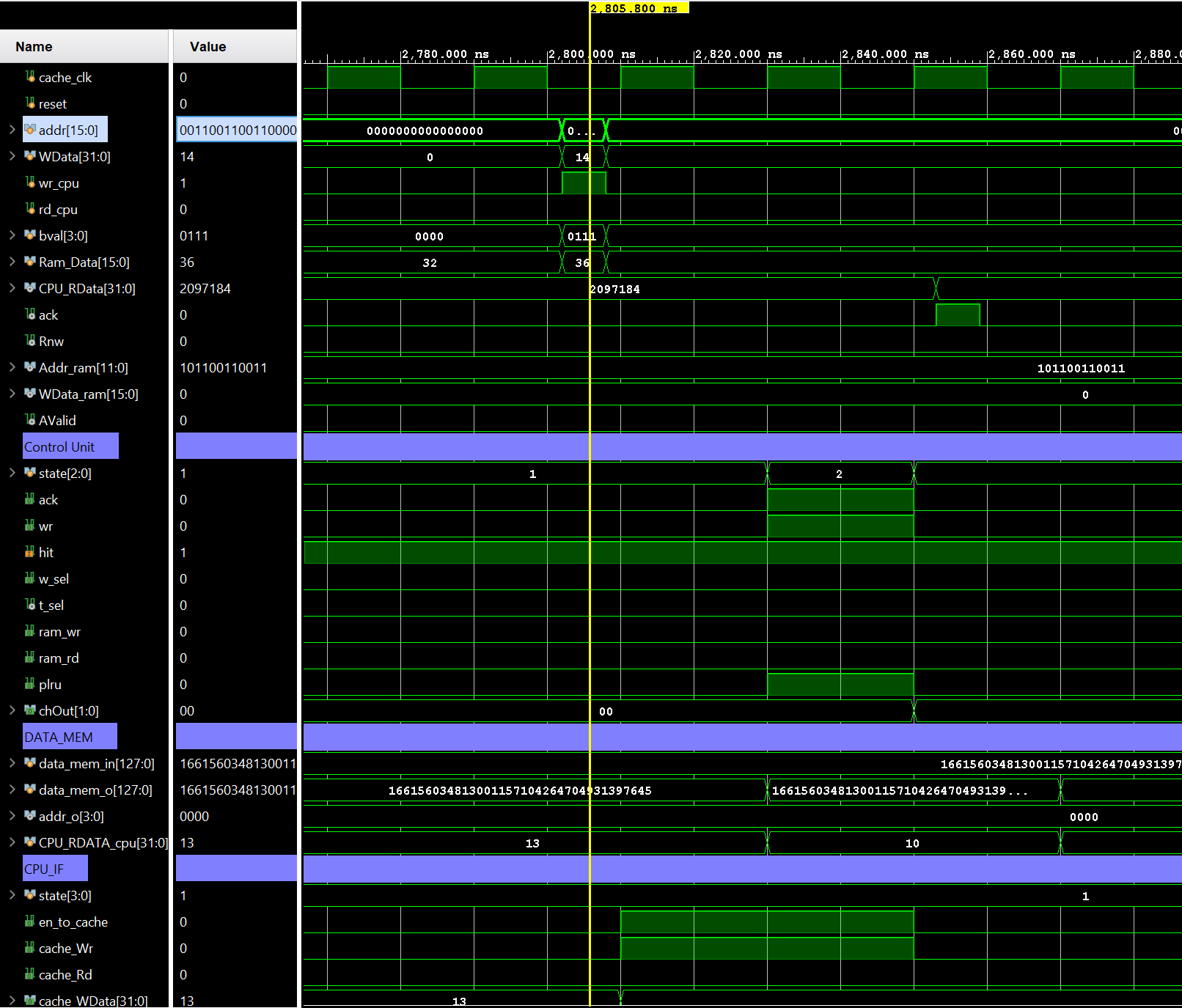
После завершения обработки ситуации записи при попадании, начинается запись измененных данных в RAM.

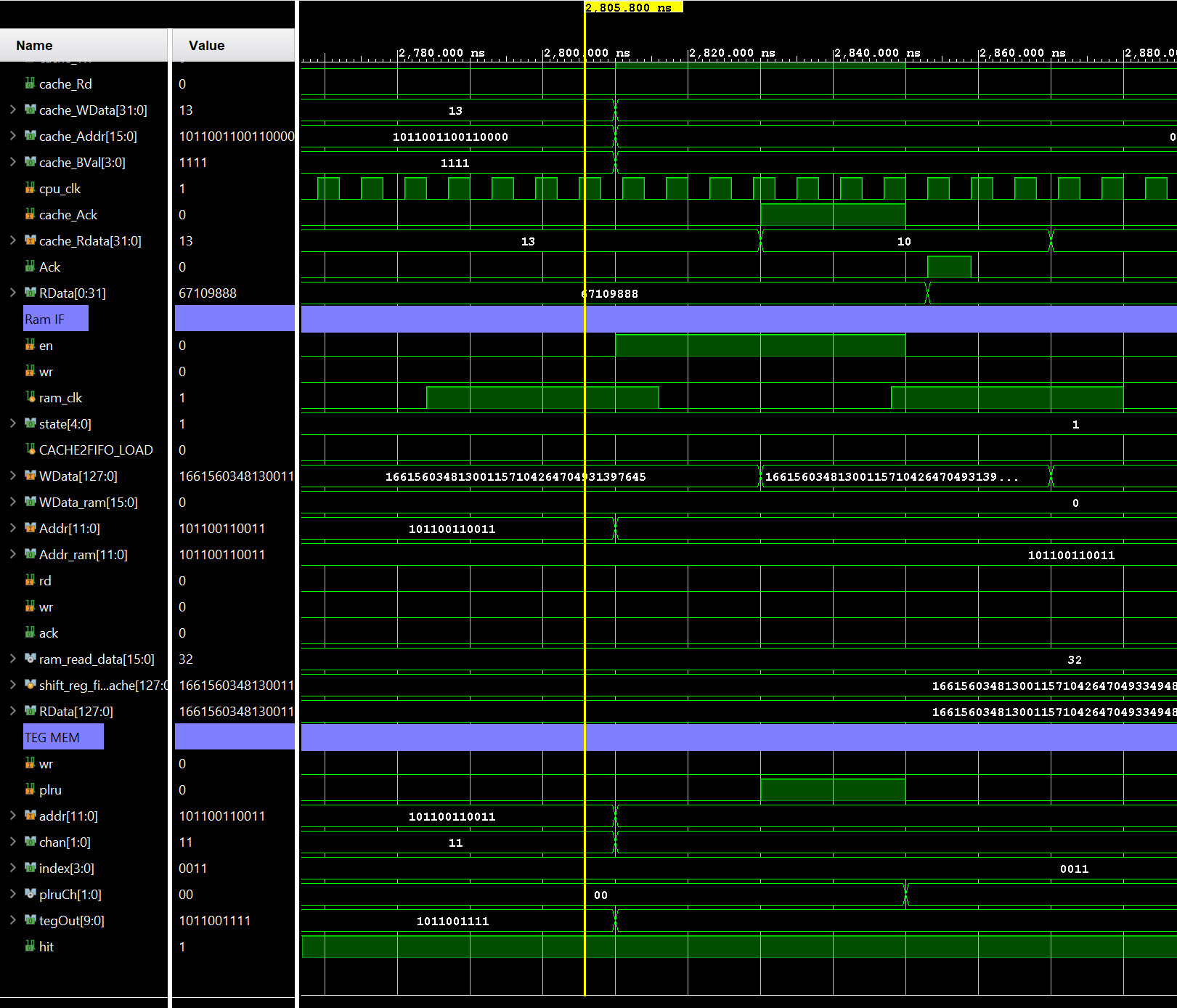




* 1. Запись попадание

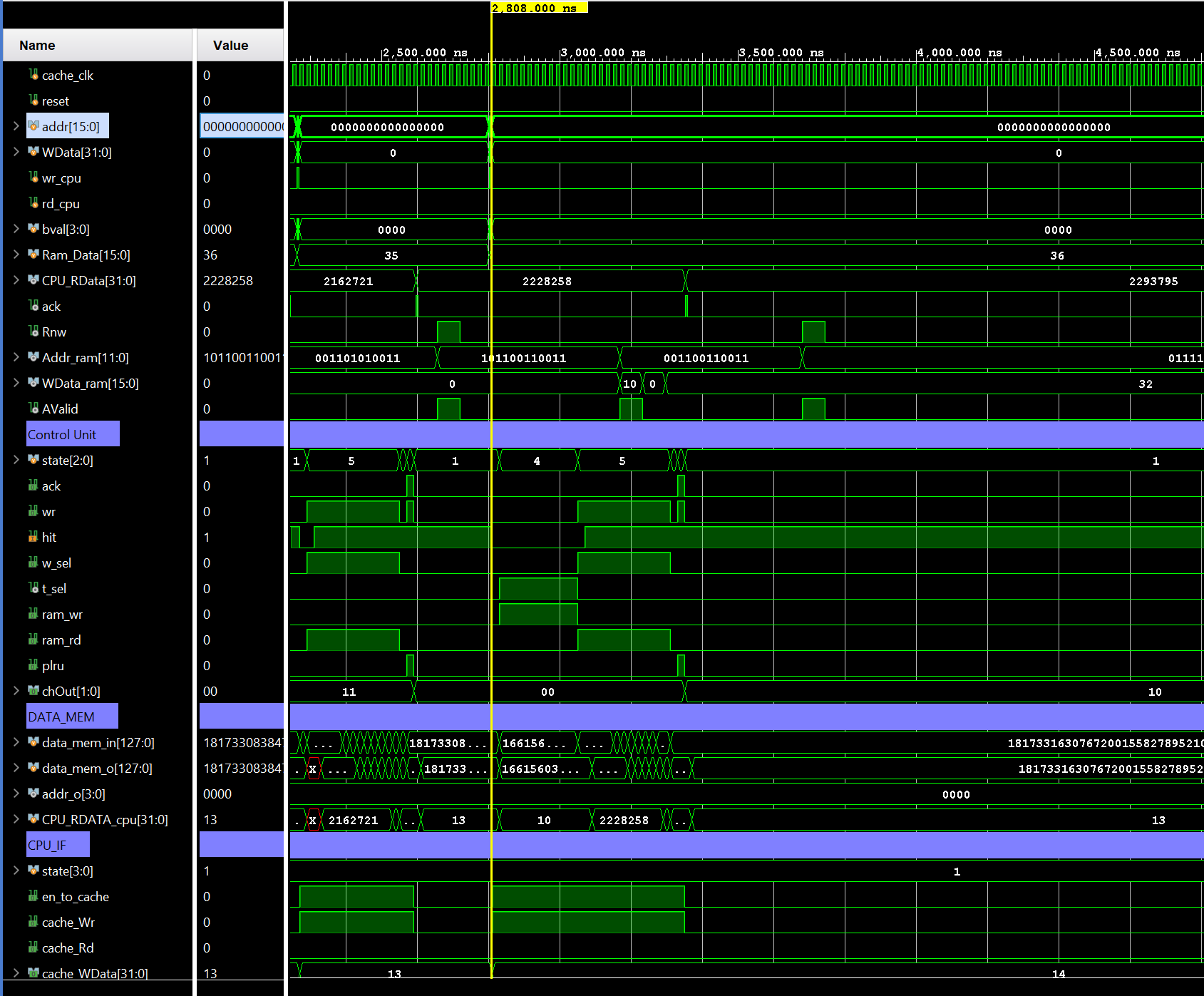
Диаграмма ниже отражает ситуацию кэш-попадания при записи. Аналогично предыдущим ситуациям, из начального состояния происходит переход в состояние пропадания при записи s\_wrhit, после чего новые данные передаются в интерфейс ОП для выгрузки в память

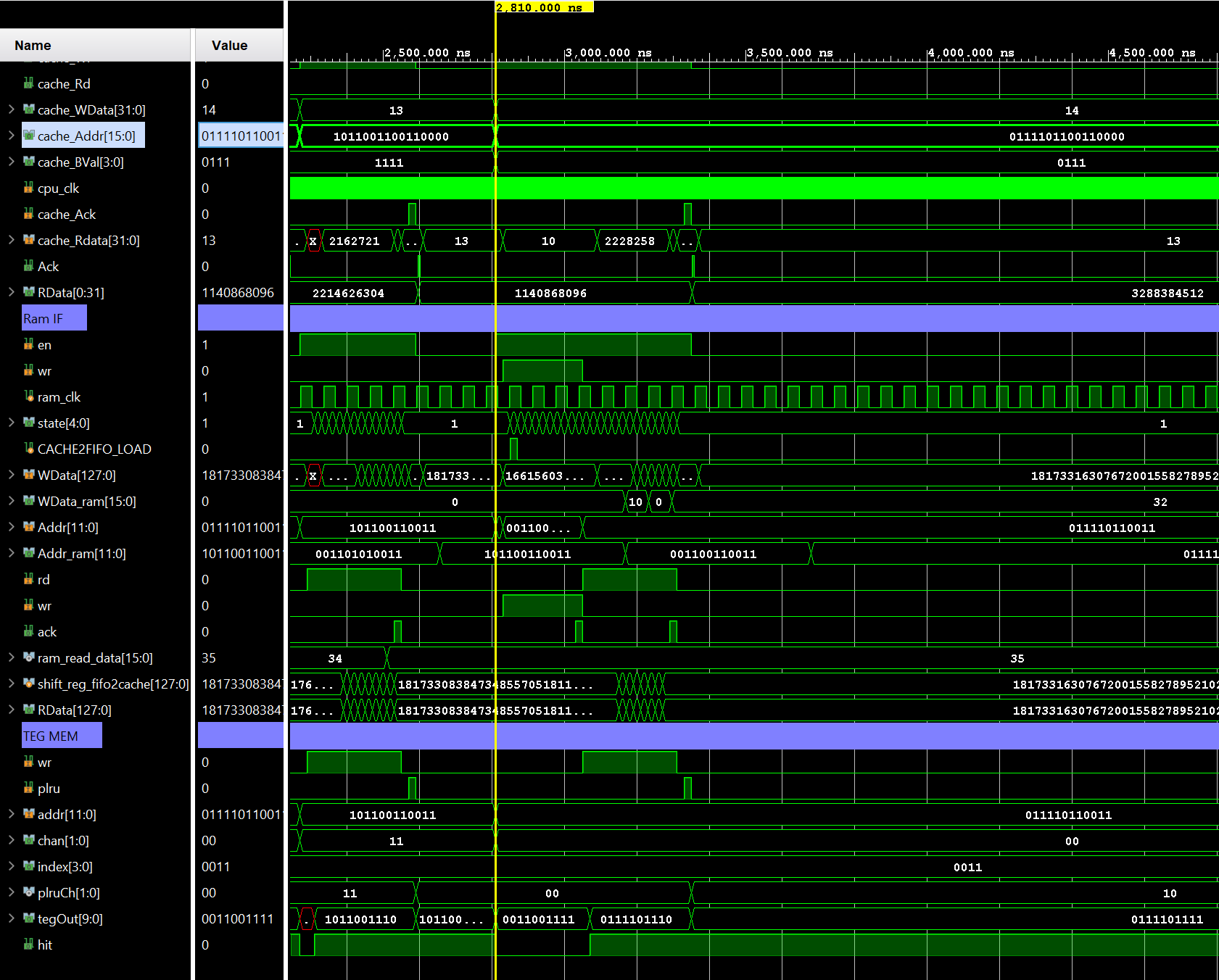




* 1. Вытеснение

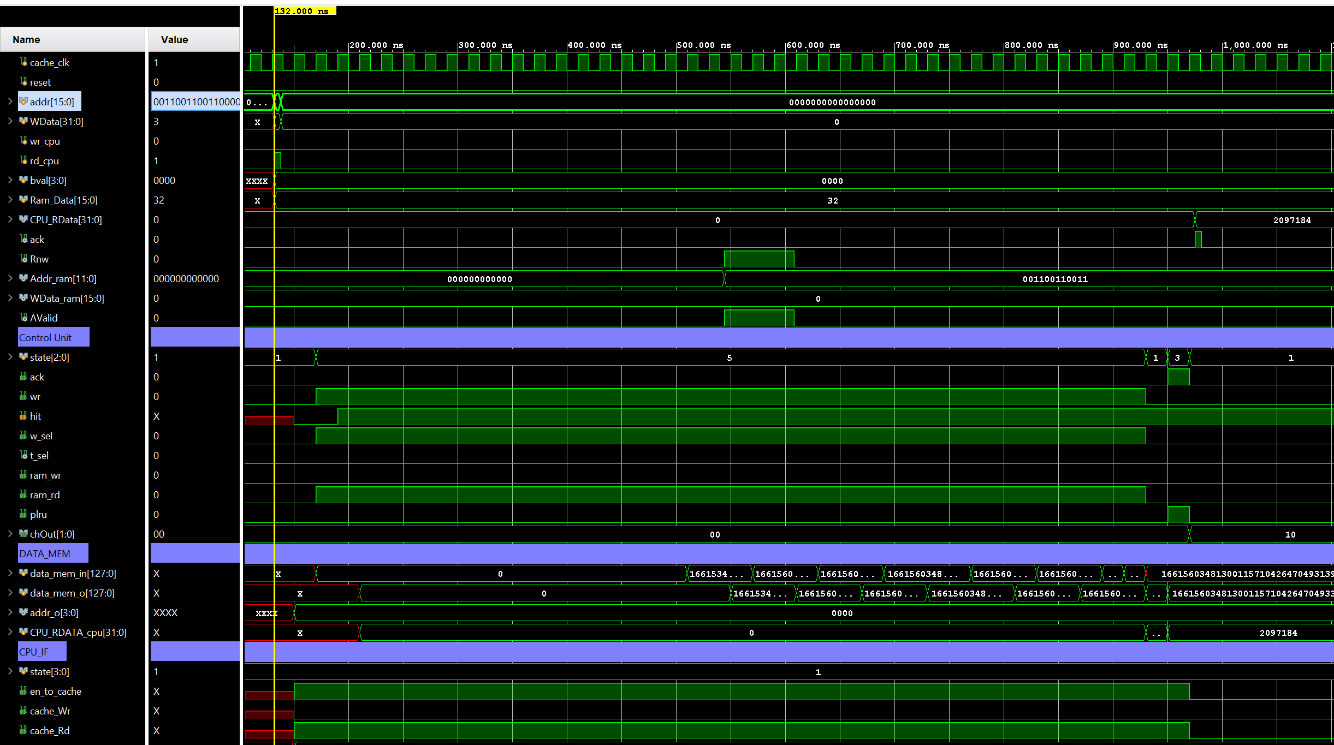
Диаграмма ниже показывает отработку алгоритма вытеснения (PLRU).

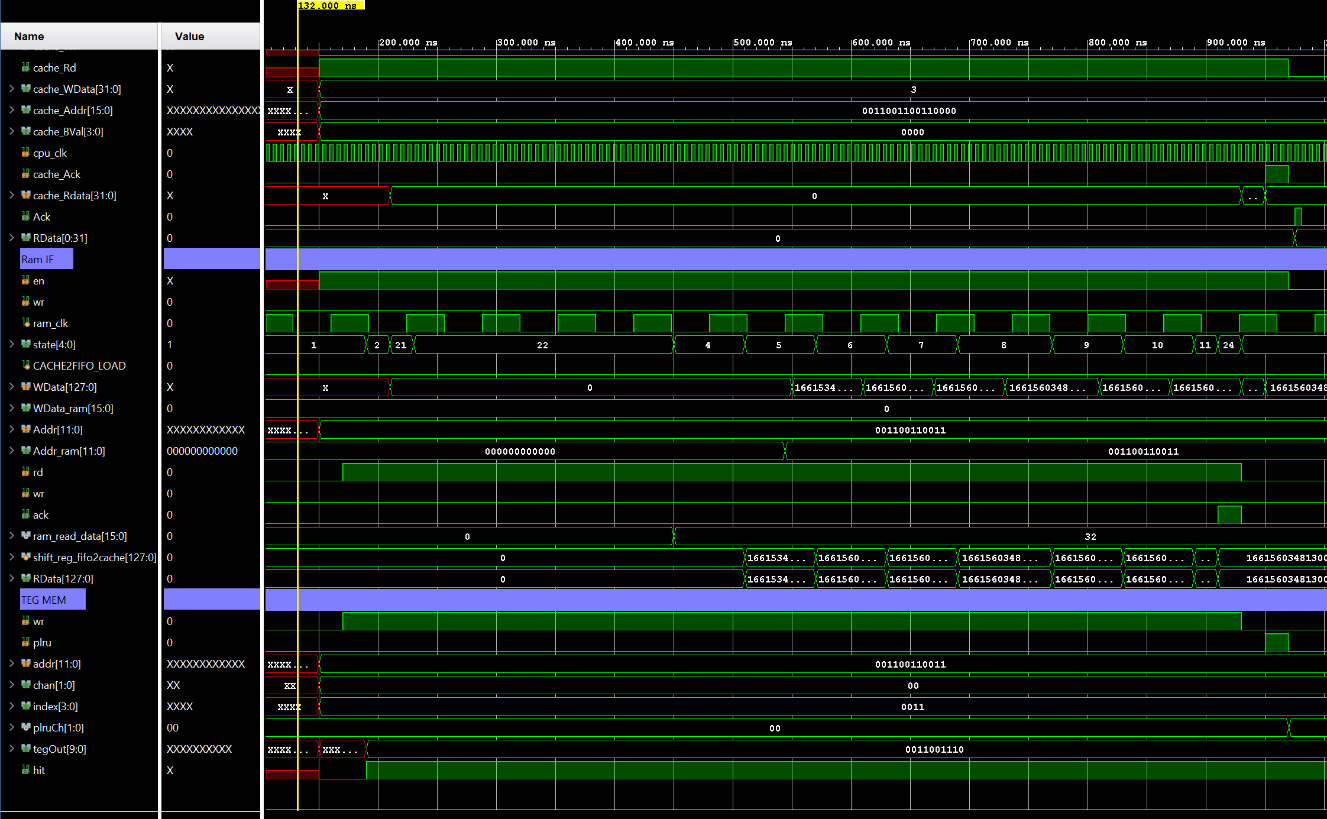




* 1. Переход через домен частот

Диаграмма ниже показывает переход через домен частот при: ram\_clk = 32, cache\_clk = 10, cpu\_clk = 3.





* 1. Переход через домен частот

Диаграмма ниже показывает переход через домен частот при: ram\_clk = 3, cache\_clk = 10, cpu\_clk = 32.



* 1. Автоматизированное тестирование

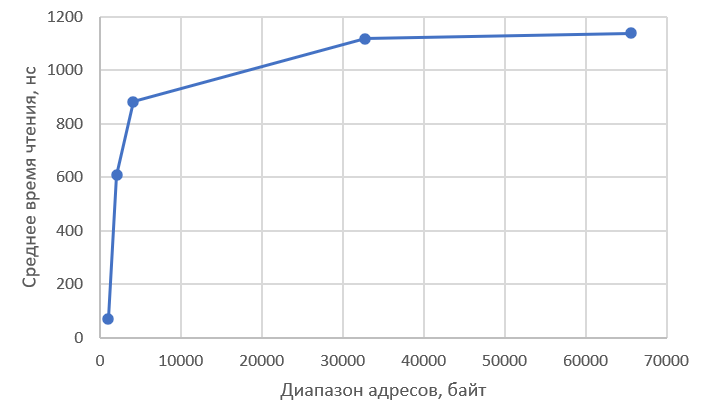
Были разработаны автоматизированные тесты, проверяющие случайное чтение/запись данных из заданного диапазона.

Для оценки зависимости времени чтения из системы памяти от диапазона адресов было проведено тестирование с различными диапазонами адресов чтения, результаты которого приведены в таблице 1 (N = 20000 чтений, T кэш = 3 нс, Т процессора = 10 нс, Т оперативной памяти = 32 нс).

Таблица 1. Результаты автоматизированного тестирования системы памяти

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объем данных, Кбайт | 1 (кэш) | 2 | 4 | 32 | 64 (ОП) |
| Время N чтений, нс | 1424154 | 12143640 | 17635074 | 22321944 | 22741020 |
| Среднее время чтения, нс | 71,21 | 607,18 | 881,75 | 1116,10 | 1137,05 |

 На рисунке 15 приведена зависимость среднего времени чтения данных из системы памяти в зависимости от объема данных



Зависимость среднего времени чтения данных из системы памяти от объема данных

# ЗАКЛЮЧЕНИЕ

В ходе работы была разработана частично-ассоциативная система памяти с обратной записью и выбором вытесняемой строки по алгоритму PLRU.

Разработана структурная схема системы памяти, функциональная схема памяти тегов, алгоритм работы устройства управления и соответствующий ему граф переходов автомата, входящего в состав устройства управления.

В ходе разработки проекта на языке Verilog была реализована система памяти, состоящая из памяти данных, памяти тегов и устройства управления и интерфейсов CPU и RAM.

Разработанная система памяти (за исключением интерфейса с оперативной памятью) была отлажена и протестирована и в итоге соответствует требованиям технического задания.

Цель работы достигнута.