**IP ядро**

**FIFO на основе DDR памяти**

**c MIG Native Interface**

**Оглавление**

[**1.** **Описание IP ядра** 3](#_Toc61685608)

[**2.** **Описание алгоритма блока управления памятью** 7](#_Toc61685609)

[**3.** **Описание временных диаграмм и блок схемы блока управления памятью** 10](#_Toc61685610)

[**3.1.** **Старт работы блока** 10](#_Toc61685611)

[**3.2.** **Начало цикла записи (x4)** 11](#_Toc61685612)

[**3.3.** **Начало цикла записи (x2)** 12](#_Toc61685613)

[**3.4.** **Выдача команд на запись** 13](#_Toc61685614)

[**3.5.** **Окончание цикла записи** 14](#_Toc61685615)

[**3.6.** **Выдача команд на считывание** 15](#_Toc61685616)

[**3.7.** **Начало цикла считывания (x4)** 16](#_Toc61685617)

[**3.8.** **Начало цикла считывания (x2)** 17](#_Toc61685618)

[**3.9.** **Окончание цикла считывания** 18](#_Toc61685619)

# **Описание IP ядра**

Ядро состоит из трех частей: входного FIFO, блока работы с памятью и выходного FIFO. Блок работы с памятью постоянно проверяет наличие данные во входном FIFO. Если данные присутствуют и в памяти есть место, то данные считываются из FIFO и переносятся в память. Блок управления памятью также постоянно проверяет наличие свободного места в выходном FIFO. Если FIFO не полное и в памяти есть данные, они переносятся из памяти в FIFO. Блок управления памятью отвечает за работу с MIG IP, формирование адресов чтения и записи и подсчет числа слов в памяти. Внешний вид ядра представлен на рис. 1.1, блок схема на рис. 1.2, в табл. 1.1 описаны входные и выходные порты.

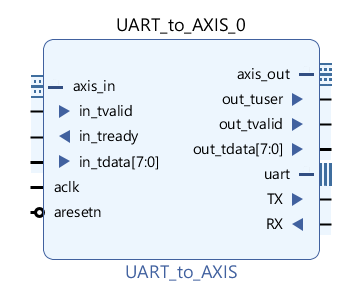


Рис. 1.1 Внешний вид IP ядра

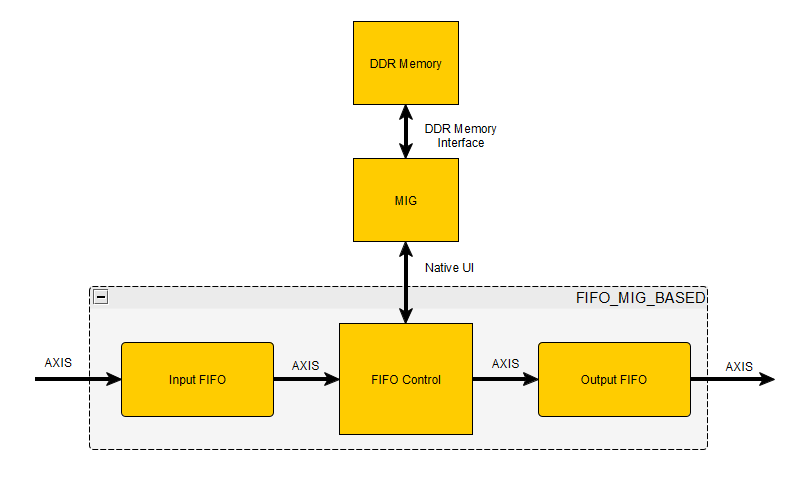


Рис. 1.2 Блок схема ядра

Таблица 1.1 Назначение портов IP ядра

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| aclk | тактовый сигнал |
| aresetn | синхронный сигнал сброса, активный низкий уровень |
| in\_tdata | данные для записи в FIFO |
| in\_tvalid | данные для записи установлены на шине in\_tdata |
| in\_tready | ядро готово получить новые данные |
| out\_tdata | выходные данные FIFO |
| out\_tvalid | данные на шинах out\_tdata установлены |
| out\_tready | ядро, следующее за FIFO готово принять данные |
| init\_calib | инициализация памяти завершена |
| app\_\* | сигналы MIG Native Interface (см. UG586 стр. 92) |

Ядро является конфигурируемым. На рис. 1.2 представлено окно настроек, в табл. 1.2 представлено описание настраиваемых параметров. В табл. 1.3 представлено описание состояний конечного автомата блока FIFO Control.

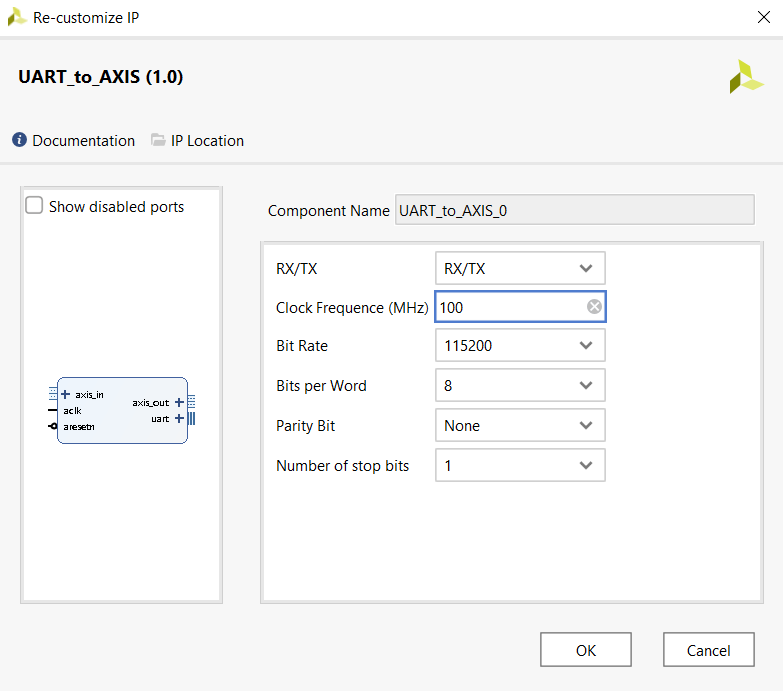
****

Рис. 1.2 Окно настроек IP ядра

Таблица 1.2 Описание настраиваемых параметров

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| PHY to UI Rate | отношение тактовой частоты работы PHY MIG к частоте работы интерфейса MIG (x2 или x4) |
| Max Burst Len | максимальная длина пачки слов за один цикл записи или считывания |
| RW Delay Value | задержка после цикла чтения или записи в тактах aclk |
| Base Address | начальный адрес памяти (в словах памяти) |
| Memory Size | максимальное количество слов в памяти |
| MIG Data Port Size | размер портов записи и считывания данных ядра MIG |
| IO FIFO Depth | глубина входного и выходного FIFO |

От этих параметров зависят внутренние константы блока:

* Размер входных и выходных портов FIFO:

Data\_Width = app\_wdf\_data при x4

Data\_Width = app\_wdf\_data\*2 при x2

* Максимальный адрес:

MAX\_ADDR = Base Address + (Memory Size – 1) \* 8

* Общая глубина ядра в словах:

Total Depth = Memory Size + 2 \* IO FIFO Depth

Таблица 1.3 Описание состояний автомата

|  |  |
| --- | --- |
| **Состояние** | **Назначение** |
| INIT | ожидание конца инициализации |
| CHECK\_WR | проверка возможности записи в память |
| CHECK\_RD | проверка возможности считывания из памяти |
| WRITE | запись в память |
| READ | считывание из памяти |
| DELAY\_WR | задержка после записи |
| DELAY\_RD | задержка после считывания |

# **Описание алгоритма блока управления памятью**

После сигнала выхода из состояния сброса блок ожидает окончания инициализации DDR памяти. На окончание инициализации указывает входной INIT\_CALIB. Далее блок переходит в режим постоянной проверки данных доступных для записи в память и считывания из памяти.

Если во входном fifo, есть данные и в памяти есть место, то блок переходит в режим записи данных. Число записываемых слов определяется, как наименьшее из числа слов в fifo, числа доступных мест в памяти и максимального числа записываемых за раз слов (MAX\_BURST\_LEN).

Если память готова принять слово данных, то данные считываются из fifo, передаются в память и число слов для записи уменьшается. Иначе блок ожидает готовности памяти для записи слова данных.

Параллельно с этим блок записывает команды на запись. Команда может быть записана, если память готова ее принять и число оставшихся команда меньше числа оставшихся слов для записи. Запись команды только после записи слова данных не необходима, но есть ограничение на задержку записи данных после записи команды. Чтобы не гарантировать это требование команда всегда записывается после данных. Если память не готова принять команду, блок ожидает ее готовности. После записи команды уменьшается число записываемых команд и обновляется адрес записи.

После записи всех команд блок переходит в состояние задержки после записи. Входное fifo имеет выход, указывающий количество слов внутри него. Данный выход обновляется с задержкой в несколько тактов. Чтобы следующий цикл записи получил правильное число доступных слов в fifo вводится задержка после цикла записи.

После задержки блок переходит к проверке возможности считывания из памяти.

Если в выходном fifo есть свободное место и в памяти есть записанные слова, то блок переходит в состояние считывания данных из памяти. Число считываемых слов определяется, как наименьшее из числа доступных мест в выходном fifo, числа слов в памяти и максимального числа считываемых за раз слов (MAX\_BURST\_LEN).

Если память не готова принять команду на считывание, блок ожидает ее готовности. После записи команды уменьшается число записываемых команд и обновляется адрес считывания.

Параллельно с некоторой задержкой память возвращает считанные данные, которые записываются в выходное fifo. fifo всегда готово принять данные, так как было проверено, что в нем есть свободные места.

После считывания всех слов блок переходит в состояние задержки после считывания. Выходное fifo имеет выход, указывающий количество слов внутри него. Данный выход обновляется с задержкой в несколько тактов. Чтобы следующий цикл считывания получил правильное число доступных мест в fifo вводится задержка после цикла записи.

После задержки блок переходит к проверке возможности записи в память.

Блок схема алгоритма представлена на рис. 3.1.

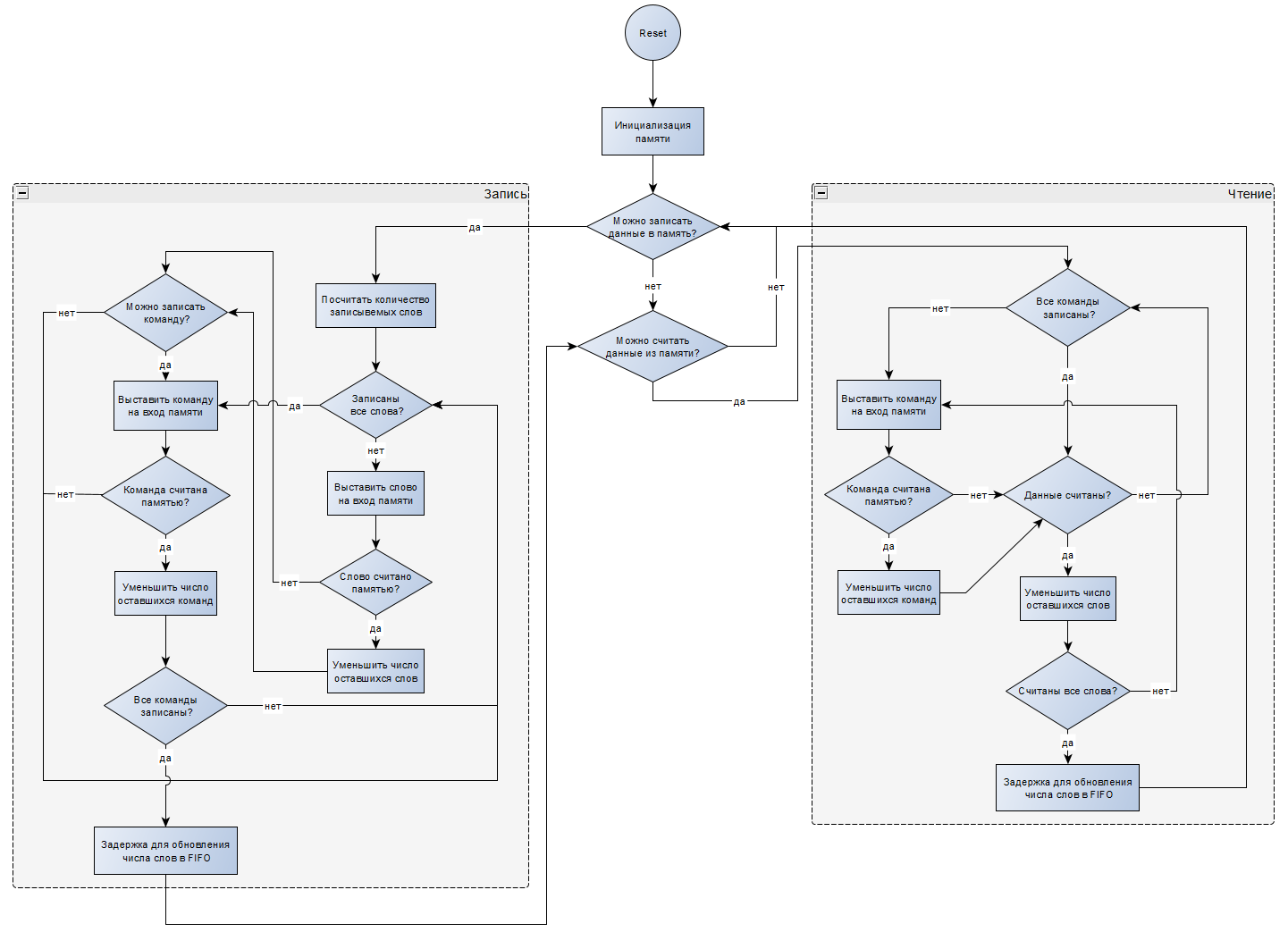
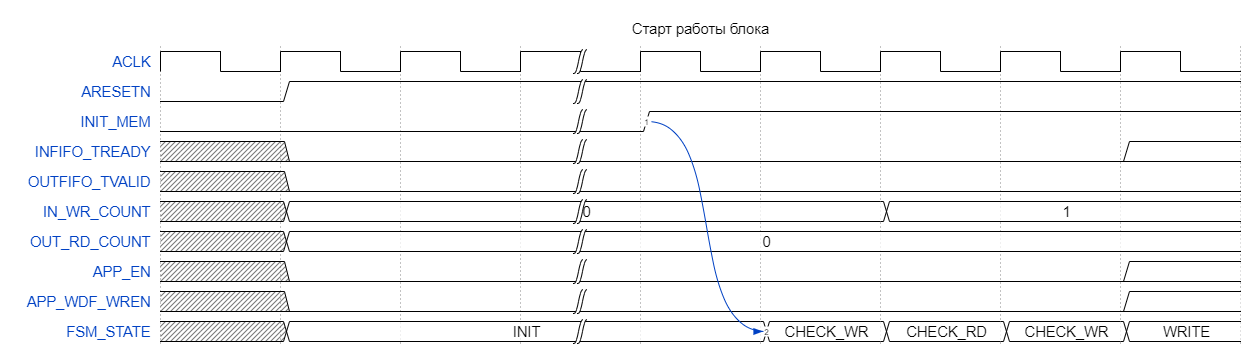


Рисунок 3.1 Блок схема алгоритм

1. **Описание временных диаграмм и блок схемы блока управления памятью**
   1. **Старт работы блока**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| ARESETN | сигнал сброса с синхронным активным низким уровнем |
| INIT\_CALIB | сигнал завершения калибровки от памяти |
| INFIFO\_TREADY | готовность принять из входного FIFO |
| OUTFIFO\_TVALID | запись данных в выходное FIFO |
| APP\_EN | запись команды в память |
| APP\_WDF\_WREN | запись данных в память |
| IN\_WR\_COUNT | число слов во входном FIFO |
| OUT\_RD\_COUNT | число слов во выходном FIFO |

В всех состояниях кроме WRITE и READ сигналы INFIFO\_TREADY, OUTFIFO\_TVALID, APP\_EN и APP\_WDF\_WREN имеют неактивные значения. После сброса и инициализации памяти проверяем возможность записи или считывания и, если проверка успешна, переходим в состояние WRITE или READ.



* 1. **Начало цикла записи (x4)**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| WR\_COUNT | количество записываемых слов |
| APP\_WDF\_RDY | память готова принять данные |
| APP\_WDF\_END | конец слова данных (используется в x2) |
| MEM\_WR\_COUNT | число свободных мест в памяти |
| MEM\_RD\_COUNT | число слова в памяти |

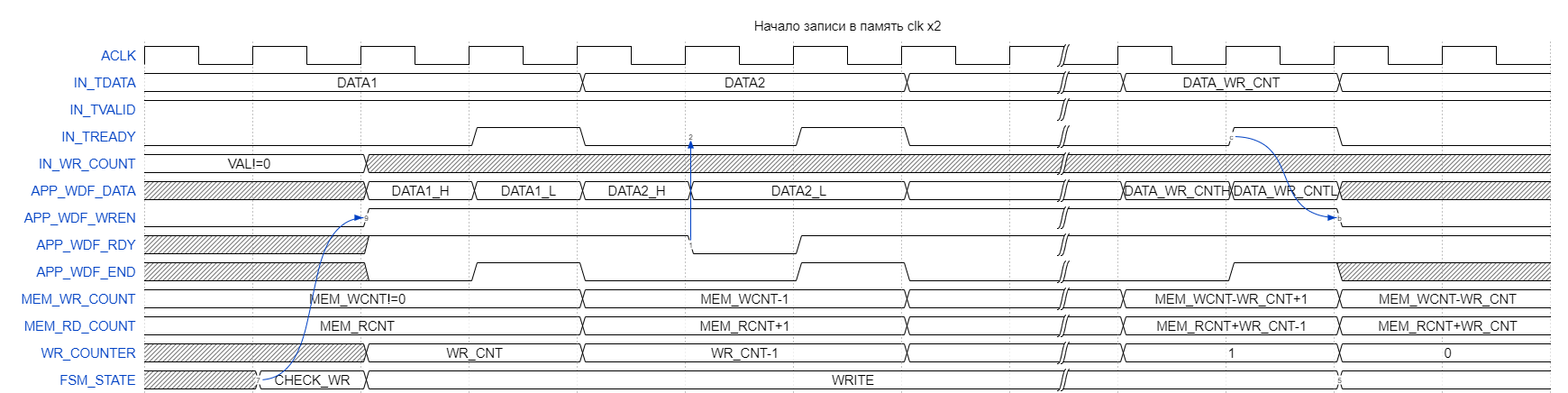
Если память не готова принять данные (APP\_WDF\_RDY = 0), то INFIFO\_TREADY = 0, счетчики не обновляются и блок ничего не делает. Если остается одно слово для записи, на следующем такте IN\_TREADY и APP\_WDF\_WREN сбрасываются. WR\_CNT – меньшее из MEM\_WR\_COUNT, IN\_WR\_COUNT и MAX\_BURST\_LEN.



* 1. **Начало цикла записи (x2)**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| WR\_COUNT | количество записываемых слов |
| IN\_TREADY | считать данные из FIFO |
| APP\_WDF\_RDY | память готова принять данные |
| APP\_WDF\_END | конец слова данных (используется в x2) |
| MEM\_WR\_COUNT | число свободных мест в памяти |
| MEM\_RD\_COUNT | число слова в памяти |

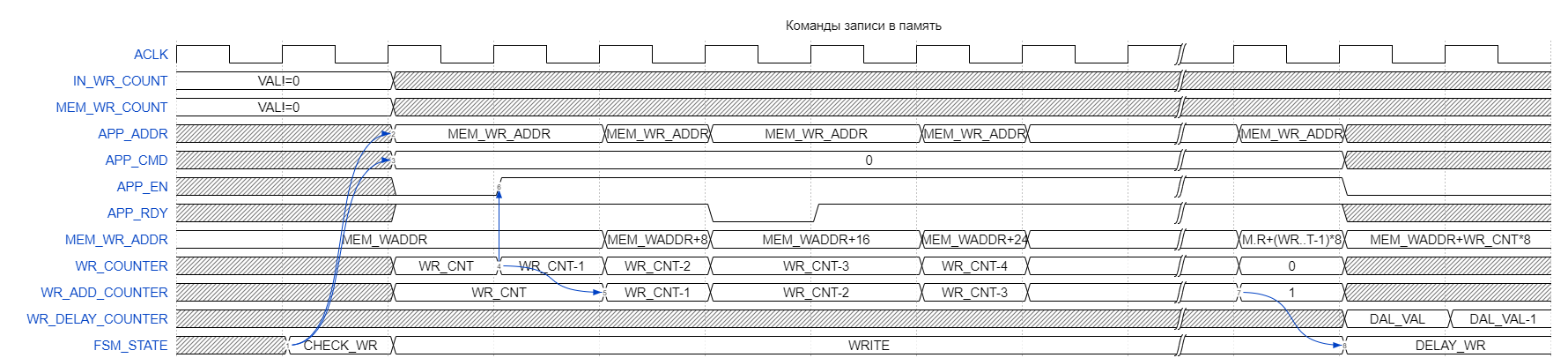
Если память не готова принять данные (APP\_WDF\_RDY = 0), то INFIFO\_TREADY = 0, счетчики не обновляются и блок ничего не делает. В память APP\_WDF\_DATA данные выдаются полусловами, сначала старшие, потом младшие. При записи младшего полуслова выставляется сигналы APP\_WDF\_END и IN\_TREADY и обновляются счетчики числа слов. Если остается одно слово для записи и выстелен IN\_TREADY, на следующем такте APP\_WDF\_WREN сбрасывается. WR\_CNT – меньшее из MEM\_WR\_COUNT, IN\_WR\_COUNT и MAX\_BURST\_LEN.



* 1. **Выдача команд на запись**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| APP\_ RDY | память готова принять команду |
| WR\_COUNT | количество записываемых слов |
| WR\_ADD\_COUNTER | количество записываемых команд |
| WR\_DELAY\_COUNTER | счетчик тактов задержки после записи |
| MEM\_WR\_ADDR | адрес записи |

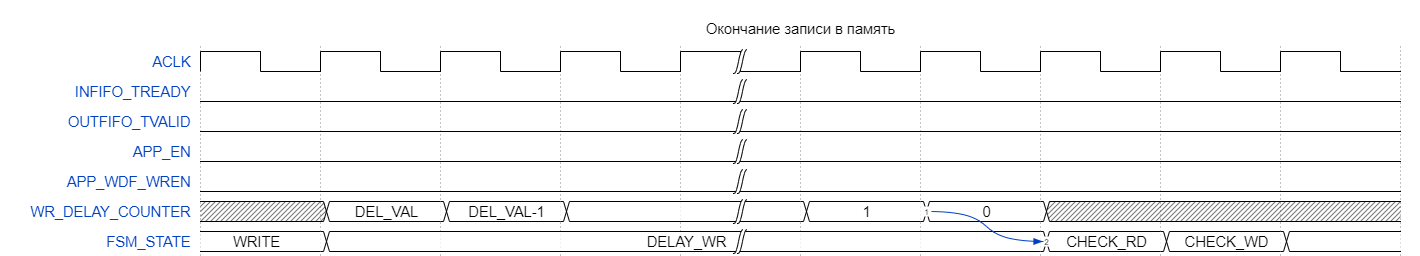
Если память не готова принять команду (APP\_RDY = 0), счетчики не обновляются и блок ничего не делает. Сигнал записи команды APP\_EN устанавливается только, если WR\_COUNT < WR\_ADD\_COUNTER, то есть слов данных загружено больше, чем команд. После загрузки команды адрес увеличивается на восемь, так как в DDR память MIG загружает слово, как последовательность из 8 слов меньшего размера. На вход APP\_ADDR устанавливается текущий адрес записи. Если остается одна команда для записи, на следующем такте APP\_EN сбрасывается. Блок переходит в состояние задержки после записи.



* 1. **Окончание цикла записи**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| INFIFO\_TREADY | готовность принять из входного FIFO |
| OUTFIFO\_TVALID | запись данных в выходное FIFO |
| APP\_EN | запись команды в память |
| APP\_WDF\_WREN | запись данных в память |
| WR\_DELAY\_COUNTER | счетчик тактов задержки после записи |

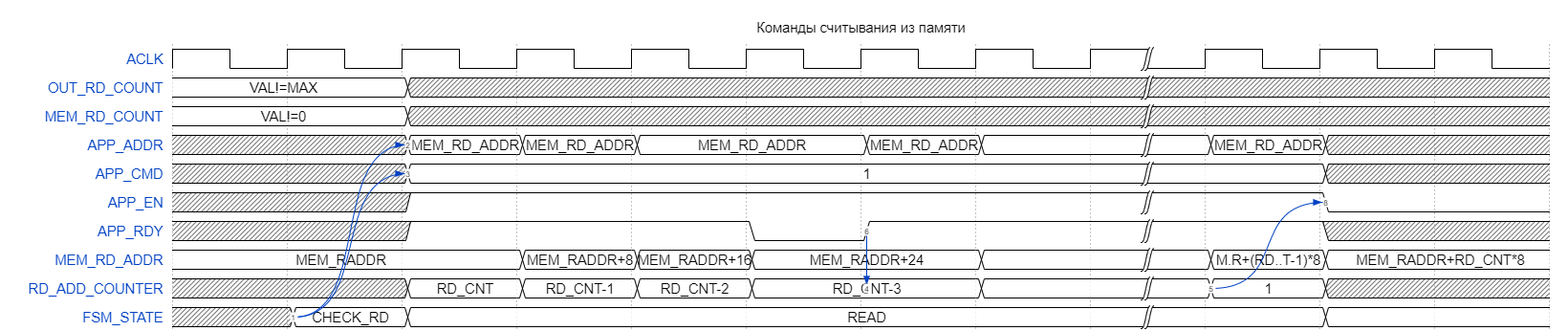
Сигналы INFIFO\_TREADY, OUTFIFO\_TVALID, APP\_EN и APP\_WDF\_WREN имеют неактивные значения. После задержки в DEL\_VAL тактов блок переходит в состояние проверки возможности считывания CHECK\_RD.



* 1. **Выдача команд на считывание**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| APP\_ RDY | память готова принять команду |
| RD\_ADD\_COUNTER | количество команд на считывание |
| MEM\_RD\_ADDR | адрес считывания |

Если память не готова принять команду (APP\_RDY = 0), счетчики не обновляются и блок ничего не делает. После загрузки команды адрес увеличивается на восемь, так как в DDR память MIG загружает слово, как последовательность из 8 слов меньшего размера. На вход APP\_ADDR устанавливается текущий адрес записи. Если остается одна команда для записи, на следующем такте APP\_EN сбрасывается.



* 1. **Начало цикла считывания (x4)**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| RD\_COUNTER | количество считываемых слов |
| APP\_RD\_DATA\_VALID | данные на шине APP\_RD\_DATA действительные |
| APP\_RD\_DATA\_END | конец слова данных (используется в x2) |
| MEM\_WR\_COUNT | число свободных мест в памяти |
| MEM\_RD\_COUNT | число слова в памяти |
| OUT\_RD\_COUNT | число свободных мест в выходном fifo |

Если память не выдает данные (APP\_RD\_DATA\_VALID = 0), то OUTFIFO\_TVALID = 0, счетчики не обновляются и блок ничего не делает. Если остается одно слово для записи, на следующем такте OUTFIFO\_TVALID сбрасывается и блок переходит в состояние задержки после считывания. RD\_CNT – меньшее из MEM\_RD\_COUNT, OUT\_RD\_COUNT и MAX\_BURST\_LEN.



* 1. **Начало цикла считывания (x2)**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| RD\_COUNTER | количество считываемых слов |
| APP\_RD\_DATA\_VALID | данные на шине APP\_RD\_DATA действительные |
| APP\_RD\_DATA\_END | конец слова данных (используется в x2) |
| MEM\_WR\_COUNT | число свободных мест в памяти |
| MEM\_RD\_COUNT | число слова в памяти |
| OUT\_RD\_COUNT | число свободных мест в выходном fifo |

Если память не выдает данные (APP\_RD\_DATA\_VALID = 0) и это не последняя часть слова (APP\_RD\_DATA\_END = 0), то OUTFIFO\_TVALID = 0 и счетчики не обновляются. Память выдает данные полусловами, сначала старшее слово, потом младшее. Целое слово записывается в fifo по сигналу OUTFIFO\_TVALID. Счетчики обновляются при APP\_RD\_DATA\_END = 1. Если остается одно слово для записи, на следующем такте блок переходит в состояние задержки после считывания. RD\_CNT – меньшее из MEM\_RD\_COUNT, OUT\_RD\_COUNT и MAX\_BURST\_LEN.



* 1. **Окончание цикла считывания**

|  |  |
| --- | --- |
| **Сигнал** | **Назначение** |
| INFIFO\_TREADY | готовность принять из входного FIFO |
| OUTFIFO\_TVALID | запись данных в выходное FIFO |
| APP\_EN | запись команды в память |
| APP\_WDF\_WREN | запись данных в память |
| RD\_DELAY\_COUNTER | счетчик тактов задержки после считывания |

Сигналы INFIFO\_TREADY, OUTFIFO\_TVALID, APP\_EN и APP\_WDF\_WREN имеют неактивные значения. После задержки в DEL\_VAL тактов блок переходит в состояние проверки возможности записи CHECK\_WD.

