

**ІР ядро**  
**FIFO на основе DDR пам'яті**  
**с MIG Native Interface**

## Оглавление

1. Описание IP ядра.....	3
2. Описание алгоритма блока управления памятью.....	6
3. Описание временных диаграмм блока управления памятью.....	9
3.1. Старт работы блока .....	9
3.2. Начало цикла записи (x4) .....	10
3.3. Начало цикла записи (x2) .....	11
3.4. Выдача команд на запись.....	12
3.5. Окончание цикла записи .....	13
3.6. Выдача команд на считывание.....	14
3.7. Начало цикла считывания (x4).....	15
3.8. Начало цикла считывания (x2).....	16
3.9. Окончание цикла считывания .....	17
4. Внутреннее устройство блока управления памятью.....	18

## 1. Описание IP ядра

Ядро состоит из трех частей: входного FIFO, блока работы с памятью и выходного FIFO. Блок работы с памятью постоянно проверяет наличие данных 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 выходное 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.

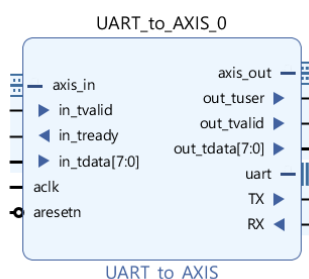


Рис. 1.1 Внешний вид IP ядра

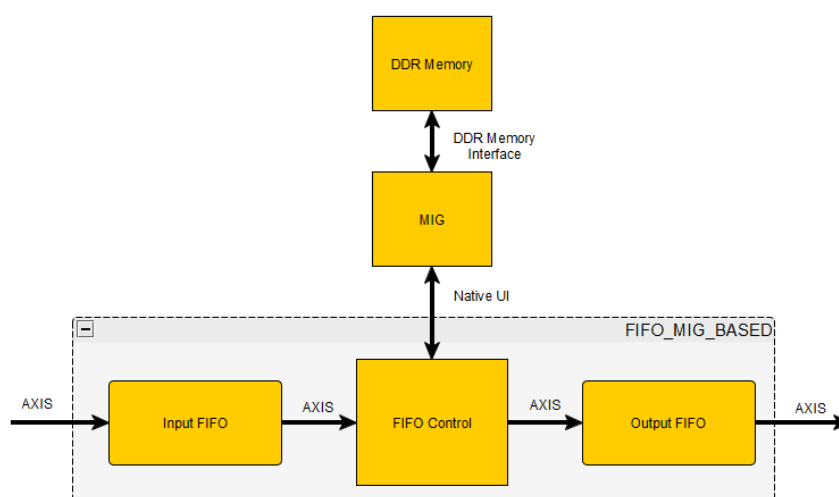


Рис. 1.2 Блок схема ядра

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.

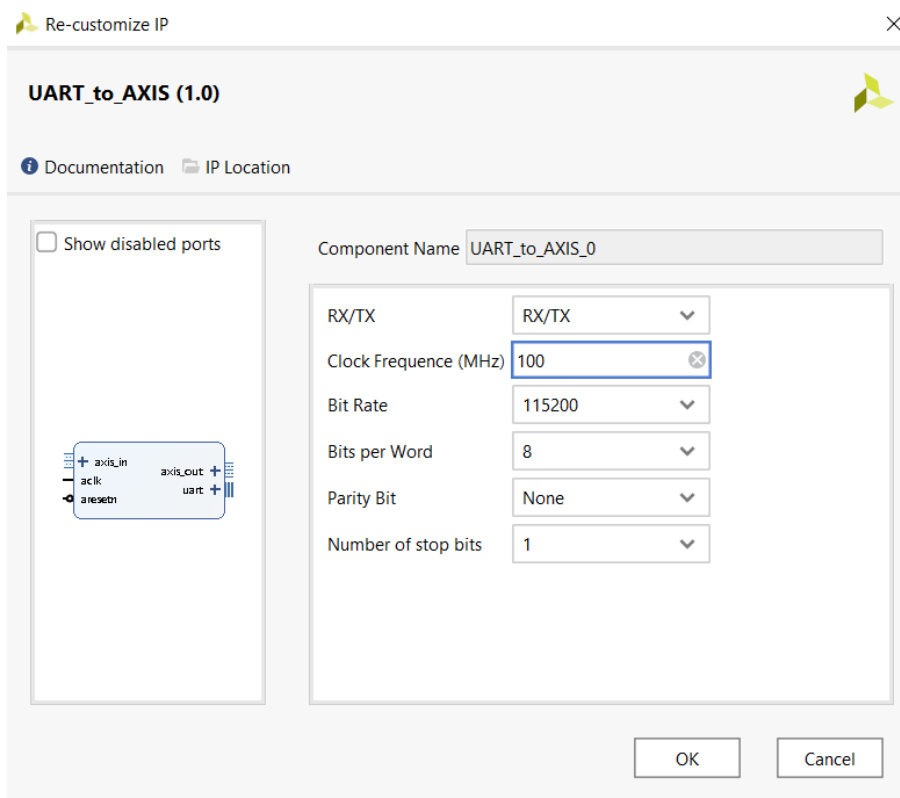


Рис. 1.2 Окно настроек IP ядра

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
MIG Addr Port Size	размер порта адреса ядра MIG
IO FIFO Depth	глубина входного и выходного FIFO

От этих параметров зависят внутренние константы блока:

- Размер входных и выходных портов FIFO:  
 $\text{Data\_Width} = \text{app\_wdf\_data}$  при x4  
 $\text{Data\_Width} = \text{app\_wdf\_data} * 2$  при x2
- Максимальный адрес:  
 $\text{MAX\_ADDR} = \text{Base Address} + (\text{Memory Size} - 1) * 8$
- Общая глубина ядра в словах:  
 $\text{Total Depth} = \text{Memory Size} + 2 * \text{IO FIFO Depth}$

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

## **2. 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 выдает команды на запись. Команда может быть записана, если память готова ее принять и число оставшихся команд меньше числа оставшихся слов для записи. Запись команды строго после записи слова данных не обязательна, но есть ограничение на задержку записи данных после записи команды. Чтобы 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 свободное место.

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. 2.1.

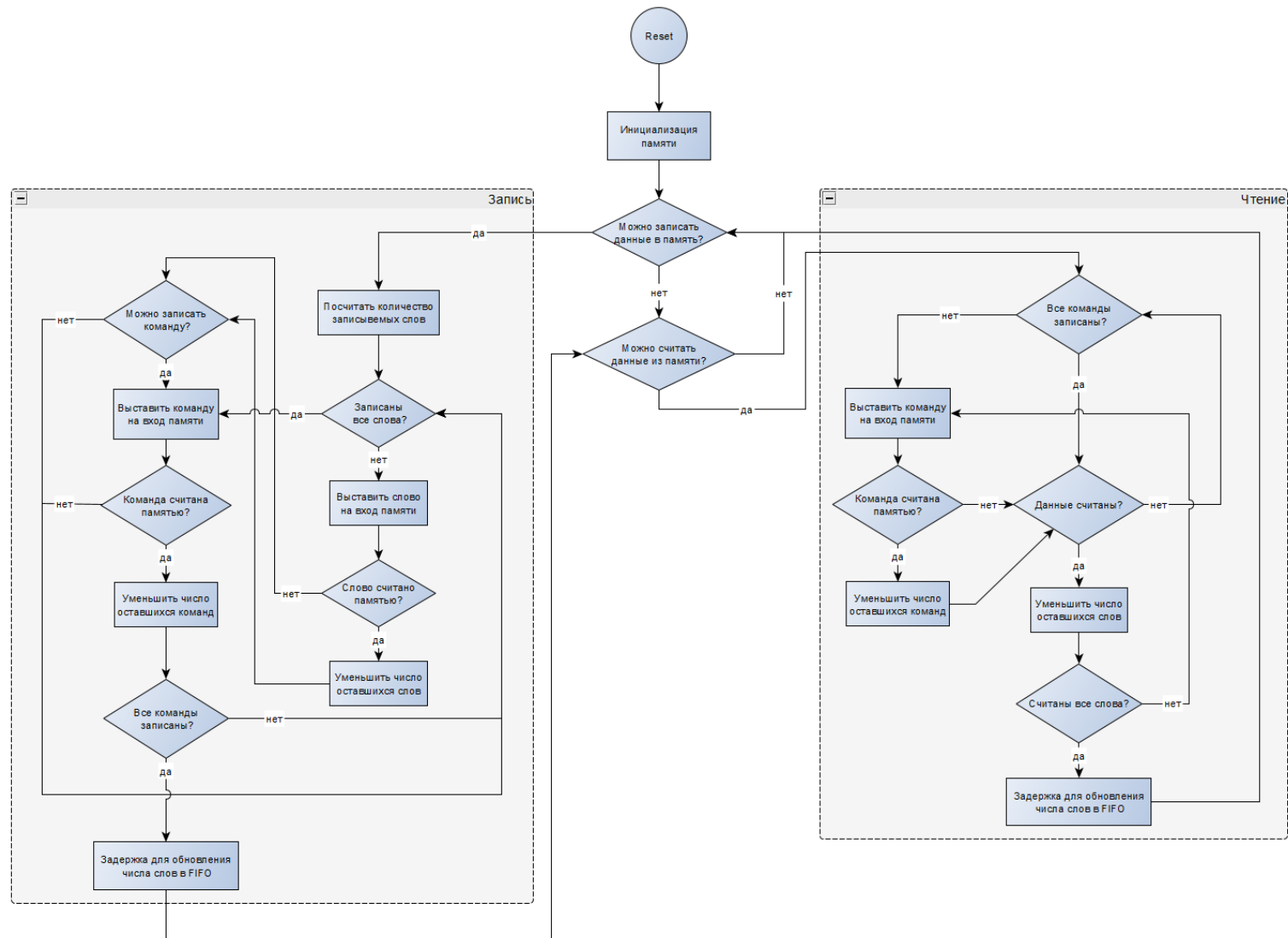


Рисунок 3.1 Блок схема алгоритма

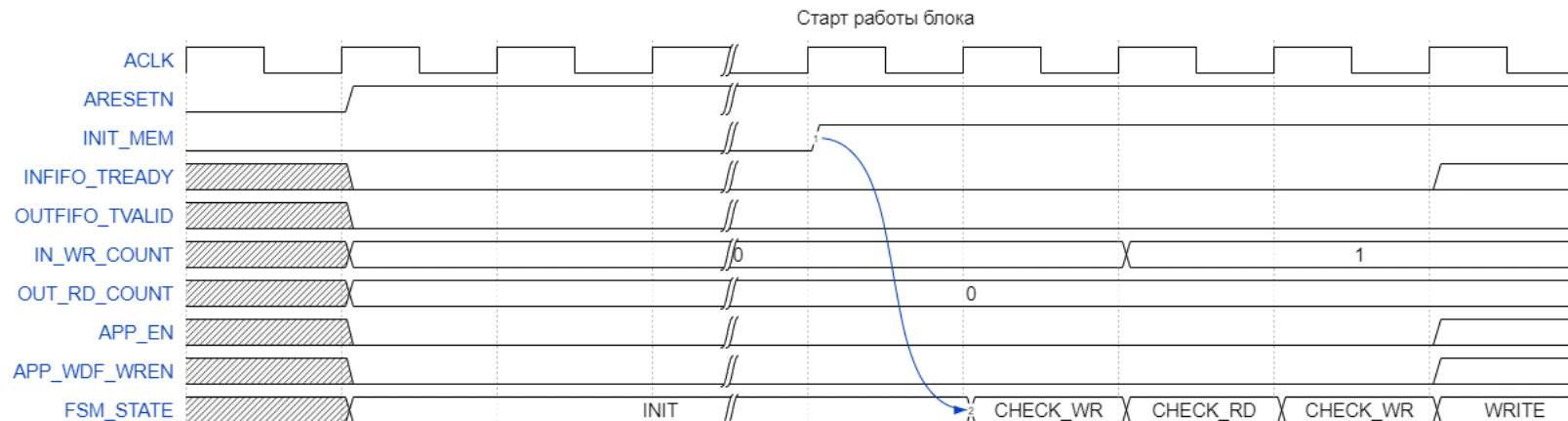


### 3. Описание временных диаграмм блока управления памятью

#### 3.1. Старт работы блока

Сигнал	Назначение
ARESETN	сигнал сброса с синхронным активным низким уровнем
INIT_CALIB	сигнал завершения калибровки памяти
INFIFO_TREADY	готовность принять данные 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 в выходном FIFO

Во 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 проверяется возможность записи или считывания и, если проверка успешна, осуществляется переход в состояние WRITE или READ.



### 3.2. 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 слов в памяти

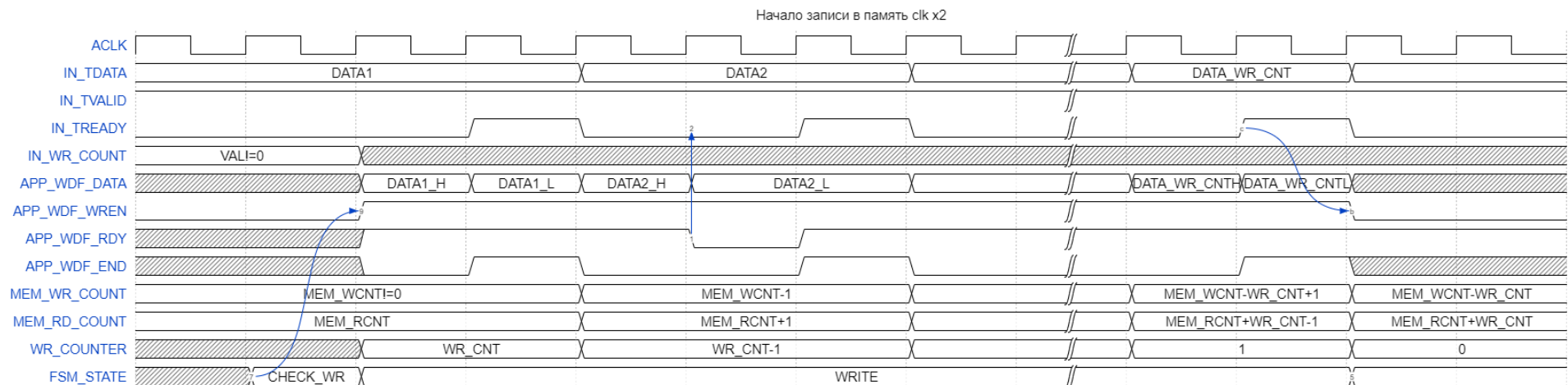
Если память не готова принять данные ( $APP\_WDF\_RDY = 0$ ), то  $INFIFO\_TREADY = 0$ , счетчики не обновляются и блок ничего не делает. Если остается одно слово для записи, на следующем такте  $IN\_TREADY$  и  $APP\_WDF\_WREN$  сбрасываются.  $WR\_CNT$  – меньшее из  $MEM\_WR\_COUNT$ ,  $IN\_WR\_COUNT$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



### 3.3. 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 слов в памяти

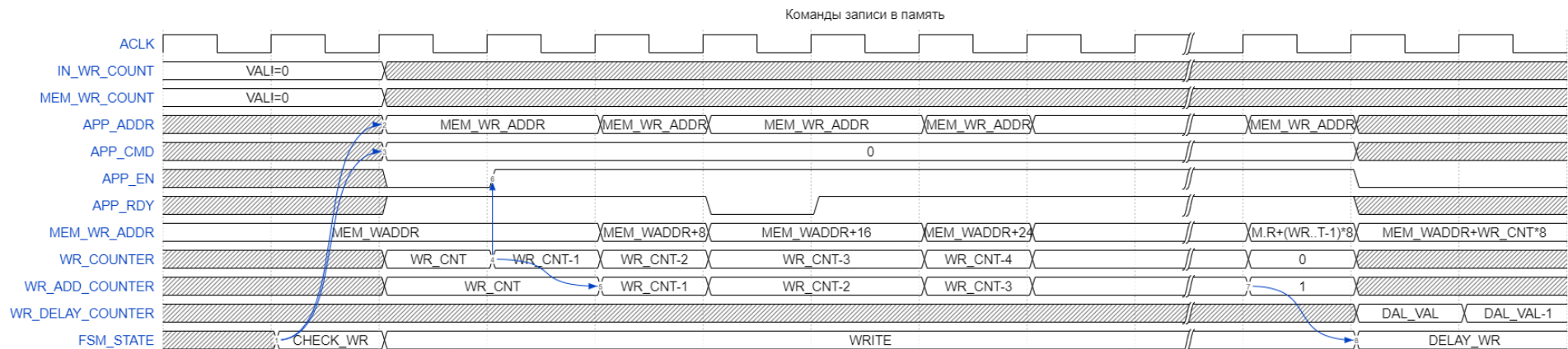
Если память не готова принять данные ( $APP\_WDF\_RDY = 0$ ), то  $INFIFO\_TREADY = 0$ , счетчики не обновляются и блок ничего не делает. В память данные ( $APP\_WDF\_DATA$ ) выдаются полусловами, сначала старшие, потом младшие. При записи младшего полуслова выставляются сигналы  $APP\_WDF\_END$  и  $IN\_TREADY$  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  $MAX\_BURST\_LEN$ .



### 3.4. 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

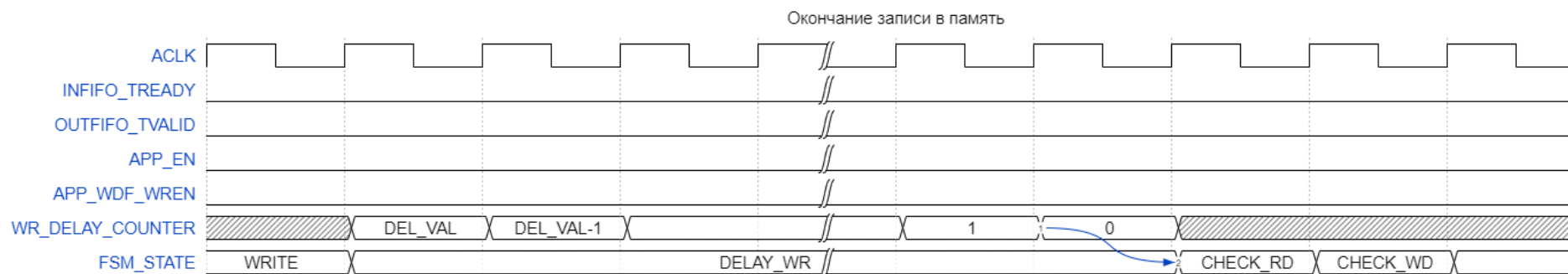
Если память не готова принять команду ( $APP\_RDY = 0$ ), счетчики не обновляются и блок ничего не делает. Сигнал записи команды  $APP\_EN$  устанавливается только, если  $WR\_COUNT < WR\_ADD\_COUNTER$ , то есть количество данных загружено больше, чем команд. После загрузки команды адрес увеличивается 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.



### 3.5. 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

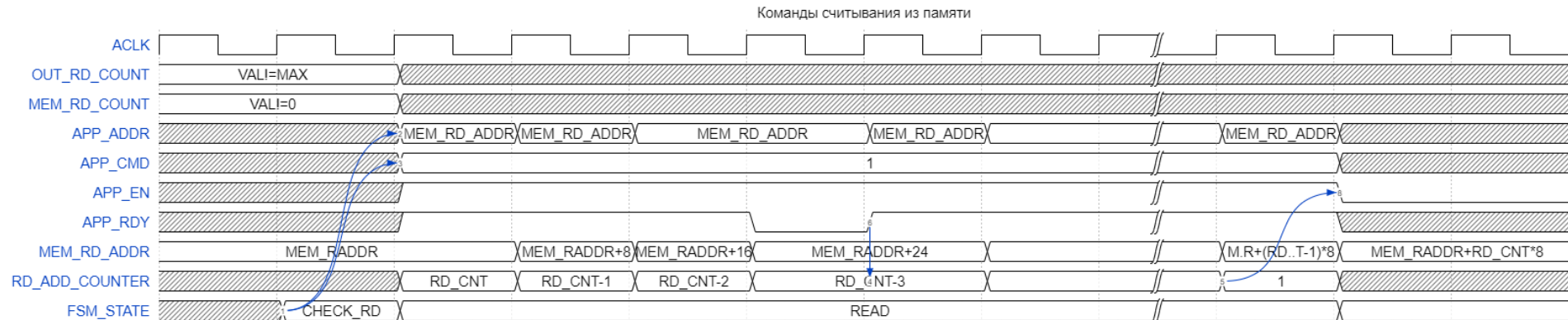
Сигналы INFIFO\_TREADY, OUTFIFO\_TVALID, APP\_EN и APP\_WDF\_WREN имеют неактивные значения. После задержки в DEL\_VAL тактов блок переходит в состояние проверки возможности считывания CHECK\_RD.



### 3.6. Выдача команд на считывание

Сигнал	Назначение
APP_RDY	память готова принять команду
RD_ADD_COUNTER	количество команд на считывание
MEM_RD_ADDR	адрес считывания

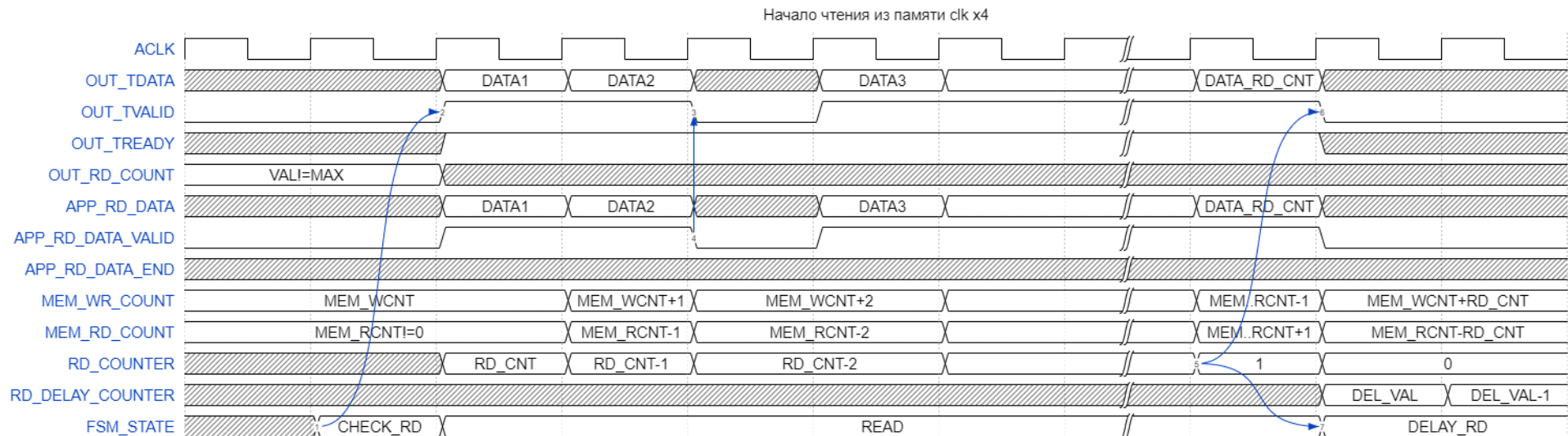
Если память не готова принять команду ( $APP\_RDY = 0$ ), счетчики не обновляются и блок ничего не делает. После загрузки команды адрес увеличивается на восемь, так как в DDR память MIG загружает слово, как последовательность из 8 слов меньшего размера. На вход APP\_ADDR устанавливается текущий адрес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



### 3.7. 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 слов в памяти
OUT_RD_COUNT	число свободных мест в выходном FIFO

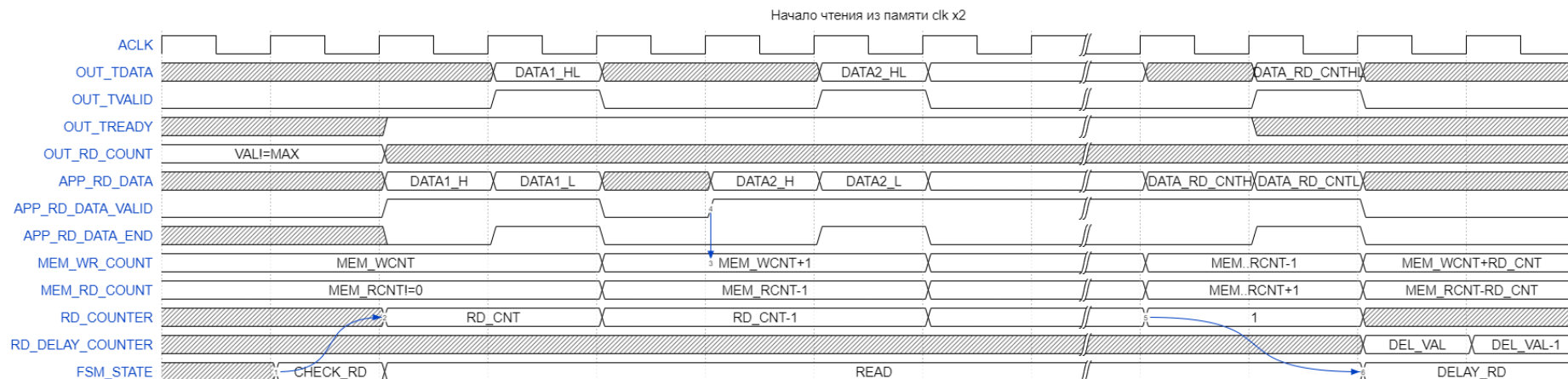
Если память не выдает данные ( $APP\_RD\_DATA\_VALID = 0$ ), то  $OUTFIFO\_TVALID = 0$ , счетчики не обновляются и блок ничего не делает. Если остается одно слово для записи, на следующем такте  $OUTFIFO\_TVALID$  сбрасывается и блок переходит в состояние задержки после считывания.  $RD\_CNT$  – меньшее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



### 3.8. 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 слов в памяти
OUT_RD_COUNT	число свободных мест в выходном FIFO

Если память не выдает данные ( $APP\_RD\_DATA\_VALID = 0$ ) или 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  $MAX\_BURST\_LEN$ .

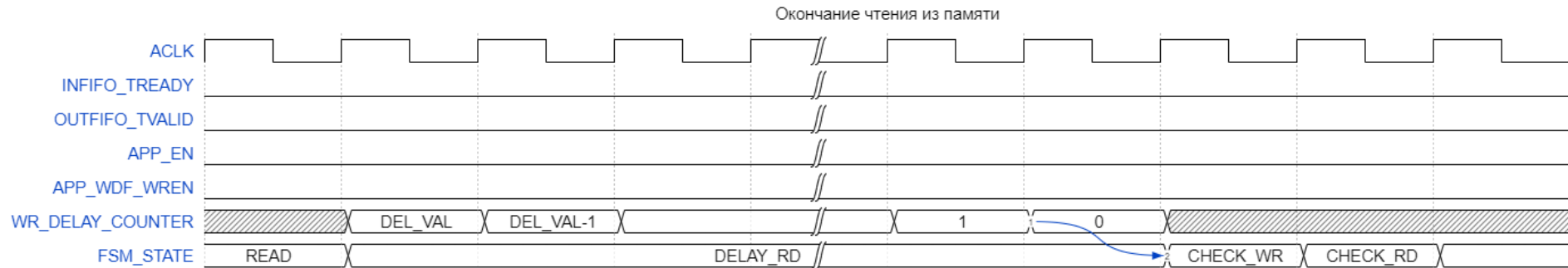




### 3.9. Окончание цикла считывания

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.



## 4. Внутреннее устройство блока управления памятью

