РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ

Мультиклеточный процессор MultiClet R1

Мультиклеточные процессоры:

MCp042R100102-LQ256

Аннотация

Микропроцессор MCp042R100102-LQ256 имеет в своем составе мультиклеточное процессорное ядро. Ядро МП содержит 4 клетки - когерентных процессорных блоков, объединенных интеллектуальной коммутационной средой. МП имеют динамически реконфигурируемое ядро, распределение вычислительных ресурсов (клеток) по задачам производится согласно алгоритму заложенному в исполняемой программе. При этом один и тот же код может выполняться любым количеством и комбинацией клеток. Мультиклеточный процессор предназначен для решения широкого круга задач управления и цифровой обработки сигналов в приложениях, требующих минимального энергопотребления и высокой производительности.

Особенности:

- Число клеток — 4
- Разрядность процессора — 32/64 бита
- ПП(на кристалле) — 256 Кбайт
- ПД(на кристалле) — 256 Кбайт
- блок операций над числами с плавающей запятой (в каждой клетке)
- Тактовая частота — до 100 МГц;
- Производительность процессора — 24 MFLOPS/МГц;

Общие характеристики:

- Корпус — LQFP-256
- Условия эксплуатации – (-60 . . . +125°C)
- Максим. потребл. мощность:
- ядра — 1,05Вт на частоте 100МГц
- выходных каскадов портов — 80мВт
- Напряжение питания (раздельное): ядра — 1,8В, выходных каскадов — 3,3В

Периферийные устройства:

- 2 интерфейса SPI с селектором «ведомых» устройств (в режиме «ведущий»)
- 4 универсальных асинхронных приёмопередатчика UART с FIFO на прием/передачу
- 2 интерфейса I2C (один «ведомый» и один «ведущий»)
- интерфейс I2S («ведомый»/«ведущий»)
- Ethernet MAC контроллер 10/100Мб/с
- USB 2.0 HS (device), внешний интерфейс ULPI
- часы реального времени с календарем
- 4 таймера общего назначения
- 6 портов ввода-вывода, общее количество вводов-выводов – 166
- 4-х канальный контроллер ШИМ
- сторожевой таймер
- АЦП(16 бит, 48 Квыб/с, 8 каналов)
- ЦАП(12 бит, 100 Мвыб/с, 1 канал)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Содержание

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

1 Условные обозначения и сокращения

1.1 Список сокращений

СФБ — сложно-функциональный блок;

МП — микропроцессор;

ПО — программное обеспечение;

MSb — старший значащий бит;

MSB — старший значащий байт;

LSb — младший значащий бит;

LSB — младший значащий байт;

ОЗУ — оперативное запоминающее устройство;

ПУ — периферийное устройство;

ЦПУ — центральное процессорное устройство;

ФБО — физический блок ОЗУ;

ПП — память программ;

ПД — память данных;

ФНЧ — фильтр нижних частот;

КДВП — контроллер доступа к внешней памяти;

ПДП — прямой доступ к памяти;

РОН — регистр(ы) общего назначения;

1.2 Принятые условные обозначения

’1’, ’0’— состояние логической единицы, логического нуля, соответственно;

REG(BIT) — такая запись используется для указания бита в регистре, где REG - название регистра, а BIT - обозначение бита или группы битов в нем. Например, "бит I2CxCR(EN)" означает, что идет указание на бит EN регистра I2CxCR, а

Все права защищены © ОАО «Мультиклет» 2010 — 2015

"I2CxPSC(PSC)" указывает на группу битов PSC. Что бы подробнее узнать об указанных регистрах и битах, надо смотреть описание этих регистров;

PUx, BLOCKx, PUxREG — В обозначении регистров, наименовании ПУ, блоков МП может использоваться символ "x". Это замещение номера, например, есть несколько идентичных периферийных блоков UART с номерами 0,1 и т.д. Для UART0 "x" - 0. Если есть регистр I2CxCR, то для блока I2C0, регистр будет именоваться I2C0CR.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

2 Описание

2.1 Основные технические характеристики

Условное обозначение	**MCp042R100102-LQ256**
Основное функциональное назначение	64-разрядная микро-ЭВМ
Максимальная тактовая частота ядра, МГц	100
Мультиклеточное ядро	4 клетки 32/64 разряда
Блок вычисления чисел с плавающей точкой(соответствует ieee754)	двойной точности
Производительность пиковая	24 MFLOPS/МГц
Встроенная память СОЗУ, Кбайт	512
Внешняя шина памяти, бит	32
Внешняя шина памяти, типы памяти	PROM, SRAM, SDRAM, IO
USB 2.0 HS device, шт.	1
UART, шт.	4
SPI, шт.	2
I2C	1 ведущий, 1 ведомый
I2S	1 ведущий/ведомый (прием/передача)
PWM, каналов	4
Ethernet 10/100, шт.	1
ЦАП, 12 бит, 125Msps, каналов	1
АЦП, 16 бит, 48Ksps, каналов	8
Пользовательские вводы-выводы, шт.	159

Все права защищены © ОАО «Мультиклет» 2010 — 2015

2.2 Структура МП

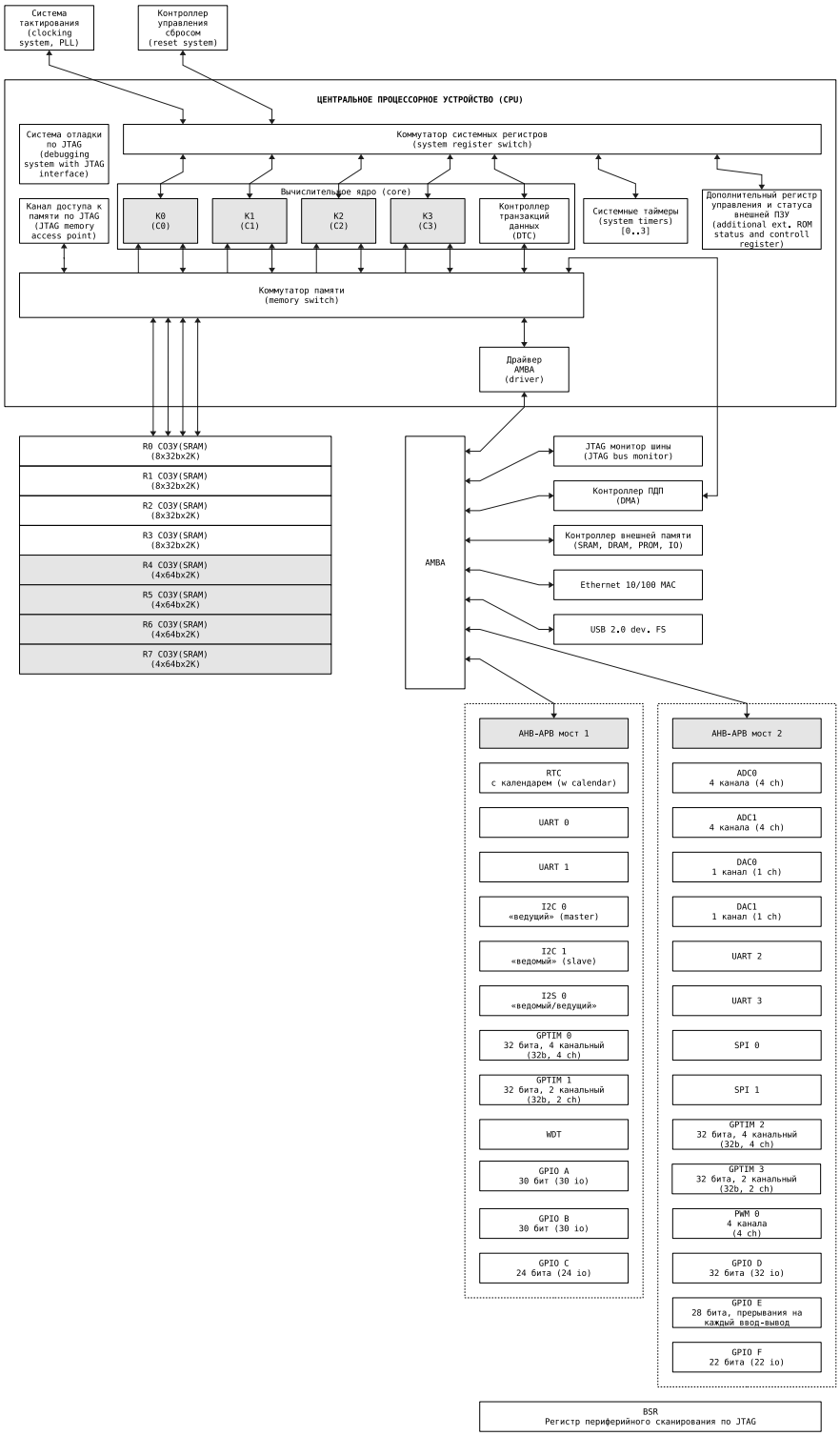

Рис. 1: Общая структура МП

Все права защищены © ОАО «Мультиклет» 2010 — 2015

МП имеет следующие основные части:

- центральное процессорное устройство (ЦПУ);
- периферийные устройства. В качестве шины периферийных устройств используется шина AMBA 2.0;
- оперативные запоминающие устройства типа СОЗУ.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3 Центральное процессорное устройство

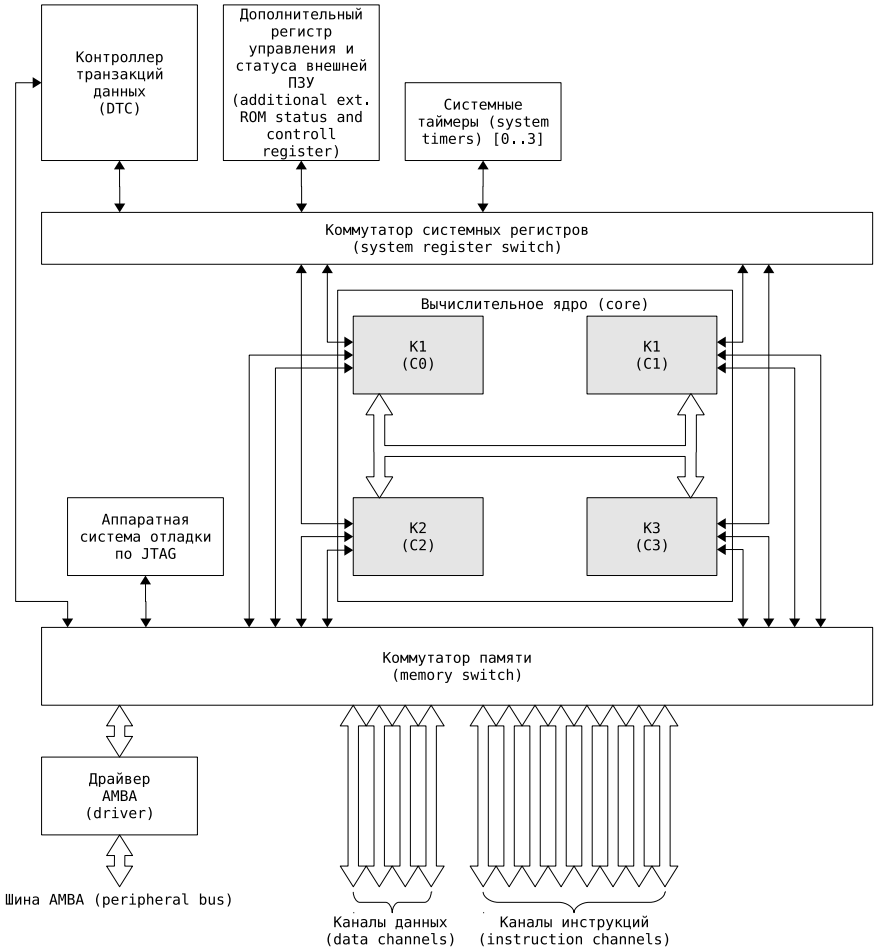

Рис. 2: Структурная схема ЦПУ

ЦПУ в своем составе имеет:

- 4-х клеточное вычислительное ядро;
- коммутатор памяти;
- системные устройства;
- коммутатор системных регстров;
- аппаратную систему отладки ПО.

Структурная схема ядра микропроцессора представлена на рис. 2

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.1 Организация вычислительного процесса

Исполнение процессором любой команды завершается формированием некоторого результата. Использование этого результата последующими исполняемыми командами, образует информационную связь между командой–источником результата и командами–приемниками этого результата. Эта информационная связь может быть как прямой, так и косвенной.

Абсолютно все современные цифровые процессоры, кроме мультиклеточных, используют косвенную связь, при которой результат доступен только после его отчуждения. Возможно всего два способа реализации этой связи.

Первый – запись его командой–источником в общедоступные регистры и/или память, имя которых при использовании задают в поле операнда команды–приемника (фоннеймановская модель процессора). Как следствие, решение задачи подобным процессором достигается пошаговым целенаправленным изменением состояния вычислительной системы. Этот процесс изменения состояний диктуется исполнением последовательности команд строго в заданном порядке, который определяют в процессе программирования.

Второй – адресная рассылка, т.е. запись его в память по адресу в который, при программировании, должна быть помещена команда–приемник (потоковые процессоры). Типичная команда потокового процессора, например, двухместная арифметическая операция, содержит, как правило, поле кода операции, поля двух операндов, одно или несколько полей адресов команд, которым передается результат выполненной операции, и, соответственно, поля номеров операндов, в которых он будет размещен. Команда выбирается и выполняется «по готовности операндов», т.е. тогда, когда в ее поля операндов будут записаны все результаты выполнения других команд, необходимые для ее выполнения. Таким образом, команды в программе могут быть размещены произвольным образом. Принцип исполнения команд потокового процессора «по готовности операндов» позволяет реализовать параллелизм «естественным» образом, т.е. без необходимости решения задачи распараллеливания.

Принципиально иной вид информационной связи между командами – это прямая связь. В этом случае команды именуются и при этом имена должны идентифицировать собственно команду, а не ее местоположение или другие особенности реализации. Доступ к результатам может осуществляться по именам как команд–источников, так команд–приемников. В первом случае используется широковещательная рассылка с последующим поименным отбором, при которой в поле операнда команды–приемника

Все права защищены © ОАО «Мультиклет» 2010 — 2015

задают имя команды–источника. Во втором – выполняется поименная рассылка, при которой в команде–источнике задают имя (имена) команды–приемника результата.

Известным примером кодирования программы с использованием прямых информационных связей между командами является язык триад, используемый в качестве промежуточного представления исходной программы в процессе компиляции.

Промежуточное представление в виде триад образуется записями с тремя полями: <op><arg1><arg2>,

где: op – поле кода операции; arg1 – поле первого операнда; arg2 – поле второго операнда.

В качестве операндов используются это либо указатели на таблицу символов (для имен, определенных программистом, или констант), либо указатели на триады–источники используемых результатов. Для следующей последовательности операторов, написанной на языке высокого уровня и образующей линейный участок:

```
L1:     a:=b+c;
        i:=a*f-g;
        a:=i/h;
        goto L2;
```

промежуточное представление на языке триад будет выглядеть так, как показано в таблице 1:

Таблица 1.

Номер триады	op	arg1	arg2
0	+	b	c
1	*	(0)	f
2	-	(1)	g
3	/	(2)	h
4	:=	i	(2)
5	:=	a	(3)
6	goto	L2	

Имена – это указатели на таблицу символов, а числа в скобках – это указатели (ссылки) на триады.

Развитием языка триад является представление программы в контекстно-зависимом виде, при котором она делится на совокупность параграфов. Внутри параграфа информационная связь между его командами только прямая. При этом команды параграфа

Все права защищены © ОАО «Мультиклет» 2010 — 2015

образуют частично упорядоченную и информационно замкнутую линейную последовательность, а именно, команды–источники предшествуют своим командам-приемникам, а команды–приемники имеют ссылки только на команды-источники своего параграфа. Соответственно, информационная связь между командами различных параграфов косвенная, через память.

Ссылки кодируются числом, значение которого равно разнице между номерами команды-приемника и команды-источника, полученными при последовательной нумерации команд параграфа. Так, например, фрагмент программы, приведенный в таблице 1, будет выглядеть следующим образом (см. таблицу 2).

Таблица 2.

Номер триады	op	arg1	arg2
0	rd		b
1	rd		c
2	rd		f
3	rd		g
4	rd		h
5	+	5	4
6	*	1	4
7	-	1	4
8	/	1	4
9	:=	2	4
10	:=	2	i
11	goto	L2	a

Последнюю команду параграфа отмечают управляющим признаком «конец параграфа». Максимальное значение ссылки (окно видимости результата исполненной команды) определяется размером ее поля в команде и равно 2*p–1, где p – размер поля ссылки в битах.

Основные особенности контекстно-зависимой программы, которые определяют организацию вычислительного процесса и аппаратную реализацию ядра мультиклеточного процессора следующие:

Во-первых, последовательность команд имеет линейный характер, а каждая команда имеет свой номер. Следовательно, по номеру команды и по значениям ссылок можно определить номера команд, результаты которых необходимы данной команде. Необходимым условием этого является упорядоченная выборка команд.

Во-вторых, информационные связи между командами указаны явно, что позволяет организовать исполнение команд по готовности операндов и таким образом обеспечить

Все права защищены © ОАО «Мультиклет» 2010 — 2015

естественную реализацию параллелизма. При этом необходимо обеспечить хранение уже выбранной команды до момента ее исполнения и сохранность полученного результата до его передачи командам-приемникам, если они еще не выбраны.

В третьих, естественная реализация параллелизма предполагает использование нескольких процессорных устройств. Так как исполняемая команда не имеет информации о потребителях получаемых результатов, ее результаты должны рассылаться всем потенциальным потребителям, а команды-приемники должны отбирать требуемые им результаты из общего потока. Ядро мультиклеточного процессора, учитывающее эти особенности, состоит из совокупности идентичных процессорных блоков (клеток), объединённых однонаправленной полносвязной коммутационной средой (SB). Каждая клетка включает устройство управления (CU), буфер команд (BUF) и набор исполнительных устройств (EU). Обобщенная структура ядра мультиклеточного процессора приведена на рис.3.

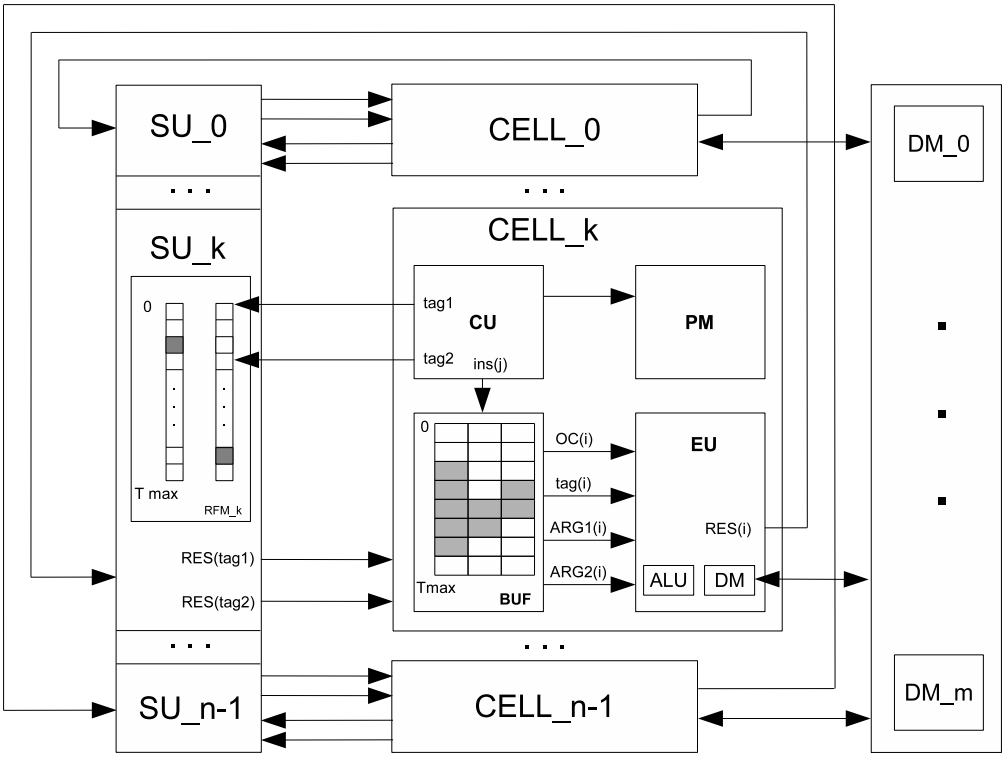

Рис. 3: Обобщенная структура ядра мультиклеточного процессора

Процесс исполнения команды включает три этапа:

1. выборки командных слов;
2. отбор результатов;
3. исполнения команд.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Выборка командных слов осуществляется устройством управления. При этом выполняются следующие действия:

Начиная с первой команды очередного параграфа, устройства управления всех клеток группы мультиклеточного процессора, совместно исполняющей программу (task-группа), синхронно выбирают и декодируют группу последовательно размещенных команд выполняемого параграфа. При этом первая клетка группы выбирает первую команду последовательности, вторая – вторую и т.д. Если среди выбранных команд есть команда с признаком «конец параграфа», то последующие выбранные команды группы считаются невалидными и не декодируются.

При декодировании каждой команде присваивается индивидуальная идентификационная метка (тег), вычисляются теги команд, результаты которых являются операндами декодируемой команды, формируется и передается запрос коммутатору на отбор этих результатов из их общего потока.

Если все клетки группы готовы к приему очередной команды, то декодируемая команда записывается в буфер команд, коммутатор запоминает запрос и выборка повторяется.

Готовность клеток к приему может определяться целым рядом факторов, например, занятостью тега, буфера команд и т.п. Процесс выборки может также приостанавливаться если после выборки последней команды параграфа не известен адрес следующего параграфа. Он возобновляется после его формирования.

Отбор результатов выполняется коммутатором, который состоит из совокупности коммутационных устройств (SU), каждое из которых обслуживает одну клетку. Коммутационное устройство принимает запросы на отбор результатов, запоминает их теги в памяти запросов (RFM) и если среди уже поступивших результатов есть те, значение тегов которых записано в RFM, передает их в буфер команд. В RFM соответствующая отметка убирается (запрос выполнен).

Процесс исполнения команды инициализируется, если в буфере команд есть хотя бы одна команда, у которой получены все операнды. При этом, если готовых к исполнению команд несколько, то выбирается, как правило, команда, которая была записана в буфер раньше всех, и эта команда передается в операционное устройство. Операционное устройство выполняет команду и выдает во все коммутационные устройства клеток своей task-группы её результат с тегом, равным тегу команды. Приостанавливается этот процесс тогда, когда в буфере нет готовых к исполнению команд.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.2 Механизмы реализации

3.2.1 Идентификация команд

Известно, что любая команда выполняется за конечное число тактов и, если в качестве тега использовать некоторое число, то через определенное время это значение будет свободно и оно может быть присвоено очередной выбираемой команде. Очевидно также, что для того чтобы определить значение, которым идентифицируются запрашиваемые результаты необходимо чтобы значения тегов образовывали последовательность и присваивались в порядке размещения команд. А именно, тег очередной команды должен быть на единицу больше тега предыдущей команды. А их общее количество (Tmax) должно быть не менее окна видимости, чтобы обеспечить гарантированное отсутствие блокировки. Таким образом, тег *i-ой* команды при выборке ее *i-ой* клеткой может быть представлен как $tag(i) := (tag(i-k) + k) mod\ Tmax$, где: k – количество клеток в группе. Ограничение ширины окна выбранных команд требует запоминания состояния значения метки (занято, незанято) для того чтобы обеспечить возможность повторного использования значения тега. Механизмом, управляющим процессом назначения тегов и обеспечивающим приостановку процесса выборки при занятости значения служит память занятых тегов (BTM).

Сопоставим каждому значению тега бит, который равен единице, если это значение занято, и нулю, если оно свободно. Бит устанавливается в единицу при записи в буфер команды, которой назначено данное значение тега, и сбрасывается после её выполнения (передачи результата в коммутатор). Регистр, образованный Tmax +1 битами (от 0 до Tmax), каждый из которых соответствует одному значению тега, называется памятью занятых тегов. При этом, занятость конкретного i-го значения тега определяется содержимым BTM(i).

Управляется регистр указателем *ifd_tag*, значение которого присваивается очередной выбранной команде в качестве тега. В исходном состоянии BTM обнулен. Значение указателя *ifd_tag* указывает на начальное значение тега. В нулевой клетке группы оно будет равно нулю, во второй единице, в третьей двойке и т.д. Начиная очередной цикл выборки команд, клетка анализирует содержимое BTM(*ifd_tag*). Если оно равно ‘1’, то выборка в этой и других клетках группы приостанавливается до тех пор, пока оно не станет равным нулю. Если оно равно ‘0’ во всех клетках группы, то выбранной команде присваивается *tag:=ifd_tag*, которое записывается вместе с ней в буфер. После записи команды в буфер значение BTM(*tag*) устанавливается в ‘1’, значение *ifd_tag* устанавливается равным *(ifd_tag+1) mod Tmax* и клетка приступает к выбору следу-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ющей команды.

3.2.2 Организация буфера команд

Буфер команд – это массив из l строк, где

$$m \leq l \leq Tmax + 1$$

m - размер окна видимости и состоит из буфера хранения операционной части команды и буфера хранения операндов.

Операционная часть представляет собой прямоадресуемый массив, строки которого кроме кода команды, включает в себя всю необходимую служебную информацию для исполнения команд и рассылки результатов, а именно, тег команды и признаки готовности первого(второго) операнда для выполнения команды.

Строки буфера хранения операндов позиционно привязаны к строкам операционной части и имеет ассоциативную адресацию. Ассоциативным адресом, по которому осуществляется запись запрошенного результата является его тег, который передается в буфер команд при декодировании команды.

В качестве операндов при выполнении операций могут использоваться: запрошенные результаты команд–источников, поступающие из коммутатора;
значения, вычисленные при декодировании командного слова, а также непосредственно присутствующие в командном слове или взятые из регистров общего назначения.

В первом случае признак готовности данного операнда при записи команды устанавливают в состояние «не готов», а во втором – в состояние «готов». После получения запрошенного результата от коммутационного устройства, признак, в первом случае, также устанавливают в состояние «готов».

Команда, получившая все операнды, проходит приоритетный отбор среди других готовых команд в буферном устройстве, после чего выдается на исполнение при условии незанятости исполнительного устройства.

Каждый полученный результат вместе с тегом его команды посылают во все коммутационные устройства, которые отбирают по номерам, запрошенные данной клеткой результаты, и передают их в ее буферные устройства в качестве операндов.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.2.3 Управление коммутационной средой

Каждое коммутационное устройство состоит из двух блоков памяти запросов и блока памяти результатов. Каждый блок памяти запросов представляет собой регистр размером Tmax+1 бит, в котором при запросе результата с тегом j для использования операнда, в j–том бите регистра записывают единицу («запрос установлен»), а после выдачи результата с этим тегом в j–тый бит записывают ноль («запрос снят»). В первом блоке записываются запросы на получение первого операнда, во втором – второго. Запись запросов и последующее решение по выдаче операндов принимаются независимо, т.е. возможна одновременная выдача операндов в буфер команд с различными тегами. Память результатов состоит из Tmax +1 строк, каждая из которых содержит два поля: значения результата; признака готовности результата.

В исходном состоянии признак готовности результата содержит ноль (результата нет). Каждый полученный клеткой результат с его тегом поступает во все коммутационные устройства группы. Каждое коммутационное устройство записывает полученный результат в поле значения результата строки, номер которой соответствует номеру тега и устанавливает в этой строке признак готовности результата в единицу (результат получен).

Строка коммутатора считается готовой выдать результат, если одновременно есть хотя бы один запрос на этот результат и готово его значение. Из всех строк, готовых к выдаче, выбирается строка с самым "старым"результатом и его значение вместе с тегом, равным номеру строки, выдаётся в буфер команд в качестве операнда. Одновременно сбрасывается признак запроса. Признак готовности и поле значения при этом не изменяются. Таким образом, учитывается, что результат может потребоваться командам, декодируемым позже. Сбрасывается признак готовности результата в i-той строке при назначении тега с этим значением новой команде. Необходимым условием этого является выполнение всех запросов на i-тый результат, т.е. i-тые строки первой и второй памяти запросов должны быть нулевые. Иначе процесс выборки и декодирования задерживается до выполнения всех запросов на i-тый результат.

3.3 Структура и организация клетки

Основой центрального процессорного устройства мультиклеточного микропроцессора является совокупность процессорных блоков (клеток), объединённых особой коммутационной средой и образующей вычислительное ядро. Структурные связи вычислительного ядра, а также структурная схема центрального процессорного устройства мульти-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

клеточного микропроцессора см. на рис.2.

В состав клетки (см на рис.4) входят следующие устройства:
1. Устройство выборки команд IDU (Instruction Distribution Unit).
2. Устройство управления CU (Control Unit).
3. Коммутационное устройство SU (Switch Unit).
4. Буфер команд(BUF).
5. Целочисленное АЛУ.
6. АЛУ с плавающей точкой.
7. Блок доступа к памяти данных DMS (Data Memory Service).
8. Мультиплексор результатов MUX.
9. Набор регистров GPR (General-Purpose Registers).
10. Контроллер прерываний IC (Interrupt Controller).
11. Отладочный блок JTAG-GPR.

Мультиклеточный микропроцессор исполняет программу, представляющую собой последовательность параграфов. Параграф — это набор команд, связанных друг с другом зависимостями по данным. Параграфы связаны друг с другом цепочками условных и безусловных переходов, задающих адрес следующего выполняемого параграфа.

3.3.1 Устройство выборки команд

После сигнала сброса все клетки микропроцессора автоматически конфигурируются в единую task–группу и начинают выборку параграфа с начального адреса. При этом устройства выборки команд клеток группы согласованно читают фрагмент памяти программ, содержащий не менее k очередных команд, которые размещаются на регистрах команд клеток, а именно i – тая команда размещается на регистре команд i – той клетки. Размещение команды сопровождается формированием признака готовности команды к декодированию. Выборка команд конвейеризирована. Одновременно, если среди выбранных k команд не было отмеченных управляющим признаком «конец параграфа», в IDU клеток формируются и записываются новые значения регистров адреса, для согласованного чтения следующего фрагмента памяти программ.

Если среди выбранных команд была команда, отмеченная управляющим признаком «конец параграфа», адрес следующего выполняемого параграфа уже вычислен и были выполнены все записи в регистры, то во всех клетках группы новое значение регистра адреса текущего параграфа устанавливается равным адресу следующего выполняемого параграфа, и выборка команд продолжается с нового адреса. Если среди выбранных ко-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

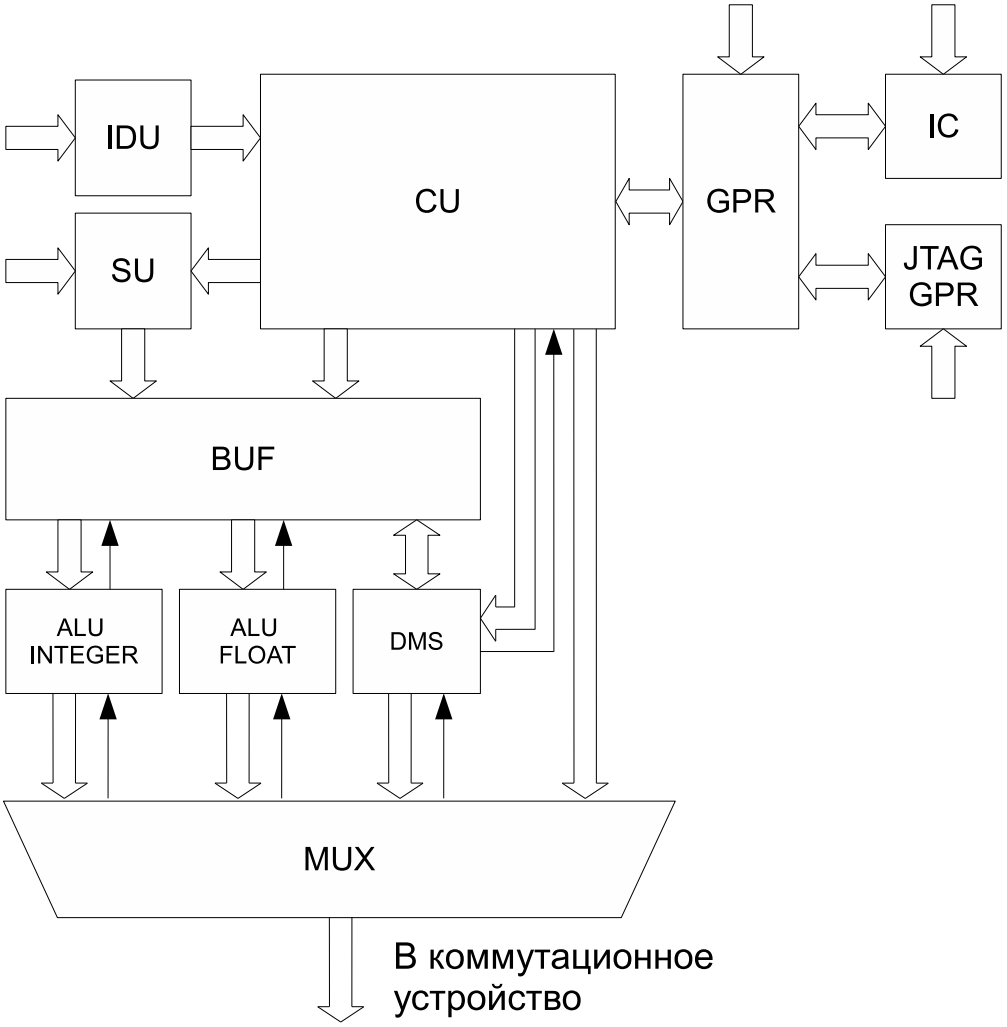

Рис. 4: Структурная схема клетки

манд была команда, отмеченная управляющим признаком «конец параграфа», а адрес следующего выполняемого параграфа на этот момент не вычислен, то выборка команд приостанавливается, и возобновляется, когда будет вычислен адрес следующего выполняемого параграфа.

3.3.2 Устройство управления

Устройство управления обеспечивает декодирование команды, ее запись в буфер команд, либо непосредственно на исполнение, а также формирование (при необходимости) запроса коммутационному устройству на получение результатов ранее выбранных команд.

В каждой клетке, при декодировании, каждой очередной выбранной команде присваивается очередное значение тега. Значение очередного тега формируется клетками син-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

хронно и изменяется циклически от 0 до Tmax.

Присвоенное значение тега сопровождает команду на всех этапах её выполнения и присваивается результату её выполнения. Если значение тега присвоено команде и результат её выполнения не получен коммутационным устройством, либо получен, но пока не выдан потребителям, то это значение тега считается занятым.

По значению тега команды и значениям её ссылок формируются теги запрашиваемых результатов для формирования первого и второго операндов команды-приёмника. Эти теги используются для формирования запроса коммутационному устройству на результаты и в качестве ассоциативного адреса при записи их значения в требуемую строку буфера команд. В последнем случае они записываются в буфер, вместе с кодом операции и ее тегом, как значения операндов, сопровождаемые нулевым признаком готовности операнда («не готов»). Если результаты не запрашиваются, а значения операндов берутся из регистров или непосредственно из командного слова, то их значения записываются в буфер команд с признаком готовности операнда равным единице («готов»).

Одновременно в клетке формируется и передаётся в устройство выборки команд сигнал готовности клетки к приёму следующей команды. Формирование сигнала приостанавливается, если хотя бы в одной клетке группы:
1. Значение тега занято;
2. Установлен режим контроля очередности чтения/записи, а декодируемая команда – команда чтения и в буфере есть не выполненные команды чтения предыдущего параграфа;
3. Установлен режим контроля очередности чтения/записи, а декодируемая команда – команда записи и в буфере есть не выполненные команды записи предыдущего параграфа;
4. В буфере команд есть свободное место.

В ряде случаев устройство управления может самостоятельно выполнить команду без помещения ее в буфер в команд:
1. Если выбранная команда — чтение регистра.
2. Если для выбранной команды требуется только константа и/или значение регистра (команды SET, GET, JMP или IRM).
3. Если выбранная команда — чтение из памяти по известному адресу, содержащемуся в константе и/или регистре, и блок DMS не занят.

Также устройство управления может самостоятельно выполнять обращение к блоку

Все права защищены © ОАО «Мультиклет» 2010 — 2015

DMS, если декодированная инструкция использует косвенную адресацию аргумента, и блок DMS не занят.

Если текущая конфигурация микропроцессора предусматривает несколько task-групп, то каждая task-группа работает независимо. При этом каждая task-группа имеет независимый набор регистров общего назначения и индексных регистров, а клетки внутри группы работают так же, как было описано выше.

Изменение текущей конфигурации процессора возможно только после выполнения всех команд текущего параграфа. Для задания конфигурации, отличных от конфигурации по умолчанию, сначала устанавливается в «1» бит 7 (бит разрешения реконфигурации) и бит 8 (бит признака очистки буферов) регистра PSW. Далее мультиклеточный микропроцессор будет функционировать следующим образом:
1. Текущий параграф («параграф 1») продолжит выполняться со «старым» значением PSW.
2. Фактическое значение PSW обновится (т.е. установятся биты 7 и 8) только при переходе на следующий параграф («параграф 2»).
3. Внутри параграфа 2 программа должна установить новые значения регистров конфигурации процессора (ICR, PSW и NEWADDR).
4. Поскольку установлен бит 8 регистра PSW, выполнение параграфа 2 не завершится, пока не не будут выполнены все команды, находящиеся в буферах.
5. По завершении параграфа 2 регистры конфигурации принимают значения, заданные программой, и формируется новый набор task-групп.
6. Каждая task-группа, работая независимо, начинает выполнение программы со «своего» адреса, заданного ей через регистр NEWADDR.

При изменении конфигурации клеток процессора они могут объединяться в одну группу (композиция), либо разбиваться по группам. При композиции клеток одна из групп является «ведущей», т.е. исполняемая ей программа «присоединяет» клетки к данной группе. При этом программы, исполняемые task-группами, должны быть построены таким образом, чтобы в момент композиции все присоединяемые task-группы были остановлены (например, с помощью параграфа, не содержащего инструкций перехода).

3.3.3 Буфер команд

Декодированная команда записывается в строку буфера кода операции буферного устройства, для этой строки формируется признак «занято». В буфер соответствующего операнда буферного устройства, если значение операнда известно, записывают его зна-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

чение, устанавливают в состояние «да» признак готовности операнда, иначе в массив тегов операнда записывается тег результата выполнения команды, который был запрошен в качестве операнда, признак готовности операнда устанавливают в состояние «нет». Если из коммутационного устройства (для первого операнда) или из коммутационного устройства или блока DMS (для второго операнда) в качестве операнда приходит результат выполнения команды с запрошенным тегом (этот тег совпадает с тегом, записанным в массив тегов операнда), то этот результат записывается в буфер операнда буферного устройства в строку, номер которой совпадает с номером строки буфера кода операции буферного устройства, в которую был записан код операции соответствующей команды. Команда передается на исполнение если:

1. Известны значения всех необходимых операндов.
2. Соответствующее исполнительное устройство готово к приёму команды.
3. В буферном устройстве нет более приоритетных (более «старых») команд, готовых к выполнению.

В буферном устройстве формируется и выдаётся устройству управления сигнал готовности буферного устройства к записи декодируемой команды, имеющий два состояния «да» и «нет», указывающие на его готовность, анализируются коды операций команд, находящихся в буферном устройстве, и формируют сигналы наличия команд чтения и записи, которые имеют два состояния «да» и «нет», указывающие на наличие или отсутствие соответствующих команд в буферном устройстве. Эти сигналы выдаются в мультиклеточный процессор, где одноименные сигналы, полученные от всех клеток объединяют схемами ИЛИ и формируют, соответственно, общие сигналы наличия команд чтения и записи. Очерёдность выполнения команд внутри параграфа подчиняется следующим правилам:

1. Если включён контроль очерёдности чтения/записи, выполнение команд записи не начнётся, пока не завершится выполнение всех команд чтения в текущем параграфе, а выполнение команд чтения не начнется, пока не будут выполнены все команды записи предыдущих параграфов.
2. При выполнении команд записи в регистры общего назначения, индексные и служебные регистры, записываемые значения помещаются в регистры только по окончании декодирования всех команд текущего параграфа, и перед выборкой команд следующего параграфа. Следует заметить, что адрес следующего параграфа также хранится в одном из служебных регистров (NEWADDR), с аналогичной логикой работы.
3. При выполнении команд записи в периферийные регистры, записываемые значения помещаются в регистры непосредственно в момент выполнения команды.
4. Во всех других случаях очерёдность выполнения команд определяется исключитель-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

но готовностью значений аргументов команд, готовностью исполнительных устройств клетки и наличием других («более старых») готовых команд.

3.3.4 Коммутационное устройство

Состав и функционирование коммутационного устройства изложены в разделах выше.

3.3.5 Целочисленное АЛУ

Целочисленное АЛУ предназначено для выполнения команд над целочисленными операндами, при этом операнды могут быть представлены в следующих форматах (в зависимости от команды):
64-битные;
32-битные;
16-битные;
8-битные;
упакованные 32-битные;
упакованные 16-битные.

Структурно целочисленное АЛУ состоит из двух частей:
умножитель – конвейеризированный блок, предназначенный для выполнения команды умножения;
вычислитель – неконвейеризированный блок, предназначенный для выполнения всех остальных команд из набора команд над целочисленными операндами.

3.3.6 АЛУ с плавающей точкой

АЛУ с плавающей точкой предназначено для выполнения команд над операндами, представленными в формате чисел с плавающей запятой одинарной (32 бита) или двойной точности (64 бита) в соответствии с требованиями стандарта IEEE-754. Структурно АЛУ с плавающей точкой состоит из трёх частей:
1. делитель одинарной точности – блок, предназначенный для выполнения команд деления и извлечения квадратного корня над операндами, представленными в формате чисел с плавающей точкой одинарной точности;

Все права защищены © ОАО «Мультиклет» 2010 — 2015

2. делитель двойной точности – блок, предназначенный для выполнения команд деления и извлечения квадратного корня над операндами, представленными в формате чисел с плавающей точкой двойной точности;
3. вычислитель FASM – блок, предназначенный для выполнения остальных команд из набора команд над операндами, представленными в формате чисел с плавающей точкой одинарной или двойной точности.

АЛУ с плавающей точкой аппаратно оптимизирован для выполнения команд комплексного умножения операндов, представленных в формате чисел с плавающей точкой одинарной точности. Оптимизация достигается за счёт одновременного выполнения операций умножения и сложения, входящих в состав команды комплексного умножения.

Делители одинарной и двойной точности — неконвейеризированные, поэтому, АЛУ с плавающей точкой не может принять на выполнение следующую команду деления или извлечения квадратного корня, пока не будет выполнена предыдущая.

Вычислитель FASM – конвейеризированный, поэтому, может принимать на выполнение следующую команду на каждом следующем такте.

3.3.7 Блок доступа к памяти данных

3.3.8 Мультиплексор результатов

Обеспечивает приоритетную выдачу результатов в коммутационные устройства, если одновременно готовы результаты выполнения команд целочисленного АЛУ, АЛУ с плавающей точкой, DMS, либо результат получен непосредственно из устройства управления. По приоритету выдачи результата исполнительные устройства распределяются следующим образом:
1. Устройство управления;
2. АЛУ с плавающей точкой;
3. DMS;
4. Целочисленное АЛУ.

Исполнительные устройства, результат работы которых на текущем такте не может быть выдан в коммутационное устройство, записывают результат выполнения команды в выходной регистр и ожидают своей очереди на выдачу результата.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.3.9 Набор регистров

В состав каждой клетки входит свой набор регистров (управляющие (служебные), индексные и регистры общего назначения). Наборы регистров клеток, входящих в одну task-группу, объединяются в единую группу, при этом чтение и запись в одинаковые регистры разных клеток происходят идентично. Со стороны центрального процессорного устройства это выглядит, как наличие у task-группы одного набора регистров (как у одной клетки). При изменении конфигурации соответственно происходит перегруппировка наборов регистров клеток.

В состав каждой клетки входит свой блок регистров (GPR), включающий:
- набор регистров общего назначения,
- набор индексных регистров,
- регистр управления микропроцессора PSW,
- регистр адреса возврата из прерываний IRETADDR,
- регистр принудительного запуска прерываний FORCE,
- регистр адреса обработчика прерываний IHOOKADDR,
- регистр модификации индексных регистров REGMOD,
- регистр адреса текущего параграфа CURRENTADDR,
- регистр адреса следующего параграфа NEWADDR,
- регистр предыдущего значения PSW LASTPSW,
- регистр конфигурации ICR.

Кроме указанных выше регистров, блок GPR имеет в своём составе интерфейс системных регистров мультиклеточного процессора. В следующих разделах имеется более подробное описание регистров.

3.3.10 Контроллер прерываний и отладочный блок

Подробно описываются в следующих разделах.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.4 Регистры

Все регистры, кроме системных, имеют ширину 64 бита. Чтение/запись регистров происходит с помощью специализированных команд.

РОН, индексные регистры и управляющие регистры имеют копии в каждой клетке. Эта особенность является необходимостью, т.к. при реконфигурации необходимо иметь набор регистров в каждой клетке. Системные регистры находятся в устройствах, таких как системный таймер, PLL и т.д. Это разделяемые ресурсы МП.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Название	Примечание
РОН, индексные регистры (дублируются в каждой клетке)		
0-7	нет	регистры общего назначения (РОН)
8-15	нет	индексные регистры
Системные регистры (существуют в одном экземпляре в системных устройствах)		
16	PLLCR	регистр управления генератором
17	PLLSTR	регистр состояния генератора
18	PLLTMCR	регистр управления модулем коммутации системы тактирования
19	SWCH0_CTRL	управления каналом памяти 0
20	SWCH1_CTRL	управления каналом памяти 1
21	SWCH2_CTRL	управления каналом памяти 2
22	SWCH3_CTRL	управления каналом памяти 3
23	ST0PRDR	период системного таймера 0
24	ST1PRDR	период системного таймера 1
25	ST2PRDR	период системного таймера 2
26	ST3PRDR	период системного таймера 3
27	ST0CR	регистр управления таймера 0
28	ST1CR	регистр управления таймера 1
29	ST2CR	регистр управления таймера 2
30	ST3CR	регистр управления таймера 3
31	*зарезервировано*	
32	ST0VAL	текущее значение таймера 0
33	ST1VAL	текущее значение таймера 1
34	ST2VAL	текущее значение таймера 2
35	ST3VAL	текущее значение таймера 3
36-38	*зарезервировано*	
39	DTC_STEP_INDX	регистр управления инкрементацией адресов DTC
40	DTC_CTRL	регистр управления DTC
41	DTC_ST	регистр состояния DTC
42	DTC_IMASK0	регистр маски 0 DTC
43	DTC_IMASK1	регистр маски 1 DTC
44	DTC_DATA	регистр данных DTC
45	DTC_S_ADDR	адрес источника
46	DTC_D_ADDR	адрес приёмника
47	ROM_CTRL	регистр управления и статуса внешнего ПЗУ
Управляющие регистры (дублируются в каждой клетке)		
48	PSW	регистр управления процессором
49	INTR	регистр прерываний
50	MSKR	регистр маски прерываний
51	ER	регистр ошибок
52	IRETADDR	регистр адреса возврата из прерываний
53	FORCE	регистр принудительного запуска прерываний
54	*зарезервировано*	
55	IHOOKADDR	адрес обработчика прерываний
56	INTNUM	регистр номера прерывания
57	REGMOD	регистр модификации индексных регистров
58	*зарезервировано*	
59	CURRENTADDR	регистр адреса текущего параграфа
60	NEWADDR	регистр адреса следующего параграфа
61	LASTPSW	регистр предыдущего значения PSW
62	DFADDR	регистр аварийного перехода
63	ICR	регистр информационной связанности

Таблица 4: Регистры процессора

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.5 Контроллер прерываний

Контроллер прерываний входит в состав каждой клетки. В МП контроллеры прерывания работают как один в выделенной группе клеток. В группе клеток контроллеры прерываний не могут быть разделены или сконфигурированы отдельно. Система прерываний МП допускает обработку 44 прерываний. Источник с номером «0» имеет наивысший приоритет при обработке прерываний.

Структура прерываний

Прерывания можно условно разделить на две группы: системные прерывания и прерывания от периферии. Системные прерывания занимают адреса с 0 по 15 в контроллере прерываний и являются более приоритетными по сравнению с прерывания от периферии. Системные прерывания с номерами с 0 по 4 являются немаскируемыми. Прерывания от периферии занимают адреса с 16 по 63.

0	Немаскируемое внешнее прерывание (ENMI)
1	*зарезервировано*
2	Немаскируемое исключение в аппаратной части (PERE)
3	Немаскируемое программное исключение (PRGE)
4	*зарезервировано*
5	*зарезервировано*
6	Маскируемое прерывание - внешний сигнал wake_up
7	Маскируемое прерывание от контроллера транзакций
8	Маскируемое прерывание от системного таймера (ST0)
9	Маскируемое прерывание от системного таймера (ST1)
10	Маскируемое прерывание от системного таймера (ST2)
11	Маскируемое прерывание от системного таймера (ST3)
12	Маскируемое программное прерывание (SW0)
13	Маскируемое программное прерывание (SW1)
14	Маскируемое программное прерывание (SW2)
15	Маскируемое программное прерывание (SW3)
16	Маскируемое прерывание от ADC0
17	Маскируемое прерывание от DAC0
18	Маскируемое прерывание от ADC1
19	Маскируемое прерывание от DAC1
20	Маскируемое прерывание от UART0
21	Маскируемое прерывание от UART1
22	Маскируемое прерывание от UART2
23	Маскируемое прерывание от UART3
24	Маскируемое прерывание от I2C0
25	Маскируемое прерывание от I2C1
26	Маскируемое прерывание от SPI0
27	Маскируемое прерывание от SPI1
28	Маскируемое прерывание от I2S0
29	Маскируемое прерывание от GPTIM0

Все права защищены © ОАО «Мультиклет» 2010 — 2015

30	Маскируемое прерывание от GPTIM1
31	Маскируемое прерывание от GPTIM2
32	Маскируемое прерывание от GPTIM3
33	Маскируемое прерывание от WDT
34	Маскируемое прерывание от RTC
35	Маскируемое прерывание от GPIOA
36	Маскируемое прерывание от GPIOB
37	Маскируемое прерывание от GPIOC
38	Маскируемое прерывание от GPIOD
39	Маскируемое прерывание от GPIOE
40	Маскируемое прерывание от GPIOF
41	Маскируемое прерывание от ETHERNET0
42	Маскируемое прерывание от USB0
43	Маскируемое прерывание от USB0_EPI
44	Маскируемое прерывание от USB0_EPO
45	Маскируемое прерывание от PWM0
46	Маскируемое прерывание от STAT
...	
63	

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.5.1 Структура контроллера прерываний

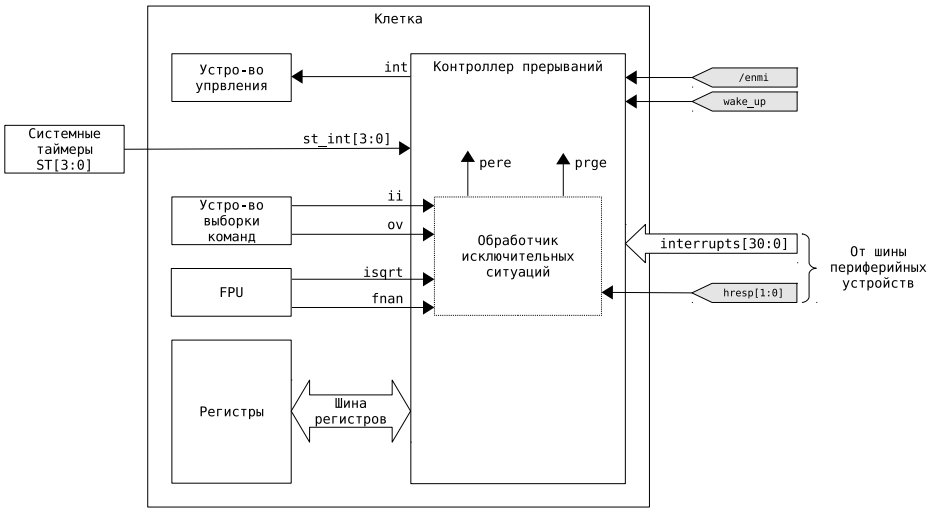

Рис. 5: Блок-схема контроллера прерываний

Контроллер прерываний выполняет следующие функции:

- определяет самое приоритетное прерывание на каждом такте и формирует его номер на шине INTNUM;
- обрабатывает информацию об аппаратных и программных сбоях, формирует запрос обработки прерывания, если возникло что-то из перечисленного.

Из клетки на контроллер прерываний приходят 4 сигнала об ошибках в ходе выполнения программы, по которым в обработчике исключений формируется сигнал PRGE:

- II – выбрана несуществующая инструкция;
- OV – возникло переполнение при попытке разделить на 0 в блоке операций с числами с плавающей запятой;
- ISQRT – попытка взять корень из отрицательного числа в блоке операций с числами с плавающей запятой;
- FNAN – в ходе операции с числами с плавающей запятой или на входе блока операций с числами с плавающей запятой возникло число = NAN.

С шины периферийных устройств на обработчик исключений поступают сигналы HRESP, состояние линий которых говорит об ошибках или их отсутствии при обращении по шине. При наличии ошибки формируется сигнал PERE.

Так же каждая клетка формирует сигнал int_st - прерывание от системного таймера. и SWI - программно-формируемое прерывание. С шины периферийных устройств при-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ходят сигналы запроса прерывания от каждого периферийного устройства. С вводов/-выводов МП приходят сигналы /enmi – внешнее немаскируемое прерывание и wake_up - сигнал вывода МП из «спящего» режима внешней системой.

3.5.2 Регистры контроллера прерываний

Для работы системы прерывания и функционирования программ обработки прерываний имеются следующие регистры:

Регистр	**Доступ**	**Описание**
INTR	*RW*	Регистр прерываний
MSKR	*RW*	Регистр маски прерываний
FORCE	*W*	Регистр установки прерывания
INTNUM	*R*	Регистр номера выработанного прерывания
ER	*R*	Регистр ошибок

Все регистры имеют разрядность 64 бита. Номера бит во всех регистрах кроме INTNUM и ER, которые имеет другую структуру, соответствуют номерам прерываний.

Регистр INTR.

Доступен как для чтения, так и для записи. Запись 1 в какие-либо биты данного регистр произведет сброс соответствующих бит данного регистра в 0. Данный регистр является регистром защелкой, и он захватывает пришедшие на его вход данные до тех пор, пока не будет сброшен записью 1 в соответствующие биты.

INTR	**Регистр прерываний**	
Номер бита	63 . . . 47	46 . . . 0
Описание	*зарезервировано*	INTR

63 . . . 47 *зарезервировано*
46 . . . 0 номер выработанного прерывания

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Регистр MSKR.

Регистр MSKR доступен как для чтения, так и для записи. Для разрешения прерываний с определенными номерами, необходимо выставить в 1 соответствующие биты данного регистра. Биты 0-3 являются немаскируемыми.

MSKR	**Регистр маски прерываний**		
Номер бита	63 . . . 47	46 . . . 4	3 . . . 0
Описание	*зарезервировано*	MSKR	*зарезервировано*

63 . . . 47	*зарезервировано*
46 . . . 4	номер маскируемого прерывания
3 . . . 0	*зарезервировано*

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Регистр FORCE.

Регистр FORCE доступен только для записи. Запись 1 в какие-либо биты данного регистра сформируют единичный импульс на вход регистра прерываний, что в свою очередь вызовет установку соответствующих бит в регистре INTR. На следующем такте после записи 1 в какие-либо биты данного регистра, все биты данного регистра будут сброшены в 0.

FORCE	Регистр установки прерываний	
Номер бита	63 . . . 47	46 . . . 0
Описание	*зарезервировано*	FORCE

63 . . . 47 *зарезервировано*
46 . . . 0 номер принудительного прерывания

Регистр INTNUM.

Регистр INTNUM доступен только для чтения, данный регистр содержит номер самого приоритетного прерывания, среди тех, что ждут обработки в данный момент.

INTNUM	Регистр управления каналом коммутатора	
Номер бита	63 . . . 6	5 . . . 0
Описание	*зарезервировано*	INTNUM

63 . . . 6 *зарезервировано*
5 . . . 0 номер самого приоритетного прерывания на момент чтения регистра

Регистр ER.

Регистр ER доступен только для чтения, данный регистр содержит информацию о программных или аппаратных сбоях. Во всех битах регистра ’1’ является признаком возникновения события.

ER	Регистр управления каналом коммутатора						
Номер бита	63 . . . 7	6 . . . 5	4	3	2	1	0
Описание	—	HRESP	II	OV	—	ISQRT	FNAN

63 . . . 7	—	*зарезервировано*
6 . . . 5	HRESP	ошибка при обращении по шине периферийных устройств (значения отличные от нуля являются признаком наличия ошибки)
4	II	выбрана несуществующая инструкция
3	OV	возникло переполнение при попытке разделить на 0 в блоке операций с числами с плавающей запятой
2	—	*зарезервировано*
1	ISQRT	попытка взять корень из отрицательного числа в блоке операций с числами с плавающей запятой
0	FNAN	в ходе операции с числами с плавающей запятой или на входе блока операций с числами с плавающей запятой возникло число = NAN

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.5.3 Порядок обработки прерываний

В случае возникновения события, вызывающего запрос на обработку прерывания, оно фиксируется в регистре INTR. Регистр MSKR является фильтром, который допускает или нет сигналы из регистра INTR к дальнейшей обработке. После регистра MSKR сигналы попадают в схему определения приоритета, где происходит определение самого приоритетного прерывания в данном такте, его номер заносится в регистр INTNUM. На выходе контроллера прерывания сигнал int переходит в активное состояние. Клетка фиксирует активный уровень сигнала запроса обработки прерывания. Переход в обработчик прерывания происходит в конце параграфа, когда завершаются все записи в память и регистры, если таковые были в параграфе.

Адрес начального обработчика прерываний находится в регистре IHOOKADDR. Ядро работает только с одним адресом перехода на обработчик прерываний. По этому адресу может находится любой алгоритм на усмотрение пользователя. Определение порядка работы с конкретным запросом прерывания возложено на программиста.

Латентность времени перехода на обработчик прерывания имеет зависимость от программы пользователя. Рекомендуется перед стартом программы очистить регистр INTR записью 1 во все биты регистра.

Важно заметить, что для каждой группы клеток можно задавать обработчик прерываний в регистре IHOOKADDR, задавать MSKR, т.к. у каждой группы клеток появляется свой набор этих регистров.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3.6 Системный таймер

Системный таймер предназначен для формирования заданных периодических или однократных временных интервалов. Таймер представляет собой инкрементирующий счетчик с делителем тактового сигнала на входе. Конечное значение счетчика записывается в регистр периода таймера, управление осуществляется через регистр STxCR, счет начинается с нуля. По истечении заданного временного интервала формируется запрос на обработку прерывания.

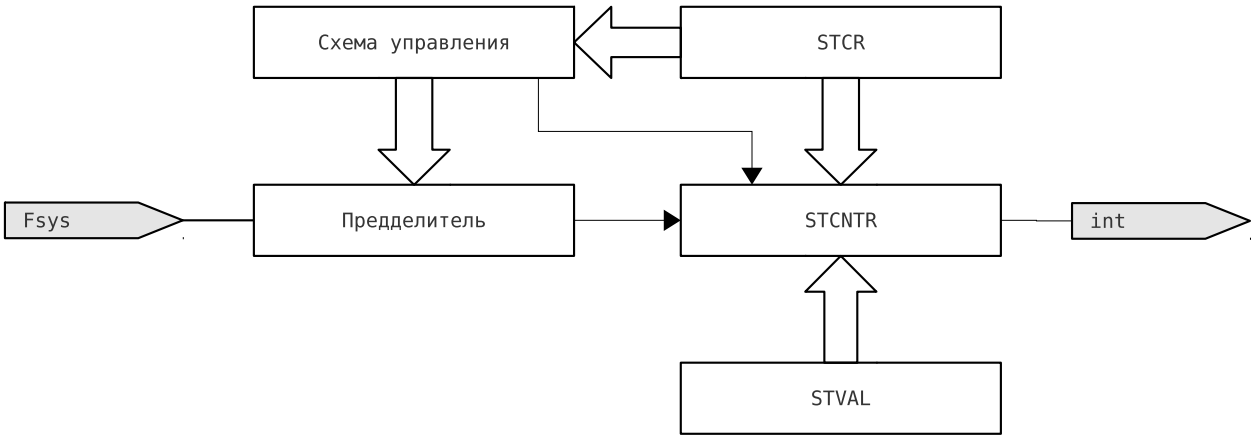

Рис. 6: Блок-схема системного таймера

На рис.6 изображена блок-схема таймера. Ниже приведены формулы для расчета частоты и периода интервалов, формируемых системным таймером. Значения PREDIV и CNTVAL задаются в регистрах STxCR и STxPRDR соответственно.

- период формируемого интервала:

$$T = T_{clk} \cdot (PREDIV + 1) \cdot (CNTVAL + 1);$$

- частота следования временных интервалов:

$$F = \frac{F_{clk}}{(PREDIV + 1) \cdot (CNTVAL + 1)};$$

3.6.1 Режимы работы

- Однократное формирование временного интервала – таймер запускается пользователем (в бит STxCR(EN)='1'), а по достижении счетчиком таймера значения, заданного в регистре периода таймера, таймером выдается запрос на обработку

Все права защищены © ОАО «Мультиклет» 2010 — 2015

прерывания, после этого бит STxCR(EN) устанавливается в ’0’, и таймер останавливается до следующей записи значения ’1’ в STxCR(EN);

- Периодическая генерация временных интервалов – таймер запускается и останавливается пользователем (в бит STxCR(EN) записывается соответствующее значение). По достижении счетчиком таймера значения, заданного в регистре периода таймера, таймером выдается запрос обработки прерывания, счетчик перезагружается значением ’0’, и работа таймера продолжается до того, как пользователь запишет значение ’0’в бит STxCR(EN).

При записи в регистр STxPRDR новое значение периода будет установлено и счетчик начнет считать с нуля. При записи в STxCR(PREDIV) новое значение предделителя будет загружено и счетчик начнет считать с нуля. Настоятельно рекомендуется перед изменением режима работы таймера сначала остановить его, записав STxCR(EN)=’0’, потом задать новые значения в регистре STxCR. Режим работы системного таймера устанавливается в STxCR(MD). Доступно 4 системных таймера: ST0, ST1, ST2, ST3.

STxPRDR	**Период системного таймера**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—																															
Описание	PERIOD																															

0-31 PERIOD Период системного таймера

STxCR	**Регистр управления системного таймера**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																								—						0	0
Описание	PREDIV																								—						MD	EN

0 EN Разрещение работы системного таймера (’0’– запрещено, ’1’– разрешено)
1 MD Выбор режима работы таймера (’0’– однократный, ’1’– многократный)
2-7 — *зарезервировано*
8-31 PREDIV Предделитель системного таймера

STxVAL	**Текущее значение системного таймера**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	VALUE																															

0-31 VALUE Текущее значение системного таймера

Все права защищены © ОАО «Мультиклет» 2010 — 2015

4 Средства динамической реконфигурации

Код для мультиклеточного ядра не зависит от количества клеток, которыми он будет исполняться, вычислительные ресурсы мультиклеточного ядра можно распределять во время работы, управление происходит программно. Способность мультиклеточной архитектуры перераспределять свои ресурсы называется — **динамической реконфигурацией**. Например, клетки мультиклеточного ядра могут быть, как угодно распределены для выполнения какого-либо алгоритма или его части. **Группа** - это часть клеток мультиклеточного ядра, которые связаны между собой для выполнения какого-либо или части алгоритма независимо от других клеток. В группе может находиться 1 и более клеток. Когда мультиклеточное ядро разделяет группу на части - это называется **декомпозиция**. Процесс объединения в группу - **композиция**. На рис. 7 показано как 4-х клеточное ядро со временем реконфигурируется (пример). Клетки, выполняющие одну задачу, имеют одно цветовое заполнение.

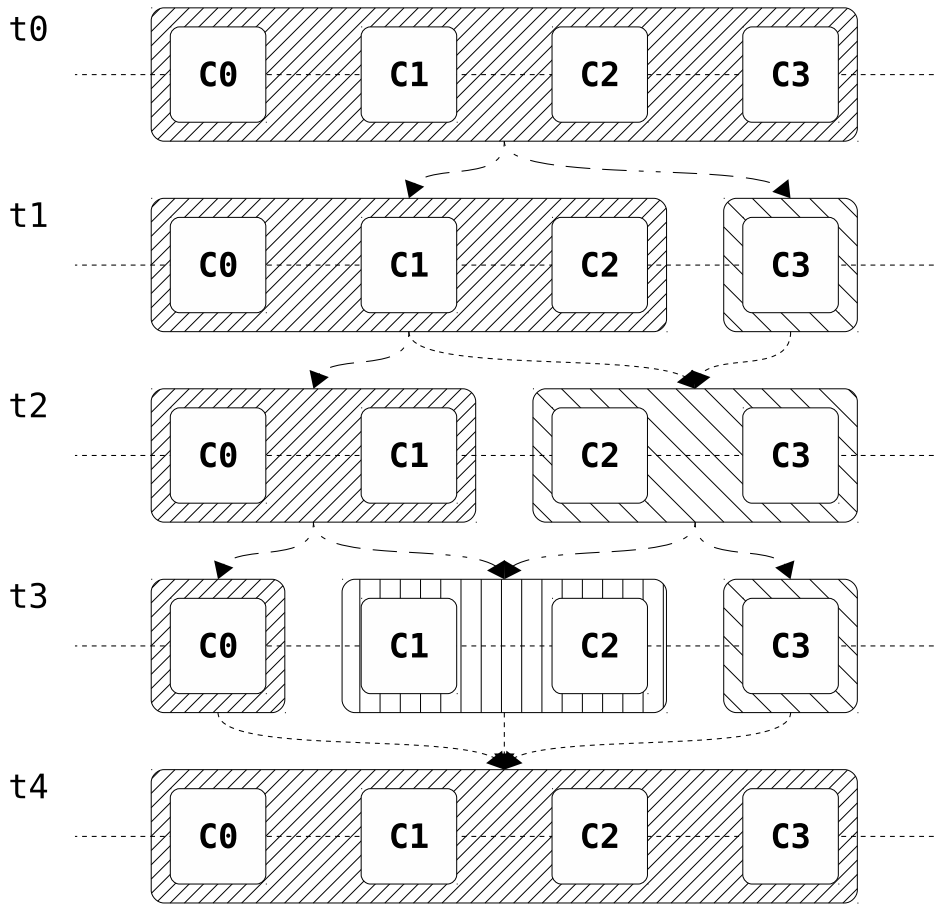

Рис. 7: Распределение вычислительных ресурсов

Архитектура мультиклеточного процессора не налагает каких-либо ограничений на ко-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

личество групп и количество клеток в группе. Все клетки могут быть объединены в одну группу. Может быть сформировано n групп, в каждой будет по одной клетке. Конкретное распределение клеток по группам определяется программистом и может неоднократно программно меняться в процессе вычислений.

Для управления динамической реконфигурацией используются: регистр перехода (NEWADDR) и регистр информационной связанности (ICR).

NEWADDR имеет размер 64 бита и определяет адрес следующего выполняемого параграфа для новой группы клеток. Значение NEWADDR формируется программно и может быть использовано только в текущем параграфе. При переходе к следующему параграфу оно не сохраняется.

NEWADDR	**Регистр перехода**					
Номер бита	63 . . . 36	35	34	33	32	31 . . . 0
Описание	*зарезервировано*	C0	C1	C2	C3	NEWADDR

63 . . . 36	—	*зарезервировано*
35	C0	клетка для которой предназначен новый адрес (’1’ - адрес предназначен)
34	C1	клетка для которой предназначен новый адрес (’1’ - адрес предназначен)
33	C2	клетка для которой предназначен новый адрес (’1’ - адрес предназначен)
32	C3	клетка для которой предназначен новый адрес (’1’ - адрес предназначен)
31 . . . 0	NEWADDR	адрес перехода

ICR имеет размер n бит, где n – количество клеток. Данный регистр определяет подмножество клеток образующих группу.

ICR	**Регистр информационной связанности**									
Номер бита	63 . . . 36	35	34	33	32	31 . . . 4	3	2	1	0
Описание	*зарезервировано*	C0_G	C1_G	C2_G	C3_G	*зарезервировано*	C0_EN	C1_EN	C2_EN	C3_EN

63 . . . 36	—	*зарезервировано*
35	C0_G	клетка войдет в состав группы (’1’ - войдет в состав)
34	C1_G	клетка войдет в состав группы (’1’ - войдет в состав)
33	C2_G	клетка войдет в состав группы (’1’ - войдет в состав)
32	C3_G	клетка войдет в состав группы (’1’ - войдет в состав)
31 . . . 4	—	*зарезервировано*
3	C0_EN	разрешение работы клетки в составе группы(’1’ - разрешено)
2	C1_EN	разрешение работы клетки в составе группы(’1’ - разрешено)
1	C2_EN	разрешение работы клетки в составе группы(’1’ - разрешено)
0	C3_EN	разрешение работы клетки в составе группы(’1’ - разрешено)

Если i и j клетки после динамической реконфигурации будут входить в одну группу, то $ICR(Ci\_G) = ICR(Cj\_G) =' 1'$, где x номер условной группы. Если они в разных группах, то $ICR(Ci\_G) = ICR(Cj\_G) =' 0'$. ICR позволяет каждой клетке выбрать из всей совокупности управляющих сигналов и результатов команд мультиклеточного процессора только те, которые относятся к ее группе. В исходном состоянии все клетки образуют одну группу и, следовательно, ICR во всех битах содержит ’1’. Новое содержимое ICR формируется только программно при динамической реконфигурации мультиклеточного процессора. При этом, для обеспечения корректной работы необходимо,

Все права защищены © ОАО «Мультиклет» 2010 — 2015

чтобы каждая клетка всегда входила только в одну группу.

Важно заметить, что если для клетки из состава группы поставить значение разрешение работы равное ’0’ то в этом случае клетка будет отключена до перезагрузки процессора.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

5 Система команд (Instruction Set Architecture)

5.1 Формат кодирования команды

5.1.1 Формат AA, размер 32 бита

Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	Cop						Sz	S1	S0	C	Top				*зарезервировано*						F1						F2					

31...26	Cop	Код операции
25	Sz	Признак наличия поля V у команды ('0' — нет, '1' — есть), и, как следствие, размера команды
24	S1	Интерпретация значения поля F2 ('0' — адрес в коммутаторе, '1' — номер регистра)
23	S0	Признак чтения из памяти данных ('0' — значение используется непосредственно, '1' — значение используется в качестве адреса памяти данных)
22	C	Признак инструкции, завершающей параграф
21...18	Top	Тип операции
17...12	—	*зарезервировано*
11...6	F1	Адрес в коммутаторе значения первого операнда
5...0	F2	Адрес в коммутаторе / номер регистра второго операнда

5.1.2 Формат AV, размер 64 бита

Старшая часть имеет такую же структуру, как и формата AA, младшие 32 бита — значение поля `V`.

5.2 Форматы адресации

Биты `S1` и `S0` образуют двухбитовый суффикс (suffix) команды. Суффикс, совместно с битом `Sz`, определяет режим адресации второго аргумента команды:

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Запись	Sz	Suffix		Интерпретация
		S1	S0	
@N, @M	0	0	0	$arg_1 = @N$ $arg_2 = @M$
@N, #R	0	1	0	$arg_1 = @N$ $arg_2 = \#R$
@N, @M + V	1	0	0	$arg_1 = @N$ $arg_2 = @M + V$
@N, #R + V	1	1	0	$arg_1 = @N$ $arg_2 = \#R + V$
@N, [@M]	0	0	1	$arg_1 = @N$ $arg_2 = [@M]$
@N, [#R]	0	1	1	$arg_1 = @N$ $arg_2 = [\#R]$
@N, [@M + V]	1	0	1	$arg_1 = @N$ $arg_2 = [@M + V]$
@N, [#R + V]	1	1	1	$arg_1 = @N$ $arg_2 = [\#R + V]$

Замечания:

- $@M$ — ссылка на результат команды, отстоящей от текущей на M команд назад,
- $\#R$ — значение регистра номер R,
- $[x]$ — значение, находящееся в памяти по адресу x (тип чтения из памяти соответствует типу родительской операции),
- все операции сложения при адресации второго аргумента — 32-разрядные целочисленные беззнаковые),
- $@0 \equiv 0$

5.3 Типы операции

Тип операции задается кодом в поле `Top`.

0000	Беззнаковый байт	(Byte)
0001	Беззнаковый short	(Short)
0010	Беззнаковый long	(Long)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

0011	Беззнаковый quad	(Quad)
0100	Упакованный беззнаковый short	(Pack short)
0101	Упакованный знаковый short	(Pack signed short)
0110	Упакованный беззнаковый long	(Pack long)
0111	Упакованный знаковый long	(Pack signed long)
1000	Знаковый байт	(Signed byte)
1001	Знаковый short	(Signed short)
1010	Знаковый long	(Signed long)
1011	Знаковый quad	(Signed quad) — не используется
1100	Float	
1101	Double	
1110	Pack	
1111	Complex	

5.3.1 Типы в ассемблере

Для резервирования памяти данных в ассемблере используются директивы резервирования памяти, используемые в сегменте данных (`.data`).

Соответствие размера резервируемой памяти директивами ассемблера коду типов операции архитектуры приведено в таблице ниже.

Необходимо понимать, что директивы резервирования памяти в сегменте данных ассемблера никак не связаны с кодом типа операции. Соответствие определить можно, но одно от другого не зависит.

Тип	**Директива**	**Размер в битах**
byte	.byte	8
short	.short	16
long	.long	32
quad	.quad	64

В ассемблере зависимость от кода типа данных появляется в мнемонике инструкции. Мнемоника инструкции состоит из двух частей:

1. корня, соответствующего коду операции, и
2. суффикса, соответствующего типу операции.

Операции системы команд будут представлены ниже, а мнемонику суффикса показы-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

вает следующая таблица:

Тип	Беззнаковый	Знаковый
byte	b	sb
short	s	ss
long	l	sl
quad	q	sq
float	—	f
double	—	d
pack	—	p
comlex	—	c

5.4 Поле регистров

Процессор имеет следующие типы регистров:

- регистры общего назначения (РОН, GPR — General Purpose Register);
- регистры индексные (РИ, IR — Index Register);
- регистры управляющие (РУ, CR — Control Register).

Биты 4 и 5 номера регистра определяют его тип:

00 Регистр общего назначения
01 Зарезервировано, не используется
10 Регистр индексный
11 Регистр управляющий

5.4.1 Регистры общего назначения

Используются в качестве сверхбыстрой памяти (Scratchpad memory). Имеют размер 64 бита.

5.4.2 Регистры индексные

Имеют следующую логическую структуру:

Номер бита	63	...	48	47	...	32	31	...	0
Описание	Index			Mask			Base		

При записи значения в индексный регистр одновременно устанавливаются все части регистра.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

При использовании индексного регистра в алгоритме операции участвует только значение базы – младшие 32 бита, старшие 32 бита формируются в соответствии с правилами размножения знака для операций, работающими с знаковыми целочисленными типами, во всех остальных операциях сбрасываются в ноль.

При этом, после выполнения параграфа, в котором существует инструкция, использующая индексный регистр, происходит модификация значения индексного регистра по следующей формуле:

$$Index := ((Index|(\sim Mask)) + 1)\&Mask$$

$$Base := Base + Index$$

Операции сложения выполняются по модулю.

Используются для определения смещения при индексном доступе к оперативной памяти. В общем случае, для организации цикла используются два индексных регистра. Один — для вычисления количества итераций цикла, второй — для вычисления смещения на каждом цикле итерации. Шаг смещения, в простейшем случае, является степенью двойки (1, 2, 4, 8, 16, . . .)

5.4.3 Регистры управляющие

Имя	Номер	Доступ	Описание
PSW	00	*RW*	Регистр управления вычислительным процессом
INTR	01	*RW*	Регистр прерываний
MSKR	02	*RW*	Регистр маски прерываний
ER	03	*R*	Регистр исключений
IRETADDR	04	*R*	Регистр адреса возврата
FORCER	05	*W*	Регистр принудительной генерации прерывания
IHOOKADDR	07	*RW*	Регистр первичного обработчика прерываний
INTNUMR	08	*R*	Регистр номера прерывания

5.5 Память

Единицей адресации памяти данных является байт, размером 8 бит. Адресация по всем блокам памяти данных сквозная. При этом аппаратно чтение данных выравнено на 64 бита. Многобайтные значения в памяти данных располагаются в порядке от младшего байта к старшему (little-endian).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Единицей адресации памяти команд является 32 бита. Адресация памяти команд блочная. Значения в блоке памяти команд располагаются в порядке от старшего байта к младшему (big-endian).

5.6 Таблица команд

Большинство инструкций формируют результат, сохраняемый в коммутаторе. Размер значения результата соответствует размеру коммутатора — 64 бит.

Результат нагружен значениями флагов:

Бит	Флаг		Описание
64	ZF (Zero Flag)	Флаг нуля	Результат нулевой
65	OF (Overflow Flag)	Флаг переполнения	Потеря значащего бита
66	CF (Carry / Inexact Flag)	Флаг переноса / Флаг потери точности результата в операциях с числами в формате плавающей точки	Операция произвела перенос из старшего бита результата/При округлении результата произошла потеря значащих битов
67	SF (Sign Flag)	Флаг знака	Состояние старшего бита результата

Флаги могут иметь значение не для каждой операции. В таблице команд приведены значения флагов, которые имеют значение.

Обозначения, используемые в таблице команд:

- все числа представлены в двоичной системе;
- рядом с двоичным значением кода операции в скобочках дано десятичное значение;
- Если число или цифра не используется то на ее знакоместе стоит знак '-';
- Если цифра в числе может принимать любое значение, на ее знакоместе стоит знак 'x';
- В колонке "Флаги" перечислены флаги, которые вырабатывает инструкция, или которые она сохраняет. Т.е. эти флаги, как характеристики значения результата, имеют смысл.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

- Примечания, помеченные символом '*' для не реализованных в аппаратуре типов приведены для общности.

5.6.1 Инструкции работы с памятью

	Cop	Suf	Top	F1	F2	Флаги	Примечание
rd	100001 (33)	xx	00xx	000000	xxxxxx	ZF SF	Считывается столько бит, сколько определено размером типа. Старшие биты нули
		xx	1000	000000	xxxxxx	ZF SF	Считываются 8 бит. 7 бит распространяется в старшие
		xx	1001	000000	xxxxxx	ZF SF	Считываются 16 бит. 15 бит распространяется в старшие
		xx	1010	000000	xxxxxx	ZF SF	Считываются 32 бита. 31 бит распространяется в старшие
		xx	1011	000000	xxxxxx	ZF SF	Считываются 64 бита
		xx	1100	000000	xxxxxx	ZF SF	Старшие биты результата расширяются нулем
		xx	1101	000000	xxxxxx	ZF SF	Старшие биты результата расширяются нулем
		xx	11xx	000000	xxxxxx	ZF	Считываются 64 бита
wr	100010 (34)	xx	xxxx	xxxxxx	xxxxxx	—	Записывается столько бит, сколько определено размером типа

`wr`:

Поле `F1` никогда не может быть равно нулю.

В отличие от большинства других команд, `wr` *не* возвращают результат (оптимизация выполнения).

5.6.2 Инструкции работы с регистрами

Все права защищены © ОАО «Мультиклет» 2010 — 2015

	Cop	Suf	Top	F1	F2	Флаги	Примечание
exa	000010 (02)	01	0010	000000	xxxxxx	ZF	Исполнительный адрес, сформированный индексным регистром
		11	0010	000000	xxxxxx	ZF	Исполнительный адрес, сформированный индексным регистром с полем V
get	000011 (03)	xx	00xx	000000	xxxxxx	ZF SF	Значение обрезается до размера типа. Старшие биты нули
		xx	10xx	000000	xxxxxx	ZF SF	Значение обрезается до размера типа. Распространение знакового бита типа
		xx	1100	000000	xxxxxx	ZF SF	Загрузка значения. Старшие биты нули
		xx	11xx	000000	xxxxxx	ZF	Загрузка значения
set	000100 (04)	00	0011	xxxxxx	xxxxxx	—	Пересылка значения коммутатора в регистр.
		01	0011	xxxxxx	xxxxxx	—	Пересылка значения регистра в регистр.
		10	0010	xxxxxx	000000	—	Запись беззнаковой константы в регистр. Старшие биты нули.
			1010	xxxxxx	000000	—	Запись знаковой константы в регистр. Распространение знакового бита типа.

`exa`:

В качестве операнда имеет смысл только индексный регистр.

`set`:

Интерпретация значения поля `F1` отличается от интерпретации, используемой для других инструкций. Значением поля `F1` является индекс регистра, в который записывается значение, сформированное при интерпретации значений поля `F2` и/или `V`.

В отличие от большинства других команд, *не* возвращают результат (оптимизация вы-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

полнения).

get:
Во всех случаях происходит преобразование типа и/или выставление флагов заново. При формировании значения аргумента инструкция не обращается к памяти данных. Формат команды с суффиксом 11 не имеет смысла, т.к. для этого используется exa. Инструкцию можно использовать для преобразования размеров значения целочисленного типа.

Прочтение мнемокода инструкций:
exa **Ex**ecutive **a**ddress

5.6.3 Арифметические инструкции

	Cop	Suf	Top	F1	F2	Флаги	Примечание
add	000101 (05)	xx	x0xx	xxxxxx	xxxxxx	ZF SF CF OF	Сложение целочисленное.
		xx	11xx	xxxxxx	xxxxxx	ZF SF OF	Сложение вещественное.
sub	000110 (06)	xx	x0xx	xxxxxx	xxxxxx	ZF SF CF OF	Вычитание целочисленное.
		xx	11xx	xxxxxx	xxxxxx	ZF SF OF	Вычитание вещественное.
mul	000111 (07)	xx	x0xx	xxxxxx	xxxxxx	ZF SF CF=OF	Умножение целочисленное.
		xx	11xx	xxxxxx	xxxxxx	ZF SF OF	Умножение вещественное.
adc	001000 (08)	00	00xx	xxxxxx	xxxxxx	ZF SF CF OF	Сложение с переносом.
sbb	001001 (09)	00	00xx	xxxxxx	xxxxxx	ZF SF CF OF	Вычитание с заемом.
insub	001010 (10)	xx	x0xx	xxxxxx	xxxxxx	ZF SF CF OF	Обратное вычитание целочисленное.
		xx	11xx	xxxxxx	xxxxxx	ZF SF OF	Обратное вычитание вещественное.
div	001011 (11)	xx	1100	xxxxxx	xxxxxx	ZF SF OF	Деление вещественное.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

sqrt	001100 (12)	xx	1100	000000	xxxxxx	ZF SF OF	Извлечение квадратного корня вещественное.
max	001101 (13)	xx	x0xx	xxxxxx	xxxxxx	ZF SF	Выбор наибольшего для целочисленных типов.
		xx	110x	xxxxxx	xxxxxx	ZF SF OF	Выбор наибольшего для вещественного.
min	001110 (14)	xx	x0xx	xxxxxx	xxxxxx	ZF SF	Выбор наименьшего для целочисленных типов.
		xx	110x	xxxxxx	xxxxxx	ZF SF OF	Выбор наименьшего для вещественного.
abs	001111 (15)	xx	1000	000000	xxxxxx	ZF OF	Абсолютное значение знакового байта.
		xx	1010	000000	xxxxxx	ZF OF	Абсолютное значение знакового слова.
		xx	110x	000000	xxxxxx	ZF	Абсолютное значение для вещественного.

Замечания:

- Арифметических операций для типов 0011 (unsigned quad) и 1011 (quad) не существует.
- Для инструкций в формате AV значение поля `V` преобразуется к типу операции по правилам для инструкций `set`.
- Поле `F1` не может быть нулём, если не оговорено особо.

Инструкции `adc`, `sbb` используют в качестве операндов арифметической операции значение *второго* аргумента и значение флага `CF` *первого* аргумента.

Значения флагов `CF` и `OF` вычисляются на разрядной сетке типа инструкции. Например, флаг переноса (`CF`) для байтового сложения выставляется при переносе единицы из седьмого бита результата в восьмой.

Для инструкции умножения `mul`, если размер результата получился больше размера операнда, выставляются оба флага `CF` и `OF`.

Прочтение мнемокода инструкций:

adc **Ad**dition with **c**arry flag
sbb **Sub**traction with **b**orrow (CF)
insub **In**verse **sub**traction

Все права защищены © ОАО «Мультиклет» 2010 — 2015

5.6.4 Логические и битовые инструкции

	Cop	Suf	Top	F1	F2	Флаги	Примечание
or	010000 (16)	xx	0011	xxxxxx	xxxxxx	ZF	Логическое сложение.
and	010001 (17)	xx	0011	xxxxxx	xxxxxx	ZF	Логическое умножение.
xor	010010 (18)	xx	0011	xxxxxx	xxxxxx	ZF	Сложение по модулю 2.
not	010011 (19)	xx	0011	000000	xxxxxx	ZF	Отрицание.
norm	010100 (20)	xx	10xx	000000	xxxxxx	ZF	Возвращает значение, используемое при нормализации числа с фиксированной точкой.
pack	010101 (21)	xx	0011	xxxxxx	xxxxxx	ZF	
patch	010110 (22)	xx	0011	xxxxxx	xxxxxx	ZF	
bsr	010111 (23)	xx	00xx	000000	xxxxxx	ZF	Сканирование битов ”назад”.
sll/sal	011000 (24)	xx	x0xx	xxxxxx	xxxxxx	ZF SF CF OF	Логический / арифметический сдвиг влево.
slr	011001 (25)	xx	00xx	xxxxxx	xxxxxx	ZF SF CF OF	Логический сдвиг вправо.
bsf	011010 (26)	xx	00xx	000000	xxxxxx	ZF	Сканирование битов ”вперёд”.
sar	011011 (27)	xx	10xx	xxxxxx	xxxxxx	ZF SF CF OF	Арифметический сдвиг вправо.
rol	011100 (28)	xx	00xx	xxxxxx	xxxxxx	ZF SF CF OF	Циклический сдвиг влево.
ror	011101 (29)	xx	00xx	xxxxxx	xxxxxx	ZF SF CF OF	Циклический сдвиг вправо.

Замечания:

- `Cop` вида ”011xxx” зарезервирован для реализации инструкций сдвигов и переходов (см. ниже).
- Инструкции `or`, `and`, `xor`, `not` реализованы для внутреннего представления (64 бита).
- Инструкции `shl` и `sal` реализованы одинаково, различие в ”знаковости” типа операции можно пренебречь.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все инструкции сдвига выставляют флаги `CF` и `OF` (далее используется: MSB — **M**ost **S**ignificant **B**it):

- Флаг `CF` равен выдвигаемому биту.
- Флаг `OF` определён, если выполнялся сдвиг на один разряд.

`sll`, `sal`, `rol`, `ror`:

$$OF = \begin{cases} 0 & \text{если } CF = MSB, \text{ т.е. два старших бита операнда были одинаковы} \\ 1 & \text{если } CF \neq MSB, \text{ т.е. два старших бита операнда были различны} \end{cases}$$

`sar`:
$OF = 0$

`slr`:
$OF = MSB$(первоначального операнда)

Прочтение мнемокода инструкций:

sll	**S**hift **l**ogical **l**eft
slr	**S**hift **l**ogical **r**ight
sal	**S**hift **a**rithmetic **l**eft
sar	**S**hift **a**rithmetic **r**ight
rol	**Ro**tate **l**eft
ror	**Ro**tate **r**ight
bsr	**B**it **s**can **r**everse
bsf	**B**it **s**can **f**orward

5.6.5 Инструкции передачи управления

	Cop	**Suf**	**Top**	**F1**	**F2**	**Условие**
jmp	011110 (30)	xx	0000	000000	xxxxxx	Безусловный
je		xx	0001	xxxxxx	xxxxxx	$ZF = 1$ ($V_1 = V_2$)
jne		xx	0010	xxxxxx	xxxxxx	$ZF = 0$ ($V_1 \neq V_2$)
js		xx	1000	xxxxxx	xxxxxx	$SF = 1$
jns		xx	1001	xxxxxx	xxxxxx	$SF = 0$
jo		xx	1010	xxxxxx	xxxxxx	$OF = 1$
jno		xx	1011	xxxxxx	xxxxxx	$OF = 0$
jb, jc		xx	0100	xxxxxx	xxxxxx	$CF = 1$ ($U_1 < U_2$, без знака)
jbe		xx	0101	xxxxxx	xxxxxx	$CF = 1$ или $ZF = 1$ ($U_1 \leq U_2$, без знака)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ja		xx	0110	xxxxxx	xxxxxx	$CF = 0$ и $ZF = 0$ ($U_1 > U_2$, без знака)
jae, jnc		xx	0111	xxxxxx	xxxxxx	$CF = 0$ ($U_1 \geq U_2$, без знака)
jl		xx	1100	xxxxxx	xxxxxx	$SF \neq OF$ ($S_1 < S_2$, со знаком)
jle		xx	1101	xxxxxx	xxxxxx	$SF \neq OF$ или $ZF = 1$ ($S_1 \leq S_2$, со знаком)
jg		xx	1110	xxxxxx	xxxxxx	$SF = OF$ и $ZF = 0$ ($S_1 > S_2$, со знаком)
jge		xx	1111	xxxxxx	xxxxxx	$SF = OF$ ($S_1 \geq S_2$, со знаком)

Все инструкции условного перехода, в отличие от большинства других команд, *не* возвращают результат (оптимизация выполнения).

Прочтение мнемокода инструкций:

je **J**ump if **e**qual
jne **J**ump if **n**ot textbfequal
js **J**ump if **s**igned
jns **J**ump if **n**ot **s**igned
jo **J**ump if **o**verflow
jno **J**ump if **n**ot **o**verflow
jc **J**ump if **c**arry
jnc **J**ump if **n**ot **c**arry
jb **J**ump if **b**elow
jbe **J**ump if **b**elow and **e**qual
ja **J**ump if **a**bove
jae **J**ump if **a**bove and **e**qual
jl **J**ump if **l**ess
jle **J**ump if **l**ess and **e**qual
jg **J**ump if **g**reater
jge **J**ump if **g**reater and **e**qual

5.6.6 Инструкции выбора второго аргумента по условию

	Cop	**Suf**	**Top**	**F1**	**F2**	**Условие**
se		xx	0001	xxxxxx	xxxxxx	$ZF = 1$ ($V_1 = V_2$)
sne		xx	0010	xxxxxx	xxxxxx	$ZF = 0$ ($V_1 \neq V_2$)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ss		xx	1000	xxxxxx	xxxxxx	$SF = 1$
sns		xx	1001	xxxxxx	xxxxxx	$SF = 0$
so		xx	1010	xxxxxx	xxxxxx	$OF = 1$
sno		xx	1011	xxxxxx	xxxxxx	$OF = 0$
sb, sc		xx	0100	xxxxxx	xxxxxx	$CF = 1$ ($U_1 < U_2$, без знака)
sbe		xx	0101	xxxxxx	xxxxxx	$CF = 1$ или $ZF = 1$ ($U_1 \leq U_2$, без знака)
sa		xx	0110	xxxxxx	xxxxxx	$CF = 0$ и $ZF = 0$ ($U_1 > U_2$, без знака)
sae, snc		xx	0111	xxxxxx	xxxxxx	$CF = 0$ ($U_1 \geq U_2$, без знака)
sl		xx	1100	xxxxxx	xxxxxx	$SF \neq OF$ ($S_1 < S_2$, со знаком)
sle		xx	1101	xxxxxx	xxxxxx	$SF \neq OF$ или $ZF = 1$ ($S_1 \leq S_2$, со знаком)
sg		xx	1110	xxxxxx	xxxxxx	$SF = OF$ и $ZF = 0$ ($S_1 > S_2$, со знаком)
sge		xx	1111	xxxxxx	xxxxxx	$SF = OF$ ($S_1 \geq S_2$, со знаком)

Прочтение мнемокода инструкций:

se **S**end value to switchboard if **e**qual
sne **S**end value to switchboard if **n**ot textbfequal
ss **S**end value to switchboard if **s**igned
sns **S**end value to switchboard if **n**ot **s**igned
so **S**end value to switchboard if **o**verflow
sno **S**end value to switchboard if **n**ot **o**verflow
sc **S**end value to switchboard if **c**arry
snc **S**end value to switchboard if **n**ot **c**arry
sb **S**end value to switchboard if **b**elow
sbe **S**end value to switchboard if **b**elow and **e**qual
sa **S**end value to switchboard if **a**bove
sae **S**end value to switchboard if **a**bove and **e**qual
sl **S**end value to switchboard if **l**ess
sle **S**end value to switchboard if **l**ess and **e**qual
sg **S**end value to switchboard if **g**reater
sge **S**end value to switchboard if **g**reater and **e**qual

Все права защищены © ОАО «Мультиклет» 2010 — 2015

5.6.7 Инструкции работы с битами аргумента

	Cop	Suf	Top	F1	F2	Флаги	Примечание
bc	100000 (32)	xx	00xx	xxxxxx	xxxxxx	ZF	Вычисление числа единичных битов в аргументе.
fcpy	100011 (35)	xx	0011	xxxxxx	xxxxxx	ZF SF OF CF	Копирование флагов первого аргумента во второй.

5.6.8 Инструкция развёртки байтов

	Cop	Suf	Top	F1	F2	Флаги	Примечание
ib	100100 (36)	xx	x000	xxxxxx	xxxxxx	ZF SF OF CF	

5.6.9 Инструкция свёртки байтов

	Cop	Suf	Top	F1	F2	Флаги	Примечание
cb	100101 (37)	xx	x000	xxxxxx	xxxxxx		

5.6.10 Инструкция свёртки байтов с насыщением

	Cop	Suf	Top	F1	F2	Флаги	Примечание
cbs	100110 (38)	xx	x000	xxxxxx	xxxxxx		

5.6.11 Инструкции выбора первого или второго аргумента по условию

	Cop	Suf	Top	F1	F2	Условие
me		xx	0001	xxxxxx	xxxxxx	$ZF = 1$ $(V_1 = V_2)$
mne		xx	0010	xxxxxx	xxxxxx	$ZF = 0$ $(V_1 \neq V_2)$
ms		xx	1000	xxxxxx	xxxxxx	$SF = 1$
mns		xx	1001	xxxxxx	xxxxxx	$SF = 0$
mo		xx	1010	xxxxxx	xxxxxx	$OF = 1$
mno		xx	1011	xxxxxx	xxxxxx	$OF = 0$

Все права защищены © ОАО «Мультиклет» 2010 — 2015

mb, mc		xx	0100	xxxxxx	xxxxxx	$CF = 1$ ($U_1 < U_2$, без знака)
mbe		xx	0101	xxxxxx	xxxxxx	$CF = 1$ или $ZF = 1$ ($U_1 \leq U_2$, без знака)
ma		xx	0110	xxxxxx	xxxxxx	$CF = 0$ и $ZF = 0$ ($U_1 > U_2$, без знака)
mae, mnc		xx	0111	xxxxxx	xxxxxx	$CF = 0$ ($U_1 \geq U_2$, без знака)
ml		xx	1100	xxxxxx	xxxxxx	$SF \neq OF$ ($S_1 < S_2$, со знаком)
mle		xx	1101	xxxxxx	xxxxxx	$SF \neq OF$ или $ZF = 1$ ($S_1 \leq S_2$, со знаком)
mg		xx	1110	xxxxxx	xxxxxx	$SF = OF$ и $ZF = 0$ ($S_1 > S_2$, со знаком)
mge		xx	1111	xxxxxx	xxxxxx	$SF = OF$ ($S_1 \geq S_2$, со знаком)

Прочтение мнемокода инструкций:

me **M**ove argument to switchboard if **e**qual
mne **M**ove argument to switchboard if **n**ot textbfequal
ms **M**ove argument to switchboard if **s**igned
mns **M**ove argument to switchboard if **n**ot **s**igned
mo **M**ove argument to switchboard if **o**verflow
mno **M**ove argument to switchboard if **n**ot **o**verflow
mc **M**ove argument to switchboard if **c**arry
mnc **M**ove argument to switchboard if **n**ot **c**arry
mb **M**ove argument to switchboard if **b**elow
mbe **M**ove argument to switchboard if **b**elow and **e**qual
ma **M**ove argument to switchboard if **a**bove
mae **M**ove argument to switchboard if **a**bove and **e**qual
ml **M**ove argument to switchboard if **l**ess
mle **M**ove argument to switchboard if **l**ess and **e**qual
mg **M**ove argument to switchboard if **g**reater
mge **M**ove argument to switchboard if **g**reater and **e**qual

Все права защищены © ОАО «Мультиклет» 2010 — 2015

5.6.12 Инструкция передачи значения из коммутатора с сохранением флагов

	Cop	Suf	Top	F1	F2	Флаги	Примечание
move	101000 (40)	xx	0011	xxxxxx	xxxxxx	ZF SF OF CF	Сохраняются флаги аргумента

5.6.13 Инструкции преобразования типов

	Cop	Suf	Top	F1	F2	Флаги	Примечание
cdf	110111 (55)	xx	1100	000000	xxxxxx	ZF OF SF	$double \rightarrow float$
clf	111000 (56)	xx	1100	000000	xxxxxx	ZF OF	$long \rightarrow float$
cfd	111001 (57)	xx	1100	000000	xxxxxx	ZF OF SF	$float \rightarrow double$
cslf	111010 (58)	xx	1100	000000	xxxxxx	ZF OF SF	$signed\,long \rightarrow float$
cfsl	111011 (59)	xx	1010	000000	xxxxxx	ZF OF SF	$float \rightarrow signed\,long$

Прочтение мнемокода инструкций:

cdf **C**onvert **d**ouble to **f**loat

clf **C**onvert **l**ong to **f**loat

cfd **C**onvert **f**loat to **d**ouble

cslf **C**onvert **s**igned **l**ong to **f**loat

cfsl **C**onvert **f**loat to **s**igned **l**ong

5.6.14 Мультимедийные инструкции

Мультимедийные команды выполняются над упакованными значениями. В данной группе команд `Top` может не совпадать с типом результата.

	Cop	Suf	Top	F1	F2	Флаги	Примечание
madd	110001 (49)	xx	1110	xxxxxx	xxxxxx	ZF OF SF	Упакованное умножение со сложением.

`madd`:

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Для двух упакованных чисел $[a, b]$, $[c, d]$ выполняет операцию:

$$r = a \cdot c + b \cdot d$$

Флаги выставляются аналогично инструкции `add` с типом `float`.

Прочтение мнемокода инструкций:

madd Packed **m**ultiply and **add**

Все права защищены © ОАО «Мультиклет» 2010 — 2015

6 Организация памяти

Память для МП с программной точки зрения представляет собой массив с байтовой адресацией, диапазон адресов: $[0, 2^{32}]$. Память не разделена на страницы и доступна вся сразу. Так же нет разделения на ПП(память программ) и ПД(память данных), но есть ограничения по размещению данных и программ, которые описаны ниже.

6.1 Карта памяти

На рис.8 приведена карта памяти МП. На карте памяти можно выделить несколько зон:

- память на кристалле МП;
- внешнее ОЗУ;
- внешнее ПЗУ;
- область устройств ввода/вывода;
- область отображения адресов 0x00000000-0x40000000;
- шина периферийных устройств.

Память на кристалле МП.

Представляет собой СОЗУ.

В диапазоне 0x00000000-0x00040000 память разбита на одинаковые по объему 4 региона. Физически это 32 блока 33×2^{11}, которые позволяют иметь одновременный доступ к данным, расположенным в соседних блоках. 33 бит не доступен программно и используется в аппаратной системе отладки ПО (п.8). Данный блок специализирован для размещения в нем исполняемого кода. В нем можно разместить любые программы с любым сочетанием типов команд. Так же в нем можно размещать и данные. Так же, только в этом диапазоне адресов можно поставить аппаратные точки останова для инструкций.

Номер блока можно вычислить по следующей формуле:

$$n = \left\lceil \frac{A_b}{2^{16}} \right\rceil + \left\lceil \frac{A_b \ mod \ 32}{8} \right\rceil, \ A_b = [0, 2^{18} - 1]$$

Все права защищены © ОАО «Мультиклет» 2010 — 2015

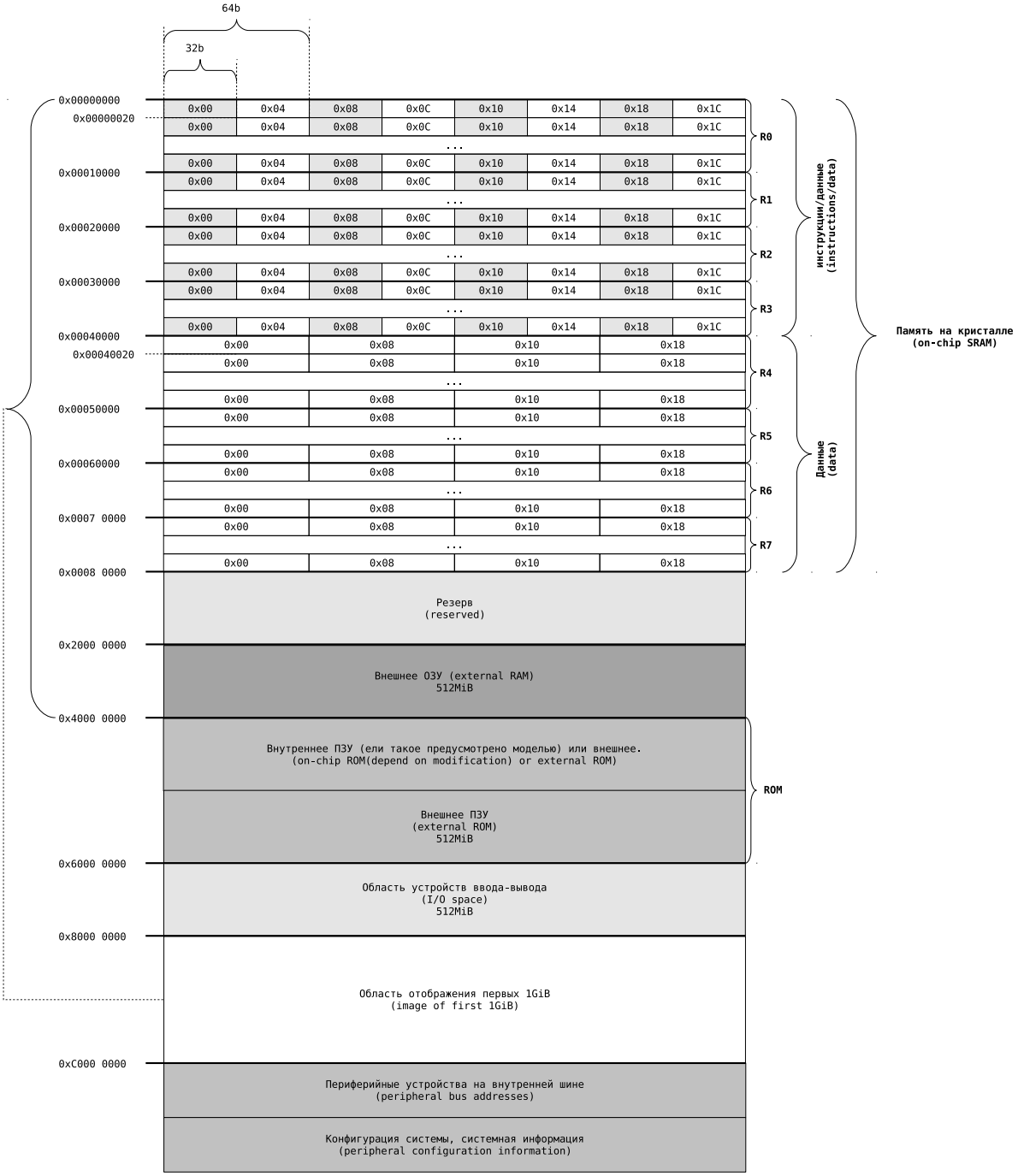

Рис. 8: Карта памяти

Адрес в байтах, округление результатов деления производится к наименьшему целому числу.

В диапазоне 0x00040000-0x00080000 память разбита на одинаковые по объему 4 региона. Но физически представляет собой 16 блоков 64×2^{11}, которые позволяют иметь одновременный доступ к данным, расположенным в соседних блоках. Не рекомендуется

Все права защищены © ОАО «Мультиклет» 2010 — 2015

передавать управление на адреса в данном блоке. В нем можно хранить программы, но возможно существенное снижение производительности, т.к. клетки, возможно, будут обращаться к одному и тому же физическому блоку, что будет причиной простоя.

Номер блока можно вычислить по следующей формуле:

$$n = \left\lceil \frac{A_b}{2^{16}} \right\rceil + \left\lceil \frac{A_b \ mod \ 32}{4} \right\rceil, \ A_b = [2^{18}, 2^{19} - 1]$$

Адрес в байтах, округление результатов деления производится к наименьшему целому числу.

Внешнее ОЗУ, внешнее ПЗУ, область устройств ввода/вывода

Для доступа к данным зонам используется специализированная шина внешней памяти. Описание интерфейса приведено в главе о периферийных устройствах (п.9). Возможно выполнять, хранить данные в данных зонах. Но следует учитывать, что это существенно снизит производительность МП, т.к. 4 клетки за один такт могут запрашивать в памяти до 32 байт информации, максимальная ширина шины внешней памяти - 32 бита (4 байта), скорость работы зависит от внешних компонентов и тактовой частоты МП.

Область отображения адресов

Данная зона используется контроллером ПДП на периферийной шине. ЦПУ так же может обратиться по адресам в этой зоне. Описание контроллера приведено в главе о периферийных устройствах (п.**??**). Контроллер является «ведомым» устройством на периферийной шине.

Шина периферийных устройств

В данной зоне расположены регистры периферийных устройств. В последних адресах расположена информация о конфигурации периферии МП.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

6.2 Коммутатор

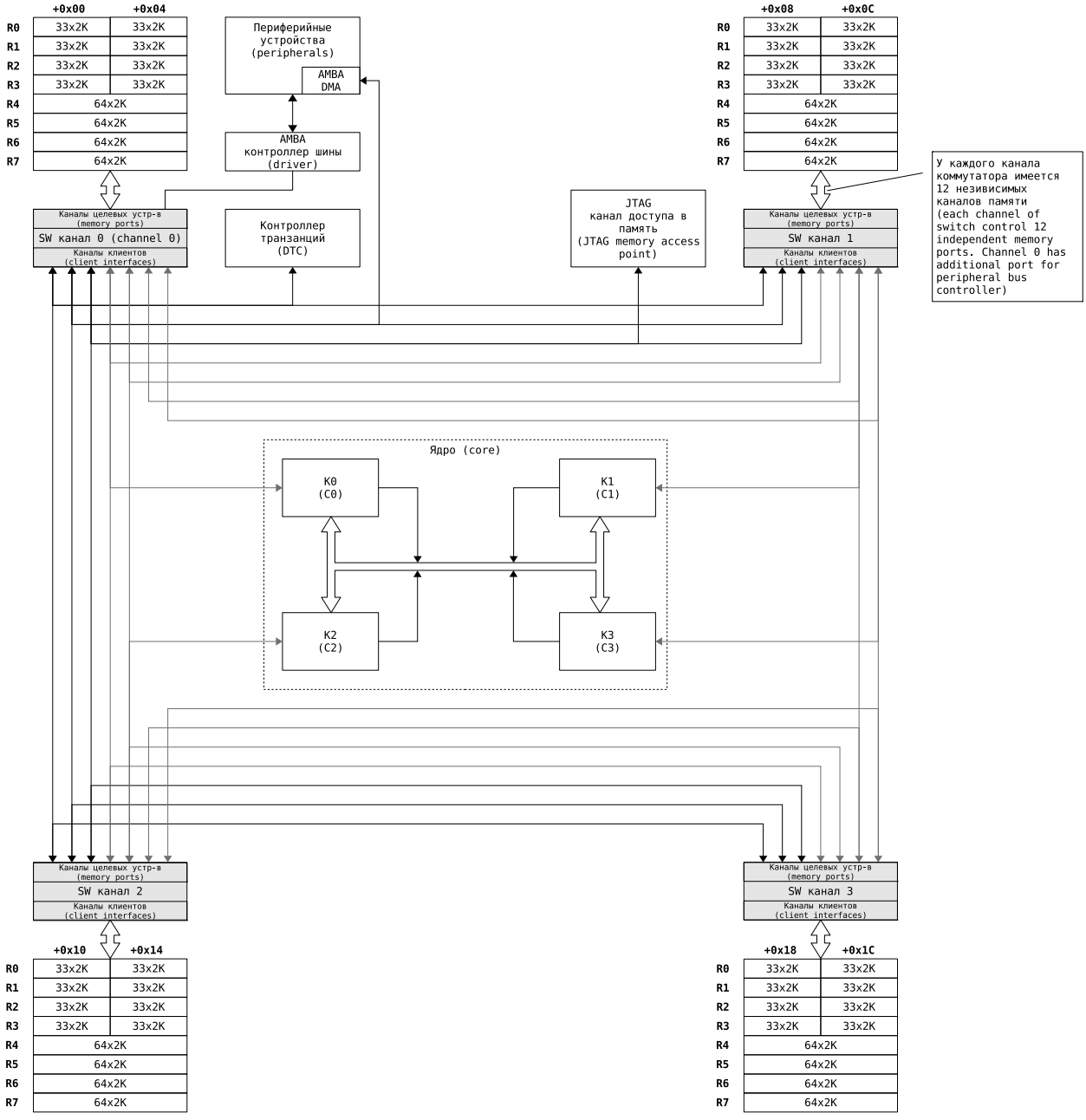

Рис. 9: Структура коммутатора

Коммутатор физически разделен на 4 канала, которые управляют своими участками памяти. Каждый блок представляет систему с 15-ю клиентскими портами и 12-ю портами доступа к памяти. Каналы памяти независимы и могут обслуживать клиентские запросы одновременно. В «0» канале коммутатора имеется дополнительный канал выхода на шину периферийных устройств.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Номер канала коммутатора, который обслуживает адрес, можно найти по следующей формуле:

$$n = \left\lceil \frac{A_b \ mod \ 32}{4} \right\rceil$$

Адрес в байтах, округление результата деления производится к наименьшему целому числу.

Запрос памяти от клиента подается на все каналы коммутатора одновременно. Канал воспринимает только тот запрос, адрес которого он обслуживает. Если несколько клиентских каналов запросили данные с одного и того же адреса, то запросы будут обслуживаться согласно приоритетам.

№	Название клиентского канала	Примечание
0	C0_I_0	Каналы запроса инструкций клетками. Каждый канал запрашивает 32 бита инструкции.
1	C0_I_1	
2	C1_I_0	
3	C1_I_1	
4	C2_I_0	
5	C2_I_1	
6	C3_I_0	
7	C3_I_1	
8	C0_D	Каналы запроса данных клетками
9	C1_D	
10	C2_D	
11	C3_D	
12	DTC_INDEX	Канал запроса от контроллера транзакций
13	AMBA_DMA_INDEX	Канал запроса от DMA канала периферийной шины
14	JTAG_MEM_INDEX	Канал запроса от аппаратной системы отладки по JTAG

Таблица 14: Клиентские каналы коммутатора

№	Название порта памяти	Примечание
0	R0_0	Порты блоков памяти регионов 0-3. 32-х битные блоки памяти.
1	R0_1	
2	R1_0	
3	R1_1	
4	R2_0	
5	R2_1	
6	R3_0	
7	R3_1	
8	R4	Порты блоков памяти регионов 4-7 64-х битные блоки памяти.
9	R5	
10	R6	
11	R7	
12	AMBA	Порт шины AMBA, обслуживающей адресное пространство после R7

Таблица 15: Порты памяти каналов коммутатора

Коммутатор в каждом канале имеет аппарат определения приоритетов. Клиентские каналы разделены на 4 типа:

Все права защищены © ОАО «Мультиклет» 2010 — 2015

- каналы инструкций;
- каналы данных;
- канал AMBA;
- канал DTC.

Каждому порту памяти в канале коммутатора устанавливается приоритет по типу канала. Внутри каналов инструкций и данных приоритет фиксированный и зависит от номера клетки осуществляющей запрос. Настройка приоритета осуществляется системным регистром SWCHx_CTRL (где x-номер канала коммутатора). Также при определении приоритетного канала учитывается время, когда запрос поступил. Т.е. один клиентский канал не может постоянно удерживать канал коммутатора. Если такому каналу запрашивается массив данных, то при возникновении запроса от каналов с меньшим приоритетом, он будет разрывать длинный запрос от канала с наибольшим приоритетом. Например, если порядок запроса с одного и того же адреса клиентскими каналами был следующим (в скобках запросы, возникшие в одно время): 0; (3, 2); 1; 3; 3 и при этом запрос от канала 0 длится достаточно долго, что успевают прийти все запросы от менее приоритетных каналов, в последовательности, указанной выше, то порядок обслуживания будет следующим: 0, 2, 0, 1, 0, 3, 0, 3, 0, 3, 0 ....

SWCHx_CTRL	**Регистр управления каналом коммутатора**												
Номер бита	25 ... 24	23 ... 22	21 ... 20	19 ... 18	17 ... 16	15 ... 14	13 ... 12	11 ... 10	9 ... 8	7 ... 6	5 ... 4	3 ... 2	1 ... 0
Описание	AMBA	R7	R6	R5	R4	R3_1	R3_0	R2_1	R2_0	R1_1	R1_0	R0_1	R0_0

25 ... 24	AMBA	Для всех полей:
23 ... 22	R7	11 - приоритет канала AMBA (№13)
21 ... 20	R6	10 - приоритет канала DTC (№12)
19 ... 18	R5	01 - приоритет каналов запроса данных (№№ 8 ... 11)
17 ... 16	R4	00 - приоритет каналов запроса инструкций (№№ 0 ... 7)
15 ... 14	R3_1	
13 ... 12	R3_0	
11 ... 10	R2_1	
9 ... 8	R2_0	
7 ... 6	R1_1	
5 ... 4	R1_0	
3 ... 2	R0_1	
1 ... 0	R0_0	

Все права защищены © ОАО «Мультиклет» 2010 — 2015

6.3 Контроллер транзакций данных (DTC)

Контроллер транзакций (перемещения) данных служит для перемещения блоков данных заданного размера с адреса источника в адрес приемника.

Каждая транзакция состоит из последовательности блоков, каждый из которых, в свою очередь, состоит из последовательности элементов. Элемент транзакции может быть размером один, два или четыре байта. В пределах блока элементы размещаются последовательно, без пропусков. "Расстояние"от начала одного блока до начала следующего конфигурируется. Таким образом, блоки могут размещаться как строго последовательно, так и с разрывом либо, наоборот, с наложением.

Запуск процесса копирования данных может осуществляться как программно, так и аппаратно, по прерываниям. При этом по каждому прерыванию может быть скопирован один элемент, один блок, либо проведена вся транзакция целиком.

Регистр управления DTC_CTRL

DTC_CTRL	Регистр управления						
Номер бита	31	30 . . . 27	26	25 . . . 24	23 . . . 12	11 . . . 2	1 . . . 0
Описание	START	MSK	REG	MODE	NUMBLK	NUMEL	ELSIZE

31	START	Установка этого бита в 1 запускает DTC
30 . . . 27	MSK	Маска битов регистра состояния, генерирующих прерывание DTC. Соответствует битам TC, TBC, TEC и EMERR
26	REG	При установке в 1 указывает, что данные помещаются в регистр DTC_DATA После обработки очередного элемента транзакция приостанавливается
25 . . . 24	MODE	Режим работы: 00 — программный запуск транзакции; 01 — запуск по прерыванию и приостановка по окончании передачи одного элемента; 10 — запуск по прерыванию и приостановка по окончании передачи одного блока; 11 — запуск по прерыванию и приостановка по окончании передачи всех блоков
23 . . . 12	NUMBLK	Число блоков в транзакции
11 . . . 2	NUMEL	Число элементов в блоке
1 . . . 0	ELSIZE	Размер элемента: 00 — 1 байт; 01 — 2 байта; 10 — 4 байта

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Регистр состояния DTC_ST

При записи в регистр состояния биты, в которые записывается 1, сбрасываются в 0. Сброс бита TIP приводит к принудительному завершению транзакции и переустановке всех внутренних счётчиков.

При записи в регистр состояния биты, в которые записывается 1, сбрасываются в 0. Если биты MODE регистра управления установлены в значение, отличное от "00 запись 1 в бит TIP регистра статуса приводит к прерыванию транзакции. В случае работы DTC в режиме программного запуска (MODE=00) прерывание транзакции не предусмотрено.

DTC_ST	Регистр состояния							
Номер бита	31	30	29	28	27	26 . . . 16	15 . . . 10	9 . . . 0
Описание	TIP	TC	TBC	TEC	EMERR	*зарезервировано*	NBLK	NEL

31	TIP	идёт выполнение транзакции
30	TC	транзакция завершена
29	TBC	передача очередного блока завершена
28	TEC	передача очередного элемента завершена
27	EMERR	при выполнении транзакции обнаружена ошибка доступа к внешней памяти
15 . . . 10	NBLK	число переданных блоков
9 . . . 0	NEL	число переданных элементов текущего блока

Регистр маски прерываний DTC_IMASK

Регистр DTC_IMASK состоит из двух регистров, DTC_IMASK0 и DTC_IMASK1, и доступен как для чтения, так и для записи. Для разрешения работы DTC по прерываниям с определенными номерами, необходимо выставить в 1 биты данного регистра, номера которых соответствуют номерам прерываний. При этом биты 63 . . . 32 содержатся в IMASK1, а биты 31 . . . 0 — в IMASK0.

DTC_IMASK0	Регистр маски прерываний
Номер бита	31 . . . 0
Описание	IMASK0

DTC_IMASK1	Регистр маски прерываний
Номер бита	31 . . . 0
Описание	IMASK1

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Регистр данных DTC_DATA

DTC_DATA	Регистр данных
Номер бита	31 . . . 0
Описание	DATA

В данном регистре размещается очередной считанный элемент. Если установлен бит REG регистра DTC_CTRL, после чтения очередного элемента и помещения его в DTC_DATA, устанавливается бит TEC в регистре состояния и транзакция приостанавливается. Для продолжения работы DTC необходимо сбросить этот бит.

Регистры DTC_S_ADDR и DTC_D_ADDR

DTC_S_ADDR	Регистр адреса источника
Номер бита	31 . . . 0
Описание	DTC_S_ADDR

DTC_D_ADDR	Регистр адреса приёмника
Номер бита	31 . . . 0
Описание	DTC_D_ADDR

Регистр шага адресов DTC_STEP

DTC_STEP	Регистр шага адресов		
Номер бита	31 . . . 20	19 . . . 10	9 . . . 0
Описание	*зарезервировано*	STEP_DST	STEP_SRC

19 . . . 10	STEP_DST	приращение адреса начала блока приемника
9 . . . 0	STEP_SRC	приращение адреса начала блока источника

Все права защищены © ОАО «Мультиклет» 2010 — 2015

7 Система тактирования

В составе МП имеется генератор для формирования системного сигнала тактирования. Генератор формирует стабильный тактовый сигнал в диапазоне частот от 20МГц до 300МГц. Синтезатор основан на схеме ФАПЧ с целочисленными коэффициентами деления синтезируемой частоты. Частота опорного сигнала от 8 до 15 МГц.

Возможна работа без задействования встроенного генератора, используя источник внешнего опорного сигнала тактирования. В качестве источника может быть выбран генератор или кварцевый резонатор.

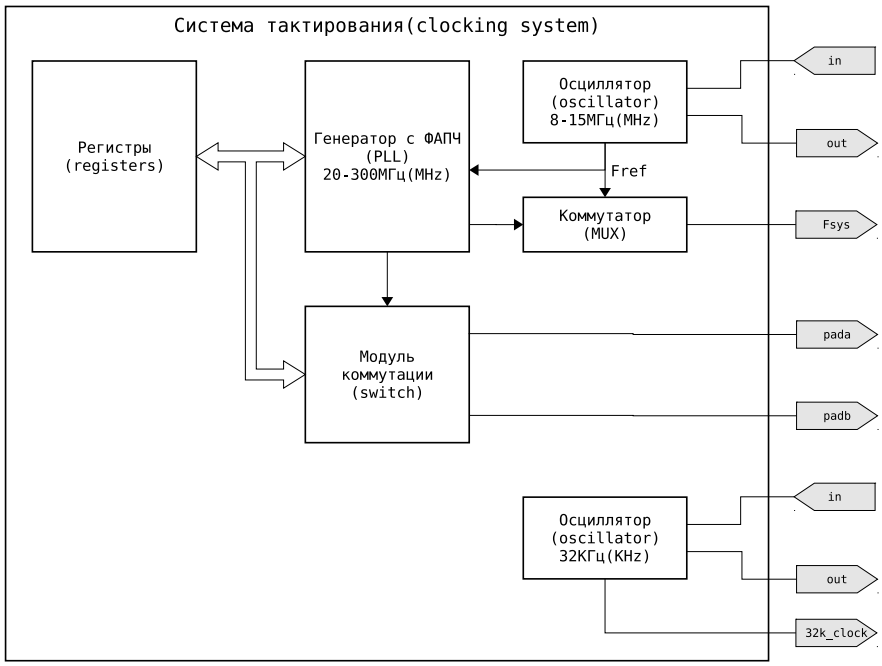

Рис. 10: Система тактирования

На рис.10 приведена блок-схема системы тактирования МП. В МП существует 2 частотных домена: домен системной частоты F_{sys} который тактирует все системы МП и домен альтернативной тактовой частоты для счетчиков часов реального времени (ЧРВ).

7.0.1 Описание работы системы тактирования

После снятия сигнала системного сброса система работает на частоте источника опорного сигнала тактирования. Внутренний генератор должен быть сконфигурирован и включен программно.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Частота F_{sys} рассчитывается по следующей формуле:

$$F_{out} = F_{ref} \cdot N \cdot \frac{1}{R} \cdot \frac{1}{K}$$

- $R = 1, 2, 4, 8$ - коэффициент деления опорной частоты (определяется битами PLLCR(RCNT));
- $N = [2, 131]$ - коэффициент деления внутреннего генератора (определяется битами PLLCR(NCNT));
- $K = 1, 2, 4, 8, 16$ - коэффициент деления выходного делителя (определяется битами PLLCR(MXVCO)).

При этом частота сравнения (на входе фазо-частотного детектора):

$$F_{cmp} = \frac{1}{R} \cdot F_{ref}$$

Частота внутреннего генератора, управляемого напряжением (ГУН):

$$F_{vco} = N \cdot \frac{1}{R} \cdot F_{ref}$$

Выходная частота:

$$F_{out} = \frac{1}{K} \cdot F_{vco}$$

Диапазон работоспособности блока задается следующими условиями:

$$1MHz < F_{cmp} < 16MHz$$

$$20MHz < F_{vco} < 340MHz$$

Пример расчета.

$F_{ref} = 15MHz$ и требуется выходная частота около $130MHz$. Задаем $R = 1$. Далее выбираем $K = 2$, тогда ГУН будет работать на частоте около $2 \cdot 130MHz = 260MHz$. Определяем N: $\frac{260}{15MHz} = 17,33$. Тогда получаем:

- при $R = 1, K = 2, N = 17, F_{out} = \frac{17 \cdot 15MHz}{2} = 127,5MHz$.
  В этом случае PLLCR(RCNT)=000b, PLLCR(MXVCO)=1100b, PLLCR(RCNT)=0x0D.
- при $R = 1, K = 2, N = 18, F_{out} = \frac{18 \cdot 15MHz}{2} = 135MHz$.
  В этом случае PLLCR(RCNT)=000b, PLLCR(MXVCO)=1100b, PLLCR(RCNT)=0x0E.

7.0.2 Описание работы модуля коммутации

Все права защищены © ОАО «Мультиклет» 2010 — 2015

7.0.3 Таблицы выбора коэффициентов R, N, K

Коэффициент R	Биты управления делителем опорной частоты			
	rcnt(2)	rcnt(1)	rcnt(0)	Значение R
	0	X	X	1
	1	0	0	2
	1	0	1	4
	1	1	0	8
	1	1	1	16

Коэффициент N	Биты управления делителя в петле обратной связи							
	ncnt(6)	ncnt(5)	ncnt(4)	ncnt(3)	ncnt(2)	ncnt(1)	ncnt(0)	Значение N
	0	0	0	0	0	0	0	2
	0	0	0	0	0	0	1	3
	0	0	0	0	0	1	0	6
	0	0	0	0	0	1	1	7
	0	0	0	0	1	0	0	8
	...	...	...	...	...	...	...	...
	1	1	1	1	1	1	0	130
	1	1	1	1	1	1	1	131

На основании данной таблицы можно составить формулу:

При ncnt(0 ... 6) = 0 следует, что N = 2

При ncnt(0 ... 6) = 1 следует, что N = 3

При ncnt(0 ... 6) = 2 ... 127 следует, что N = ncnt(0 ... 6) + 4

Коэффициент K	Биты управления мультиплексором выходного сигнала				
	mxvco(3)	mxvco(2)	mxvco(1)	mxvco(0)	Значение K
	0	X	X	X	Fref
	1	0	X	X	1
	1	1	0	0	2
	1	1	0	1	4
	1	1	1	0	8
	1	1	1	1	16

7.0.4 Порядок работы с блоком PLL

Включение после сброса или подачи питания

1) Установить биты PLLCR(NCNT) и PLLCR(RCNT) в нужное значение

2) Установить бит PLLCR(EN) в '1'

3) Дождаться установки бита PLLSTR(LOCK) в '1'

4) Переключиться на нужную частоту установкой битов PLLCR(MXVCO)

Смена частоты

1) Установить третий бит MXVCO, т.е. PLLCR[11] в '0' , чтобы переключиться на Fref

2) Изменить биты PLLCR(NCNT) и PLLCR(RCNT)

3) Установить третий бит MXVCO, т.е. PLLCR[11] в '1' , чтобы переключиться на новую частоту

Выключение блока

1) Установить третий бит MXVCO, т.е. PLLCR[11] в '0' , чтобы переключиться на Fref

2) Дождаться установки бита PLLSTR(REFACTIVE) в '1'

Все права защищены © ОАО «Мультиклет» 2010 — 2015

3) Установить бит PLLCR(EN) в ’0’

Все права защищены © ОАО «Мультиклет» 2010 — 2015

7.0.5 Регистры системы тактирования

PLLCR	Регистр управления генератора										
Номер бита	31 … 24	23	22	21	20	19 … 18	17 … 16	15	14 … 12	11 … 8	6 … 0
Описание	—	LOCK_LS	LOCK_RANGE	PDDLY	IRMODE	CPCTR	LOCKDLY	EN	RCNT	MXVCO	NCNT

31… 24	—	*зарезервировано*
	Параметры для подстройки генератора	
23	LOCK_LS	настройки блока детектора захвата: 0 - при частоте после делителя опорного сигнала [4,20]МГц 1 - при частоте после делителя опорного сигнала [1,8]МГц
22	LOCK_RANGE	настройка блока детектора захвата (разъяснение по использованию см. ниже)
21	PDDLY	изменение длительности сигнала сброса фазочастотного детектора, нс: 0 - 1нс 1 - 0,5нс
20	IRMODE	настройка тока: 0 - выходной ток температурно-стабилизированный, с низкой зависимостью от напряжения питания vdda 1 - выходной ток температурно-стабилизированный, с высокой зависимостью от напряжения питания vdda
19 … 18	CPCTR	выбор величины тока в блоке ГПЗ, мкА 00 - 4; 01 - 2; 10 - 8; 11 - 6;
17 … 16	LOCKDLY	настройка детектора захвата опорной частоты, задержка в тактах опорной частоты до установления сигнала lock: 00 - 8; 01 - 16; 10 - 32; 11 - 64;
	Биты управления генератором	
15	EN	включение блока
14… 12	RCNT	коэффициент деления опорной частоты
11… 8	MXVCO	коэффициент деления выходного делителя
6… 0	NCNT	коэффициент деления внутреннего генератора

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Пояснение по использованию бита PLLCR(LOCK_RANGE).

Блок детектора захвата опорной частоты устанавливает выходной сигнал PLLSTR(LOCK) в '1', когда длительность сигнала ошибки по частоте сформированного сигнала F_{ref} не превышает время T_{error}. Время T_{error} зависит от коэффициента деления PLLCR(RCNT), значения сигналов PLLCR(LOCK_LS) и PLLCR(LOCKDLY) согласно формуле:

$$T_{error} = min(T_{error\%}, T_{errorA})$$

Таблица выбора значения $T_{error\%}$

PLLCR(RCNT)	$T_{error\%}$ (PLLCR(LOCK_RANGE) = '0')	$T_{error\%}$ (PLLCR(LOCK_RANGE) = '1')
значения $T_{error\%}$ даны в процентах от периода F_{ref}		
1	1,5%	8%
2	2%	12%
4	3%	14%
8	3%	15%
16	3%	16%

Таблица выбора значения T_{errorA}

PLLCR(LOCK_RANGE)	T_{errorA} (PLLCR(LOCK_LS) = '0')	T_{errorA} (PLLCR(LOCK_LS) = '1')
1	7нс	12нс
2	40нс	70нс

Пример настройки параметров генератора.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

PLLSTR	Регистр состояния генератора			
Номер бита	31...3	2	1	0
Описание	—	REFACTIVE	FREFLOW	LOCK

31... 3	—	*зарезервировано*
2	REFACTIVE	наличие опорного тактового сигнала на входе генератора ('1' - подано)
1	FREFLOW	частота опорного тактового сигнала ниже 150кГц ('1' - частота ниже 150кГц, '0' - частота больше 190кГц)
0	LOCK	частота на выходе генератора стабильно ('1' - частота стабильна)

PLLTMCR	Регистр управления модулем коммутации									
Номер бита	31...24	23	22...15	14...11	10...7	6	5	4...2	1	0
Описание	—	EN	—	SELA	SELB	CFGOA	CFGOB	SELOCUR	INVERTA	INVERTB

31...24	—	*зарезервировано*
23	EN	разрешение работы ('1' - включен, '0' - выключен)
22...15	—	*зарезервировано*
14...11	SELA	управление мультиплексором канала A: 0100 - F_{ref} 1001 - F_{sys} остальные значения зарезервированы
10...7	SELB	управление мультиплексором канала B: 0100 - F_{ref} 1001 - F_{sys} остальные значения зарезервированы
6	CFGOA	управление состоянием выходного буфера A ('0' - значение тока выходного буфера зависит от значения PLLTMCR(SELOCUR) '1' значенине выходного тока - 240мкА)
5	CFGOB	управление состоянием выходного буфера B ('0' - значение тока выходного буфера зависит от значения PLLTMCR(SELOCUR) '1' значенине выходного тока - 240мкА)
4...2	SELOCUR	управление током выходных буферов 000 - 2,0 мА 001 - 4,8 мА 010 - 7,7 мА 011 - 10,6 мА 100 - 4,0 мА 101 - 9,6 мА 110 - 15,4 мА 111 - 21,2 мА
1	INVERTA	инвертирование сигнала канал A('1' - инвертирован)
0	INVERTB	инвертирование сигнала канал B('1' - инвертирован)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

8 Аппаратная система отладки и тестирования

В состав МП входят блоки обеспечивающие доступ к аппаратной части при отладке и тестировании. Доступ к данным блокам осуществляется по интерфейсу JTAG. Компоненты JTAG, входящие в состав МП поддерживают все обязательные команды по стандарту ieee1149.1. Дополнительно введены команды для обеспечения работы специализированных блоков, обеспечивающих такие возможности как доступ к памяти МП, внутренней шине периферийных устройств, регистрам и блокам управления клетками в режиме отладки ПО.

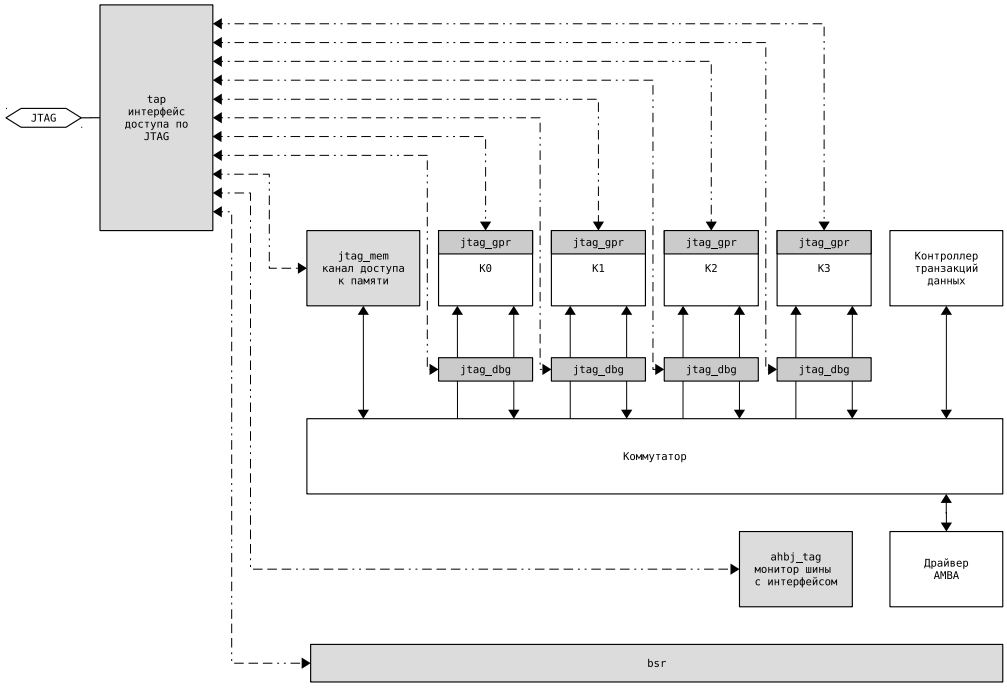

Рис. 11: Компоненты системы отладки и тестирования в МП

8.1 Контроллер (tap)

Интерфейс доступа по стандарту ieee1149.1 (JTAG). Контроллер сконфигурирован на поддержку 11 каналов, т.е. есть доступ к 11 блоками, обеспечивающим тестирование МП и отладку ПО. На рис.11 они изображены с заливкой.

0	jtag_gpr0
1	jtag_gpr1
2	jtag_gpr2
3	jtag_gpr3
4	jtag_dbg0

Все права защищены © ОАО «Мультиклет» 2010 — 2015

5	jtag_dbg1
6	jtag_dbg2
7	jtag_dbg3
8	bsr
9	ahb_jtag
10	jtag_mem

Длина инструкций, поддерживаемых tap контроллером, составляет - 8 бит.

Команда	**код (HEX)**	**Блок, к которому относится команда**
SAMPLE	0x01	bsr
PRELOAD	0x02	
IDCODE	0x04	
HIGHZ	0x05	
INTEST	0x08	
EXTEST	0x09	
BYPASS	0xff	
AMBA_ADDR	0x80	ahb_jtag
AMBA_DATA	0x81	
MEM_ADDR	0xC0	jtag_mem
MEM_DATA	0xC1	
CELL0_REGS_ADDR	0xC4	jtag_gpr0
CELL1_REGS_ADDR	0xC5	jtag_gpr1
CELL2_REGS_ADDR	0xC6	jtag_gpr2
CELL3_REGS_ADDR	0xC7	jtag_gpr3
CELL0_REGS_DATA	0xC8	jtag_gpr0
CELL1_REGS_DATA	0xC9	jtag_gpr1
CELL2_REGS_DATA	0xCA	jtag_gpr2
CELL3_REGS_DATA	0xCB	jtag_gpr3
DBG0_ADDR	0xE0	jtag_dbg0
DBG1_ADDR	0xE1	jtag_dbg1
DBG2_ADDR	0xE2	jtag_dbg2
DBG3_ADDR	0xE3	jtag_dbg3
DBG0_DATA	0xE4	jtag_dbg0
DBG1_DATA	0xE5	jtag_dbg1
DBG2_DATA	0xE6	jtag_dbg2
DBG3_DATA	0xE7	jtag_dbg3

Внешний интерфейс соответствует стандарту ieee1149.1 и включает в себя пять сигналов: ntrst (активный уровень на входе МП - '0'), tck, tms, tdi, tdo.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

8.2 Канал доступа к памяти (jtag_mem)

Предназначен для доступа к устройствам памяти, находящимся во всем доступном МП диапазоне адресов. Является одним из клиентских каналов коммутатора. Доступ осуществляется через расширенные команды JTAG.

Через jtag_mem возможно одновременно записать или считать 4 последовательно расположенных байта из любого доступного места адресного пространства МП.

Запись/чтение производятся следующим образом:

- перевести контроллер tap в состояние приема команд (в соответствии со стандартом ieee1149.1);
- передать команду MEM_ADDR (она переключит tap на работу с блоком jtag_mem);
- перевести контроллер tap в состояние приема данных;
- передать данные, содержащие адрес и режим работы;
- перевести контроллер tap в состояние приема команд;
- передать команду MEM_DATA;
- перевести контроллер tap в состояние приема данных;
- передать/принять данные;

Структура поля данных, передаваемого после MEM_ADDR

Номер бита	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	RW	RW_MODE		ADDR																																	

34	RW	Выбор режима ('1' - запись, '0' - чтение)
32-33	RW_MODE	Выбор режима записи. 00b - *зарезервировано*, 01b - записать данные с 0-31 бит, 10b - записать только точку останова (33 бит в поле данных для записи), 11b - записать данные и точку останова
0-31	ADDR	Адрес

Структура поля данных, передаваемого после MEM_DATA в режиме записи

Номер бита	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	BP	DATA																															

34	BP	Точка останова ('1' - установлена, '0' - не установлена). Может быть установлена только для данных в диапазоне 0x00000000-0x0004000, для остальных адресов будет проигнорирована
0-31	DATA	Данные для записи

Все права защищены © ОАО «Мультиклет» 2010 — 2015

8.3 Канал доступа к регистрам общего назначение и регистрам системной периферии (jtag_gpr)

Канал доступа к регистрам общего назначение и регистрам системной периферии. Доступ осуществляется через расширенные команды JTAG.

Через jtag_gpr возможно одновременно записать или считать один регистр МП. В каждой клетке присутствует свой канал доступа к регистрам. В МП содержимое регистров может быть разным в разных группах клеток.

Запись/чтение производятся следующим образом(x - номер клетки):

- перевести контроллер tap в состояние приема команд (в соответсвии со стандартом ieee1149.1);
- передать команду CELLx_REGS_ADDR (она переключит tap на работу с блоком jtag_gprx);
- перевести контроллер tap в состояние приема данных;
- передать данные, содержащие адрес и режим работы;
- перевести контроллер tap в состояние приема команд;
- передать команду CELLx_REGS_DATA;
- перевести контроллер tap в состояние приема данных;
- передать/принять данные;

Структура поля данных, передаваемого после CELLx_REGS_ADDR

Номер бита	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	RW	-	ADDR																																

34	RW	Выбор режима ('1' - запись, '0' - чтение)
32-33	—	*зарезервировано, поле не воспринимается системой*
0-31	ADDR	Адрес

Структура поля данных, передаваемого после CELLx_REGS_DATA в режиме записи

Номер бита	63 . . . 0
Описание	DATA

0-63	DATA	Данные для записи

Все права защищены © ОАО «Мультиклет» 2010 — 2015

8.4 Канал доступа к устройствам на шине AMBA (ahb_jtag)

Доступ к любому устройству на шине AMBA. Доступ осуществляется через расширенные команды JTAG. ahb_jtag позволяет осуществлять пакетную передачу данных. При этом существуют следующие ограничения:

- размер пакета не должен превышать 1КБайт;
- чтение/запись осуществляется словами по 32 бита;

Запись/чтение производятся следующим образом:

- перевести контроллер tap в состояние приема команд (в соответсвии со стандартом ieee1149.1);
- передать команду AMBA_ADDR;
- перевести контроллер tap в состояние приема данных;
- передать данные, содержащие адрес и режим работы;
- перевести контроллер tap в состояние приема команд;
- передать команду AMBA_DATA;
- перевести контроллер tap в состояние приема данных;
- передать/принять данные;

Структура поля данных, передаваемого после AMBA_ADDR

Номер бита	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	RW	RW_MODE	ADDR																																

34	RW	Выбор режима (’1’ - запись, ’0’ - чтение)
32-33	RW MODE	Выбор режима записи. 00b - байт, 01b - полуслово (16 бит), 10b - слово (32 бита), 11b - *зарезервировано*
0-31	ADDR	Адрес в диапазоне адресов выделенных периферийной шине

Структура поля данных, передаваемого после AMBA_DATA в режиме записи

Номер бита	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	SEQ	DATA																															

32	SEQ	’1’ - режим пакетной передачи, адрес обращения будет инкрементированное значения переданного после команды AMBA_ADDR адреса или инкрементированное значение адреса, полученного при предыдущем обращении. При выполнении операции чтения, в бите SEQ, принимаемых данных, ’1’ будет означать, что данные валидны, ’0’ - транзакция по шине периферийных устройств не завершена. ’0’ - адрес, к которому обращаются, будет взят из поля ADDR, переданного в данных после команды AMBA_ADDR

Все права защищены © ОАО «Мультиклет» 2010 — 2015

0-31	DATA	данные, порядок байтов big-endian (от старшего к младшему), при размере данный меньше 32 бит, выравнивание к MSB

Все права защищены © ОАО «Мультиклет» 2010 — 2015

8.5 Модуль отладки ПО (jtag_dbg)

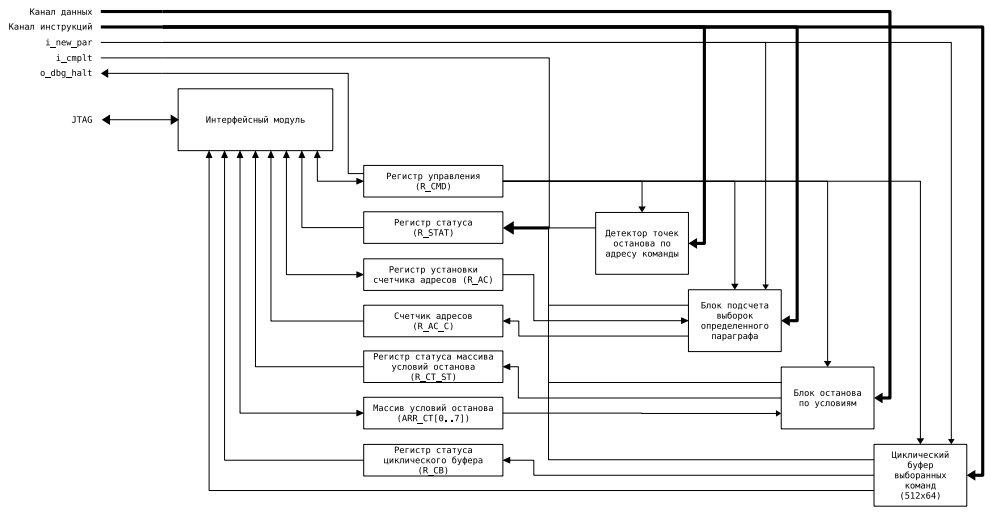

Рис. 12: Блок-схема модуля отладки ПО

Модуль управления клеткой при отладке ПО. Обеспечивает задание аппаратных точек останова по различным параметрам. Доступ осуществляется через расширенные команды JTAG.

Ядро МП содержи 4 модуля jtag_dbg, которые анализируют информацию на шинах данных и инструкций каждого ядра. Модули работают независимо. Каждый модуль вырабатывает сигнал останова системы, эти сигналы по ИЛИ образуют один системный сигнал останова.

Ядро МП останавливается только по концу параграфа, после завершения всех записей в память и регистры.

Запись/чтение производятся следующим образом:

- перевести контроллер tap в состояние приема команд (в соответсвии со стандартом ieee1149.1);
- передать команду DBGx_ADDR (она переключит на работу с конкретным блоком jtag_dbgx);
- перевести контроллер tap в состояние приема данных;
- передать данные, содержащие адрес и режим работы;
- перевести контроллер tap в состояние приема команд;
- передать команду DBGx_DATA;

Все права защищены © ОАО «Мультиклет» 2010 — 2015

- перевести контроллер tap в состояние приема данных;
- передать/принять данные;

Структура поля данных, передаваемого после DBGx_ADDR

	Структура поля данных, передаваемого после DBGx_ADDR		
Номер бита	34	33 ... 32	31 ... 0
Описание	RW	—	ADDR

34	RW	Выбор режима ('1' - запись, '0' - чтение)
32-33	—	*зарезервировано*
0-31	ADDR	Адрес в регистра: 0x01 - R_CMD 0x02 - R_STAT 0x03 - R_AC 0x04 - R_AC_C 0x05 - R_CB_ST 0x06 - R_CT_STAT 0x08-0x10 - таблицы условий останова ARR_CT[x], где $= [0, 7]$ 0x20-0x220 - адреса циклического буфера $[0, 511]$

Структура поля данных, передаваемого после DBGx_DATA

	Структура поля данных, передаваемого после DBGx_DATA	
Номер бита	260	259...0
Описание	—	DATA

260	—	*зарезервировано*
259-0	DATA	содержание зависит от регистра, к которому обращаются. Данные смещены вправо к 0 биту. Регистры описаны ниже.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

jtag_dbg состоит из следующих частей:

- управление и связь;
- детектор точек останова по адресу команды;
- блок подсчета выборок определенного параграфа;
- блок останова по условиям;
- циклический буфер выбранных команд;

Управление и связь

Данная часть представлена интерфейсным модулем JTAG, который отвечает за прием/передачу команд, регистром управления R_CMD и регистром статуса R_STAT. Загрузка/выгрузка информации производится внешней системой по интерфейсу JTAG. Возможно разрешить работу любой из частей jtag_dbg и системы вцелом, принудительно остановить систему. Полная остановка системы определяется по наличию сигнала R_STAT(CMPLT) = '1'. Только после этого можно изменять память МП, регистры и т.д. Иначе возможны коллизии и, как следствие, непредсказуемые действия МП.

R_CMD	Регистр управления								
Номер бита	8	7	6	5	4	3	2	1	0
Описание	EN_AC1_BP	EN_AC0_BP	EN_CB_BP	EN_CT_BP	EN_I_BP	CB_MOD	HALT	CLR_BP	EN

8	EN_AC1_BP	разрешение точки останова по 1-му счетчику адресов команд
7	EN_AC0_BP	разрешение точки останова по 0-му счетчику адресов команд
6	EN_CB_BP	разрешение точки останова по кольцевому буферу команд
5	EN_CT_BP	разрешение точек останова по таблице условий канала данных
4	EN_I_BP	разрешение точек останова по адресу инструкции
3	CB_MOD	режим буфера команд: '1' - фиксация всех адресов команд, '0' - фиксация только начальных адресов параграфов
2	HALT	принудительный останов системы
1	CLR_BP	сброс сработавших точек останова
0	EN	разрешение работы системы отладки

Все права защищены © ОАО «Мультиклет» 2010 — 2015

R_STAT	Регистр статуса					
Номер бита	5	4	3	2	1	0
Описание	AC_BP	CB_BP	CT_BP	I_BP	HALT	CMPLT

5	AC_BP	остановка счетчику инструкции
4	CB_BP	остановка по кольцевому буферу
3	CT_BP	остановка по таблице условий канала данных
2	I_BP	остановка по адресу инструкции
1	HALT	принудительная остановка системы
0	CMPLT	параграф завершен

Детектор точек останова по адресу команды

Детектор получает данные с канала инструкций и позволяет остановить систему при обнаружении признака останова у выбранной команды. Признаки останова можно записать только для команд, расположенных в диапазоне адресов 0x00000000-0x0004000. Сигнал останова возникает тогда, когда выбрана команда с признаком останова.

Блок подсчета выборок определенного параграфа

Блок получает данные с канала инструкций и позволяет подсчитывать кол-во переходов на 2 предустановленных адреса начала параграфа. Сигнал останова возникает тогда, когда количество выборок параграфа совпадет с установленным лимитом.

R_AC	Регистр установки счетчика адресов			
Номер бита	127 … 96	95 … 64	63 … 32	31 … 0
Описание	CNT1	CNT0	ADDR1	ADDR0

96 . . . 127	CNT1	значение 1-го счетчика кол-ва совпадений адреса при котором будет произведен останов
64 . . . 95	CNT0	значение 0-го счетчика кол-ва совпадений адреса при котором будет произведен останов
32 . . . 63	ADDR1	адрес инструкции по которому ведет подсчет 1-й счетчик
0 . . . 31	ADDR0	адрес инструкции по которому ведет подсчет 0-й счетчик

Все права защищены © ОАО «Мультиклет» 2010 — 2015

R_AC_C	Счетчики адресов	
Номер бита	63 … 32	31 … 0
Описание	CNT1	CNT0

32 . . . 63	CNT1	1-й счетчик кол-ва совпадений адреса
0 . . . 31	CNT0	0-й счетчик кол-ва совпадений адреса

Циклический буфер выбранных команд

Буфер получает информацию с канала инструкций. Выполняет функции тарссировщика. Команды фиксируются согласно R_CMD(CB_MOD). В буфер может быть записано до 512 длинных команд (64бита). Сигнал останова возникает тогда, когда буфер переполнился. При этом буфер может продолжать принимать информацию, счетчик переполнений инкрементируется.

R_CB	Состояние циклического буфера команд	
Номер бита	24 … 16	15 … 0
Описание	OVL_CNT	POS_CNT

16 . . . 24	OVL_CNT	текущая позиция в буфере
0 . . . 15	POS_CNT	кол-во переполнений буфера с момента разрешения его работы

Блок останова по условиям

Блок получает информацию с канала данных и, согласно таблице условий, принимается решение о выработке сигнала останова. Всего возможно задать 8 условий останова, учитывающих значение и адрес данных. Сигнал останова возникает тогда, когда выполнилось хотя бы одно из заданных условий.

ARR_CT	Таблица условий						
Номер бита	260 … 197	196 … 133	132 … 69	68 … 37	36 … 5	4 … 1	0
Описание	—	OPRND	MSK	ADDR_E	ADDR_B	OC	RW

197 . . . 260	—	*зарезервировано*
133 . . . 196	OPRND	данные для операции, 64 бита
69 . . . 132	MSK	маска на входные данные, 64 бита, вылоняется операция логического И с данными, маска учитывается при всех операциях ОС в которых анализируется значение данных
37 . . . 68	ADDR_E	конечный адрес блока данных значения которых анализируются, 32 бита
5 . . . 36	ADDR_B	начальный адрес блока данных или адрес конкретной ячейки памяти значения(е) которых(ой) анаизируются, 32 бита
1 . . . 4	OC	код операции, условия останова или функция, которая выполняется над OPRND и входными данными из указанного адреса: 0x1 - останов производится, если на шине данных была произведена операция указанная в WR 0x2 - останов производится, если на шине данных появилось значение равное OPRND 0x3 - останов производится, если на шине данных появилось значение больше OPRND 0x4 - останов производится, если на шине данных появилось значение меньше OPRND 0x5 - останов производится, если логическое И данных и OPRND дало результат ноль

Все права защищены © ОАО «Мультиклет» 2010 — 2015

0	RW	тип операции на шине данных ’1’ - чтение, ’0’ - запись

8.6 Регистр периферийного сканирования (bsr)

Регистр периферийного сканирования поддерживает все обязательные команды по стандарту ieee1149.1.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9 Периферийные устройства

Периферийное устройство	Базовый адрес
ADC0	0xC01D0000
ADC1	0xC01D0100
DAC0	0xC01D1000
DAC1	0xC01D1100
ETHERNET0	0xC0005000
EXTMEM	0xC00EE000
GPIOA	0xC00F0000
GPIOB	0xC00F0100
GPIOC	0xC00F0200
GPIOD	0xC01F0300
GPIOE	0xC01F0400
GPIOF	0xC01F0500
GPTIM0	0xC0010000
GPTIM1	0xC0010100
GPTIM2	0xC0110000
GPTIM3	0xC0110100
I2C0	0xC0001000
I2C1	0xC0001100
I2S0	0xC0002000
PWM0	0xC0112000
RTC	0xC001F000
SPI0	0xC0104000
SPI1	0xC0108000
STAT	0xC01E1000
UART0	0xC0000100
UART1	0xC0000200
UART2	0xC0100100
UART3	0xC0100200
WDT	0xC00E0000
USB	0xFFFF0000

Таблица 16: Базовые адреса периферийных устройств

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.1 Порт ввода-вывода (GPIO)

9.1.1 Краткие характеристики

- МП содержит 1 порт на 32 бита, 2 порта по 30 бит, 1 порт 28 бит, 1 порт на 24 бит и 1 порт на 22 бита;
- каждый бит порта может быть индивидуально настроен на ввод или вывод и может опционально генерировать прерывания;
- запрос прерывания может формироваться по уровню либо по фронту (передний-/задний) сигнала;
- вводы-выводы портов могут переключаться на альтернативную функцию;

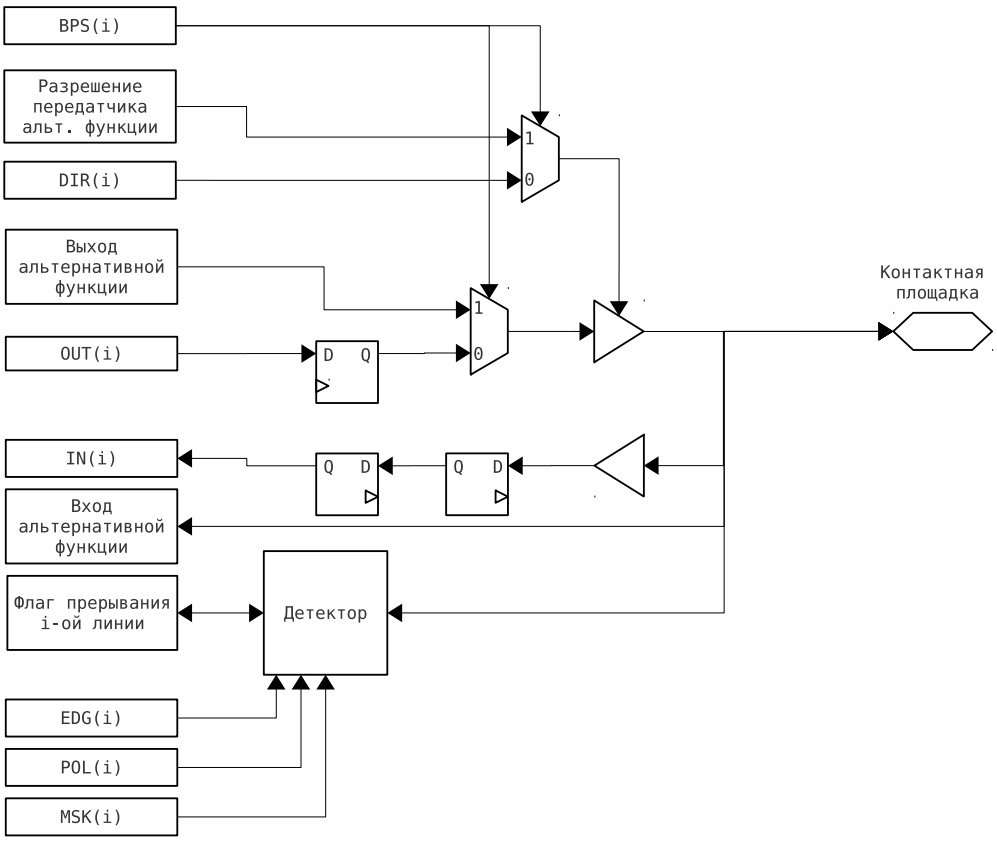

Рис. 13: Блок-схема i-ой линии GPIO

9.1.2 Функционирование GPIO

Порты ввода-вывода реализованы как двунаправленные буферы с программируемым разрешением вывода. Вход каждого буфера синхронизирован с помощью двух последо-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

вательно включенных триггеров, чтобы исключить возможность возникновения метастабильности. Синхронизированное значение может быть прочитано из регистра принимаемых данных (GPIOxIN) порта ввода-вывода. Разрешение вывода управляется через регистр разрешения передачи (GPIOxDIR). Логическая единица ('1') в данном разряде порта ввода вывода конфигурирует соответствующую ей линию на вывод. Выходное значение берется из регистра передаваемых данных (GPIOxOUT) порта ввода-вывода.

Каждая линия порта ввода-вывода может быть сконфигурирована на формирование прерывания. Формирование прерывания управляется тремя регистрами: регистром маски прерываний (GPIOxMSK), регистром настройки прерывания по событию, полярности сигнала (GPIOxPOL) и регистром настройки прерывания по событию, составляющая сигнала (GPIOxEDG). Для разрешения прерывания, соответствующий бит в маске прерываний должен быть установлен ('1'). Если регистр составляющей сигнала сброшен ('0') , то прерывание формируется по уровню сигнала. Если регистр полярности сигнала сброшен ('0'), то прерывание возникает при активном низком уровне сигнала, если же регистр полярности сигнала установлен ('1'), то прерывание возникает при активном высоком уровне сигнала. Если регистр составляющей сигнала установлен ('1'), то прерывание формируется по фронту сигнала. Если регистр полярности сигнала сброшен ('0'), то прерывание возникает по переднему фронту сигнала, если же регистр полярности сигнала установлен ('1'), то прерывание возникает по заднему фронту сигнала.

Каждый вывод порта ввода-вывода может быть совместно использован для других типов сигналов, выполняющих альтернативные функции. Для разрешения задания альтернативных функций какой-либо линией порта необходимо в регистре разрешения альтернативных функций (GPIOxBPS) установить соответствующий бит ('1'). Описание всех вводов-выводов МП смотрите в п.10

Важное замечание: если альтернативная функция битов порта GPIO, задаваемая через BPS - выход, то биты порта конфигурируются в соответствии с направлением BPS. Если альтернативная функция битов порта GPIO, задаваемая через BPS - вход, то биты порта конфигурируются в соответствии с направлением DIR. Таким образом, если установлен бит в DIR в "1 то установка в BPS "1"для текущего бита не позволит ему сконфигурироваться на вход(только на выход). В этом случае необходимо сначала сконфигурировать бит в DIR в "0"и из такого положения с помощью BPS регистра бит конфигурируется на любое направление для альтернативной функции.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.1.3 Описание регистров

Порты ввода-вывода общего назначения:

GPIOA: базовый адрес - 0xC00F 0000; ширина порта - 30 разрядов.
GPIOB: базовый адрес - 0xC00F 0100; ширина порта - 30 разрядов.
GPIOC: базовый адрес - 0xC00F 0200; ширина порта - 24 разряда.
GPIOD: базовый адрес - 0xC01F 0300; ширина порта - 32 разряда.
GPIOE: базовый адрес - 0xC01F 0400; ширина порта - 28 разрядов.
GPIOF: базовый адрес - 0xC01F 0500; ширина порта - 22 разряда.
Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра.

Регистр	Смещение адреса	Доступ	Описание
GPIOxIN	00h	*R*	Регистр принимаемых данных.
GPIOxOUT	04h	*RW*	Регистр передаваемых данных.
GPIOxDIR	08h	*RW*	Регистр разрешения передачи.
GPIOxMSK	0Ch	*RW*	Регистр маски прерываний.
GPIOxPOL	10h	*RW*	Регистр настройки прерывания по событию, полярность сигнала.
GPIOxEDG	14h	*RW*	Регистр настройки прерывания по событию, составляющая сигнала.
GPIOxBPS	18h	*RW*	Регистр разрешения альтернативных функций

GPIOxIN	**Регистр принимаемых данных**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр принимаемых данных. Каждый разряд регистра соответствует каждой линии порта.

GPIOxOUT	**Регистр передаваемых данных**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр передаваемых данных. Каждый разряд регистра соответствует каждой линии порта.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

GPIOxDIR	Регистр разрешения передачи																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр разрешения передачи. Каждый разряд регистра соответствует каждой линии порта. Если любой разряд регистра установлен в '1', то у соответствующей линии порта передача данных разрешена, если же разряд установлен в '0'- передача запрещена.

GPIOxMSK	Регистр маски прерываний																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр маски прерываний. Каждый разряд регистра соответствует каждой линии порта. Если любой разряд регистра установлен в '1', то у соответствующей линии порта разрешено прерывание по событию, если же разряд установлен в '0'- прерывание по событию запрещено.

GPIOxPOL	Регистр настройки прерывания по событию, полярность сигнала																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр настройки прерывания по событию, полярность сигнала. Каждый разряд регистра соответствует каждой линии порта. Если любой разряд регистра установлен в '1', то соответствующая линия порта формирует прерывание по переднему фронту/высокому уровню, если же разряд установлен в '0'- формирует прерывание по заднему фронту/низкому уровню. Выбор фронта или уровня зависит от настройки соответствующего бита в регистре GPIOxEDG.

GPIOxEDG	Регистр настройки прерывания по событию, составляющая сигнала																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр настройки прерывания по событию, составляющая сигнала. Каждый разряд регистра соответствует каждой линии порта. Если любой разряд регистра установлен в '1', то соответствующая линия порта формирует прерывание по фронту, если же разряд установлен в '0'- формирует прерывание уровню.

GPIOxBPS	Регистр разрешения альтернативных функций																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Описание	DATA																															

0-31 DATA Регистр разрешения альтернативных функций. Каждый разряд регистра соответствует каждой линии порта. Если любой разряд регистра установлен в '1', то у соответствующей линии порта выполнение альтернативной функции разрешено, если же разряд установлен в '0'- выполнение альтернативной функции запрещено.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2 Контроллер шины внешней памяти (MCTRL0)

9.2.1 Краткие характеристики

- поддерживает работу с памятью типа: PROM, SRAM, SDRAM;
- автоматическая регенерация SDRAM;
- внешняя шина данных до 32 бит;
- работа с устройствами ввода/вывода (I/O);

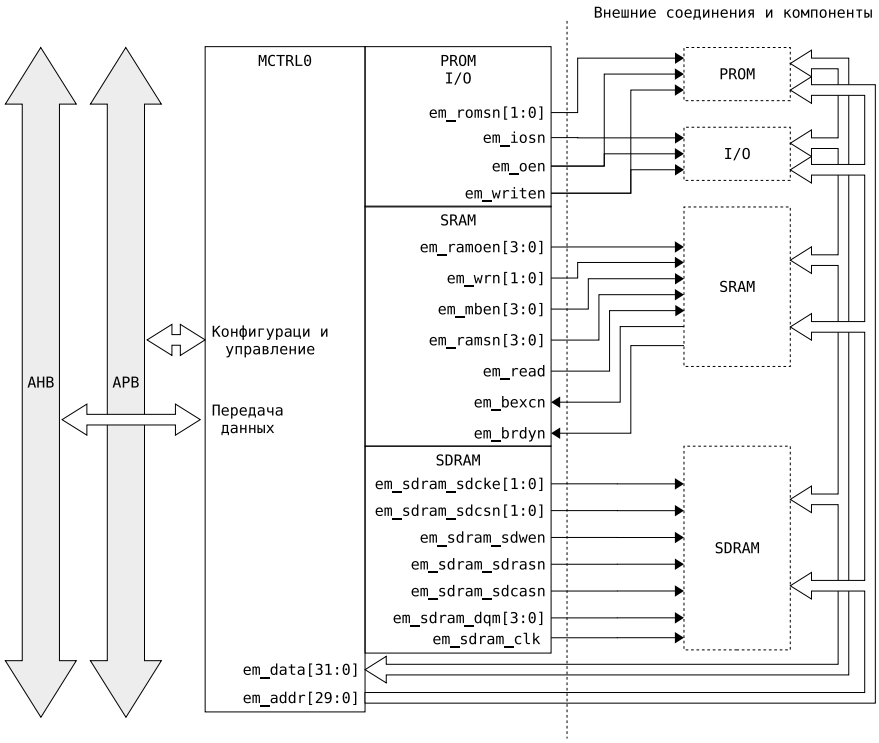

Рис. 14: Контроллер шины внешней памяти

Контроллер может управлять 2-мя банками PROM, одним банком I/O, 4-мя банками SRAM и двумя банками SDRAM.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.2 Интерфейс PROM

Временные диаграммы работы с PROM памятью похожи на диаграммы работы с RAM память. Отличие состоит в том, что циклы работы PROM могут иметь до 15 циклов задержки.

Ниже приведены диаграммы работы интерфейса.

Диаграммы чтения:

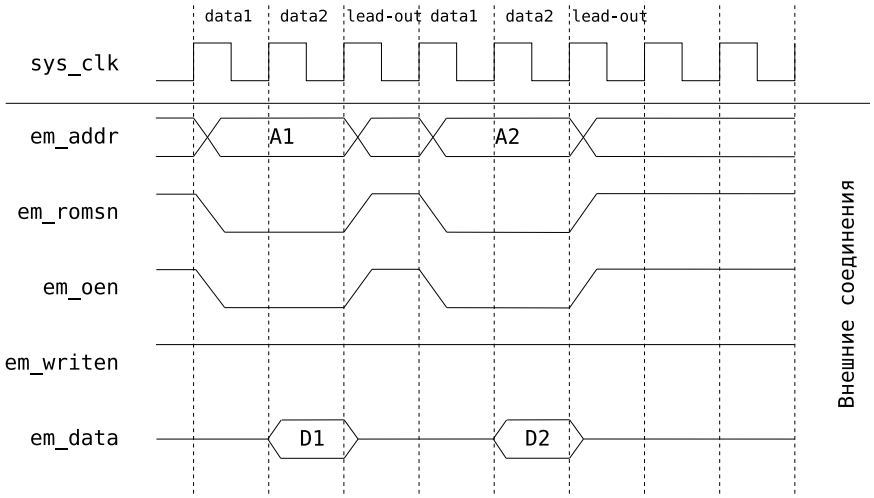

Рис. 15: Диаграмма непоследовательного чтения PROM

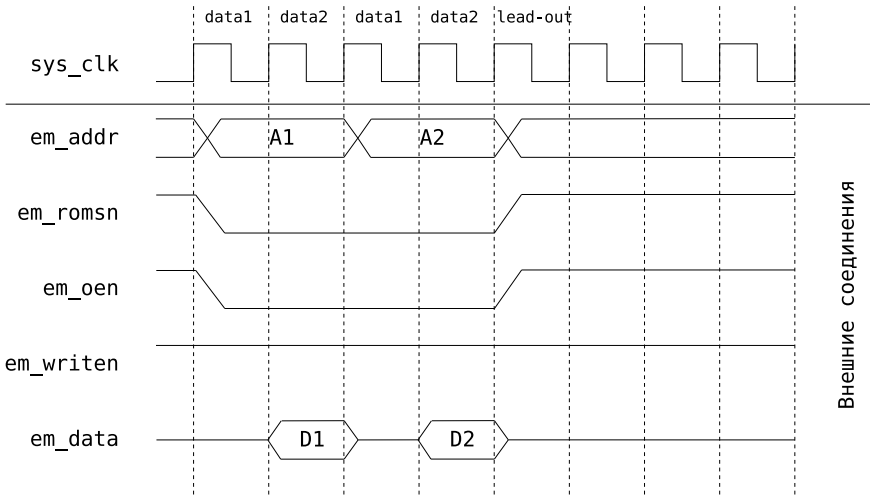

Рис. 16: Диаграмма последовательного чтения PROM

Все права защищены © ОАО «Мультиклет» 2010 — 2015

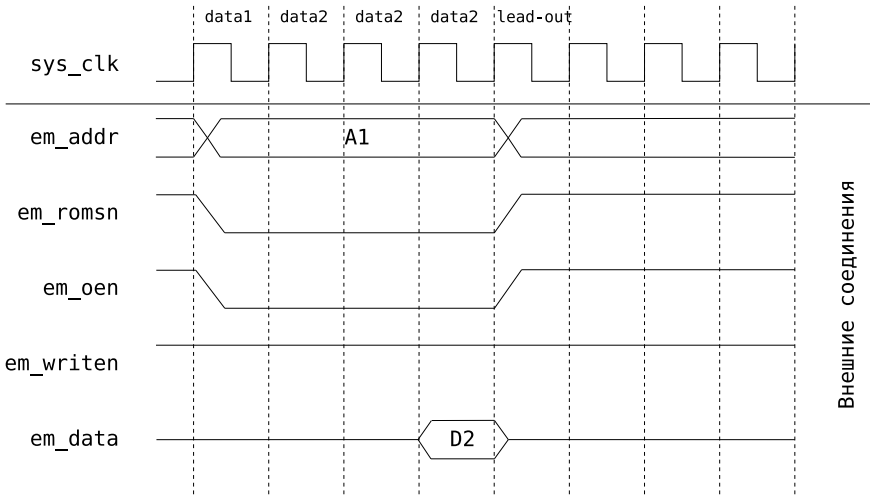

Рис. 17: Диаграмма чтения PROM с 2-мя циклами задержки

Диграммы записи:

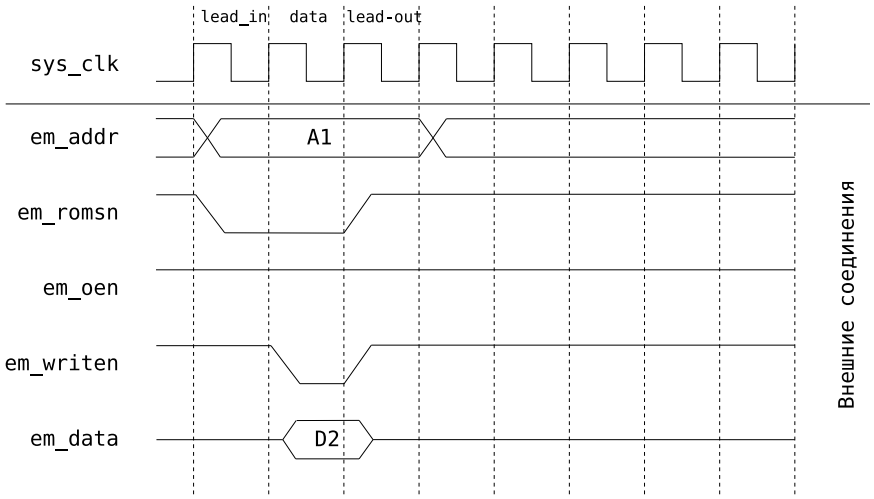

Рис. 18: Диаграмма записи PROM (без циклов задержки)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

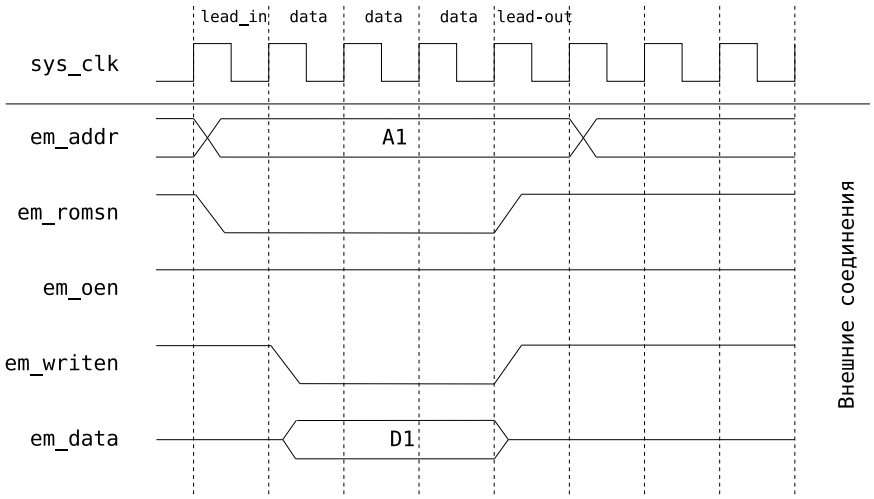

Рис. 19: Диаграмма записи PROM (2 цикла задержки)

В зависимости от того в какую половину (верхнюю или нижнюю) области памяти PROM происходит обращение задействуются выводы ROMSN[1:0]. Если обращение происходит в нижнюю половину адресов, то задействуется вывод ROMSN[0]. Если обращение происходит в верхнюю половину адресов, то задействуется вывод ROMSN[1].

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.3 Интерфейс I/O

Временные диаграммы работы с I/O памятью похожи на диаграммы работы с RAM/ROM памятью. Отличие состоит в том, что дополнительные циклы задержки могут быть добавлены путем перевода состояния вывода BRDYN в логический "0". Сигнал выбора IOSN интерфейса памяти I/O задерживается на 1 такт для обеспечения стабильности адреса до перехода сигнала IOSN в логическую "1".

Ниже приведены диаграммы работы интерфейса.

Диаграммы чтения:

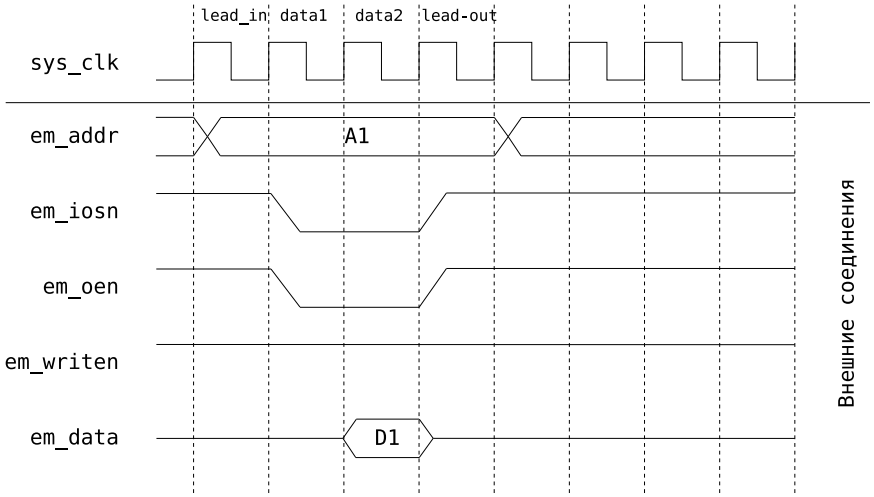

Рис. 20: Диаграмма чтения I/O памяти (без циклов задержки)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Диграммы записи:

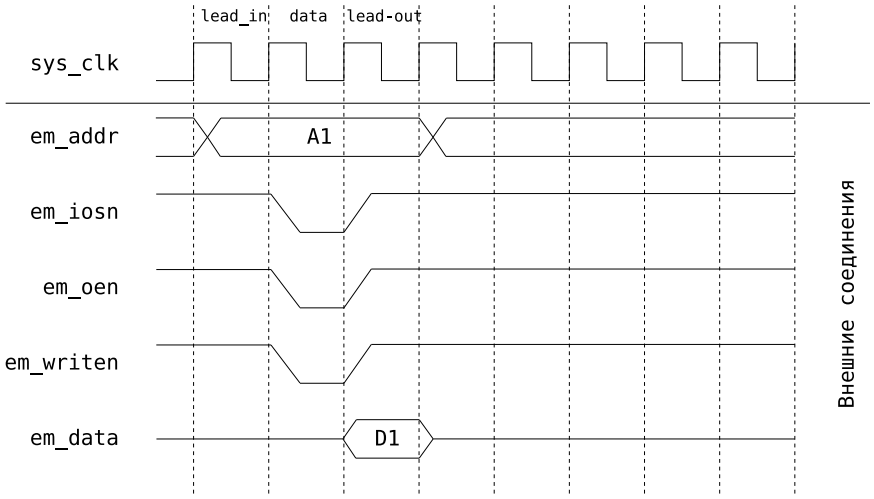

Рис. 21: Диаграмма записи I/O памяти (без циклов задержки)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.4 Интерфейс SRAM

Область памяти SRAM можеть иметь максимальный размер 256 Мегабайт. Из них младшие 256 Мегабайт могут занимать до 4-х банков SRAM, старшую половину занимает SDRAM, если она включена. Но есть бит MCFG2(SI), и если этот бит установлен в '1', то в младших 256 Мегабайтах области размещения RAM памяти размещается SDRAM. Размер банков SRAM программируется двоичными шагами от 8 Кбайт до 256 Мегабайт.

Ниже приведены иллюстрации размещения SRAM и SDRAM памяти:

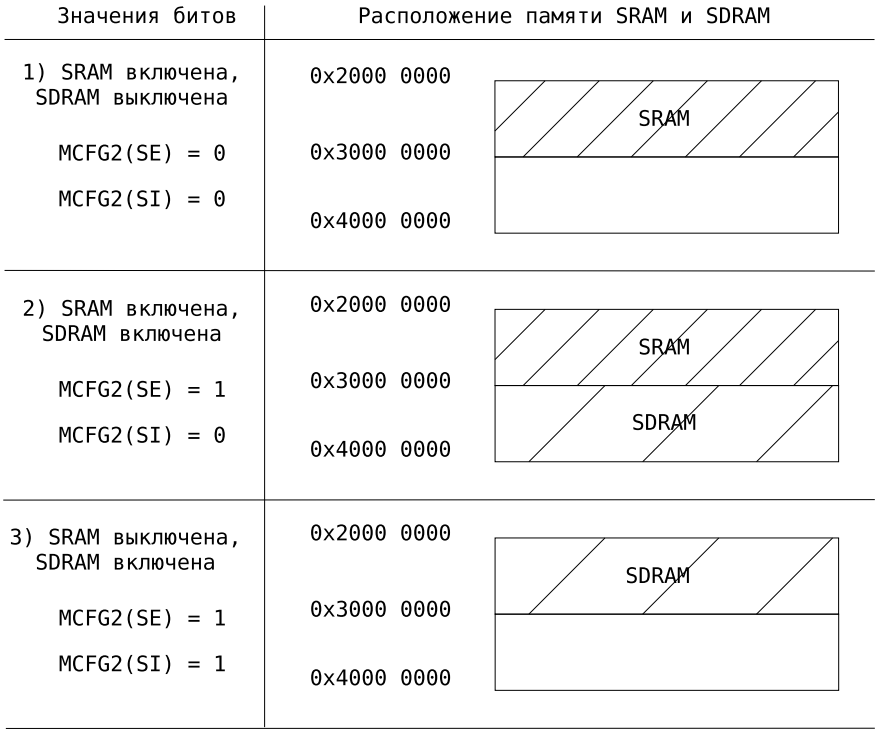

Рис. 22: Взаимное размещение SRAM и SDRAM памяти в RAM области карты памяти

Чтение из памяти SRAM состоит из двух циклов чтения данных и задержками от 0 до 3 (устанавливаются пользователем). При непоследовательном чтении добавляется lead-out цикл для предотвращения несогласованности на шине из-за медленного переключения памяти или I/O устройств.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Ниже приведены диаграммы работы интерфейса.

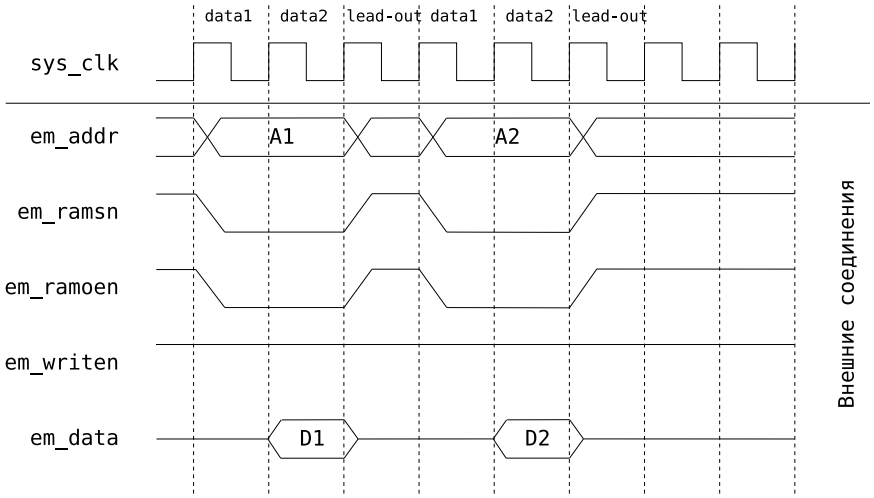

Рис. 23: Диаграмма непоследовательного чтения SRAM памяти

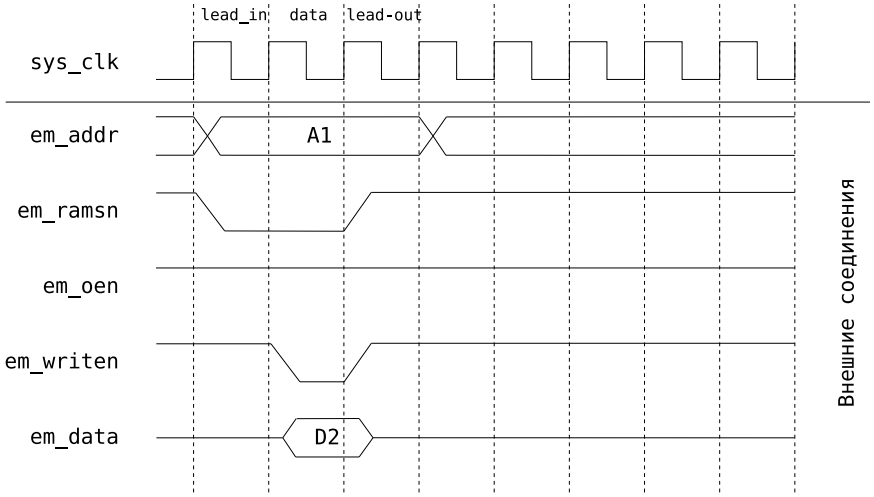

Рис. 24: Диаграмма записи SRAM памяти (без циклов задержки)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.5 Интерфейс SDRAM

Размещение памяти SDRAM в области RAM карты памяти процессора происходит в соответствии с рис. 22. Доступно 2 банка памяти SDRAM с максимальным размером банка в 256 Мегабайт, начиная от 4 Мегабайт с двоичным шагом и позволяет работать с 8-12 битным адресом столбца и 13 битным адресом строки. SDRAM контроллер работает только в 32-х битном формате по шине данных. Контроллер поддерживает mobile SDRAM.

Инициализация

Когда включен SDRAM контроллер, то прозводятся автоматически инициализационные последовательности PRECHARGE, 2x AUTO-REFRESH, LOAD-MODE-REG во всех банках одновременно. Когда включено функционирование mobile SDRAM, то инициализационная последовательность прикрепляется по команде LOAD-EXTMODE-REG. Контроллер программирует SDRAM на последовательное чтение страниц памяти и одиночную локальную запись.

Конфигурация временных параметров

Для обеспечения оптимального взаимодействия с различными SDRAM устройствами некоторые параметры могут быть установлены пользователем в регистре MCFG2.
Ниже представлена таблица возможных параметров:

Функция	Параметр	Диапазон	Размерность
RAS/CAS задержка	Tcas, Trcd	2-3	такт
Предварительная зарядка для включения	Trp	2-3	такт
Период команды автообновления	Trfc	3-11	такт
Период автообновления	Trf	10-32768	такт

В случае включения поддержки mobile SDRAM появляется дополнительный параметр Txsr для настройки в регистре MCFG4.
Txsr задает задержку между окончанием режима самообновления и началом команды.

Обновление

Контроллер SDRAM содержит функцию обновления, которая периодически посылает команду AUTO-REFRESH в каждый банк SDRAM. Период между командами обновления записывается в тактах в регистре MCFG3. В зависимости от типа SDRAM необходимый период обычно от 7,8 до 15,6 мкс (соответствует 780 и 1560 тактам на частоте 100 МГц). Период обновления формируется по формуле:

$$Trefresh = (SDRRV + 1)/Tsysclk$$

Функция обновления включается установкой MCFG2(31) в '1'.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Самообновление

"Режим самообновления"используется для сохранения данных в SDRAM памяти, когда остальная система отключена. В этом случае SDRAM сохраняет данные без внешнего тактирования и обновление происходит внутри нее. Область памяти, которую необходимо обновлять в ходе операции самообновления определяется в расширенном регистре режима. Эти настройки могут быть изменены путем установки значений в MCFG4(PASR). Расширенный регистр режима памяти SDRAM автоматически обновится при изменении MCFG4(PASR). Поддерживаются следующие режимы частичного самообновления: полный, наполовину, одна четвертая, одна восьмая, одна шестнадцатая области памяти. "Режим частичного самообновления"поддерживается только при разрешенном функционировании mobile SDRAM. Для разрешения функционирования режима самообновления установите MCFG4(PMODE) в '010'. Контроллер будет в режиме самообновления после каждого доступа к памяти (когда контроллер в режиме ожидания в течение 16 тактов), пока не будет установлен MCFG4(PMODE) в '0'. При выходе из данного режима контроллер произведет задержку на период Txsr и команду AUTO-REFRESH до того как любой доступ к памяти будет разрешен. Минимальная длительность данного режима определяется Tras.

Режим отключения

Когда установлен "Режим отключения"все входные и выходные буферы, за исключением SDCKE, отключаются. Все данные SDRAM сохраняются в течение данной операции. Для включения "режима отключения"установите MCFG4(PMODE) в '001'. Контроллер будет выходить в "режим отключения"после каждого доступа к памяти(когда контроллер в режиме ожидания в течение 16 тактов), пока не будет установлен MCFG4(PMODE) в '0'. Команда REFRESH еще будет выдана в этом режиме. При выходе из данного режима задержка в один такт будет выдана до любого команды к памяти.

Режим глубокого отключения

"Режим глубокого отключения"используется для достижения максимальной экономии энергии за счет отключения питания области памяти. Данные SDRAM не будут сохранены при переходе в данный режим. Для перехода в данный режим установите MCFG4(PMODE) в '101'. Для выхода из данного режима установите MCFG4(PMODE) в '0'. На попытки доступа к памяти контроллер будет выдавать ERROR на шину AMBA в данном режиме.

Температурная компенсация режима самообновления

Температурная компенсация частоты регенерации памяти устанавливается в MCFG4(TCSR). Расширенный регистр режимов автоматически обновится при изменении битов MCFG4(TCSR).

Команды SDRAM

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Контроллер SDRAM выдавать 4 команды, записаннхы в поле MCFG2(SDRAM CMD): PRE-CHARGE, AUTO-REFRESH, LOAD-MOD-REG(LMR), LOAD-EXTMODE-REG(EMR). Если выполнена команда LMR, то CAS задержка будет установлена как запрограммировано в MCFG2, остальные поля также будут исправлены: последовательное очередное чтение страниц, одиночная локальная запись, последовательная очередь. Если выполнена команда EMR, то будут использоваться параметры установленные в MCFG4(DS, TCSR, PASR). Для выполнения команды EMR бит MCFG4(EMR) должен быть установлен в '1'. Поле MCFG2(SDRAM CMD) будет очищено после выполнения команды. При изменении CAS задержки необходимо выставлять команду LOAD-MODE-REGISTER.

Циклы чтения

Цикл чтения начинается после выполнения команды ACTIVATE для желаемого банка и строки, следующей за командой READ после CAS задержки. Режим чтения пачками осуществляется при приходе соответсвующего запроса на шине AHB.

Между циклами чтения не висят открытые банки.

Циклы записи

Циклы записи аналогичны циклам чтения, только для активации записи необходима команда WRITE. Запись пачками формируется по соответствующему запросу на шине AHB без циклов простоя.

Подключение шины адреса

Шина адреса подключается выводами A[14:2], для адресации банков используются выводы A[16:15].

Шина данных

Шина данных состоит из 32 линий, которые подключаются к линиям данных внешней памяти SDRAM. SDRAM память должна иметь 32 линии данных, также SDRAM шина данных в 32 линии может быть составлена из двух устройств памяти с шиной данных в 16 бит.

Тактирование

Для синхронизации SDRAM можно использовать сигнал системной тактовой частоты доступный на ножке SYS_CLK.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.6 Использование сигнализации готовности шины

Сигнал готовности шины памяти BRDYN используется для памяти PROM, I/O и сигнализирует о том, что память(либо устройство ввода-вывода) завершила выполнение запрашиваемой операции. Доступ к памяти всегда будет иметь по крайней мере установленные циклы задержки в регистрах MCFG1 и MCFG2, но если исполнение операции задерживается и память снимает в '0' сигнал BRDYN, то цикл доступа будет растянут до момента установки BRDYN в '1'. Ниже представлены временные диаграммы работы:

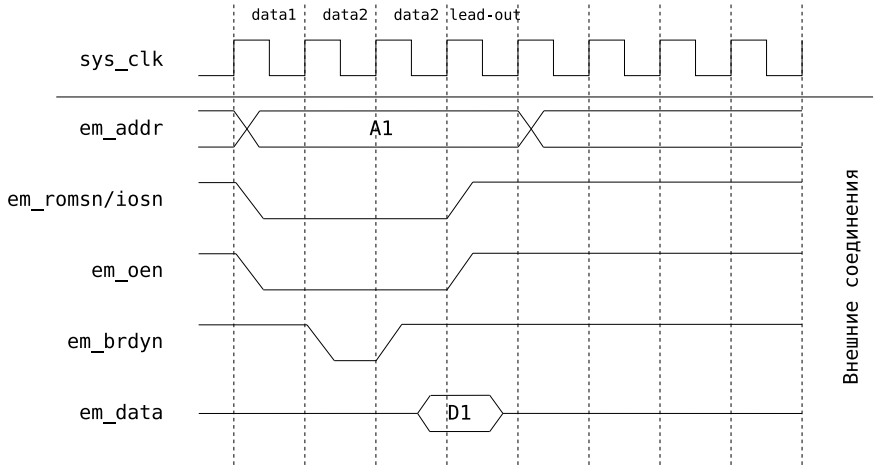

Рис. 25: Диаграмма чтения PROM, I/O с одним дополнительным циклом data2(цикл lead-out только для I/O)

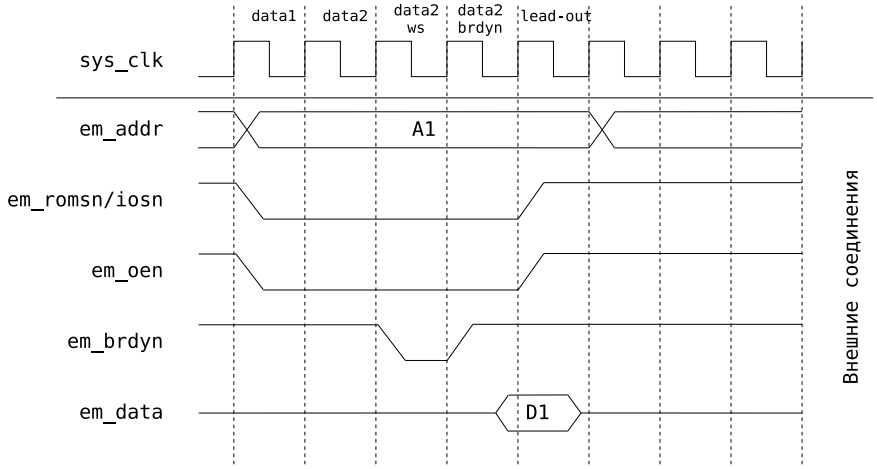

Рис. 26: Диаграмма чтения PROM, I/O с одним предустановленным циклом задержки и одним циклом задержки сформированным BRDYN

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.7 Ошибки доступа

Ошибка доступа может быть просигнализирована выставлением сигнала BEXCN, который устанавливается вместе с данными. Если использование сигнала BEXCN установкой MCFG1(BEXCN) в '1', то ошибки в ответ будут формироваться на шине AMBA. BEXCN активируется для всех областей внешней памяти(PROM, I/O, RAM). Ниже представлены временные диаграммы работы:

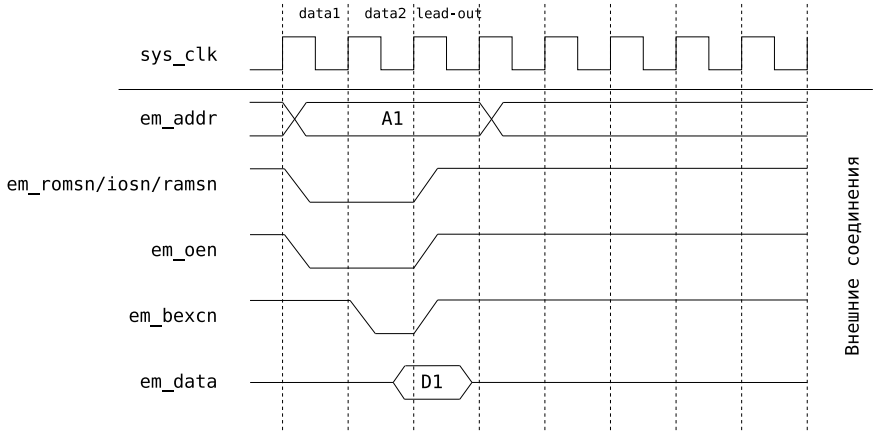

Рис. 27: Диаграмма чтения PROM, I/O, RAM с BEXCN

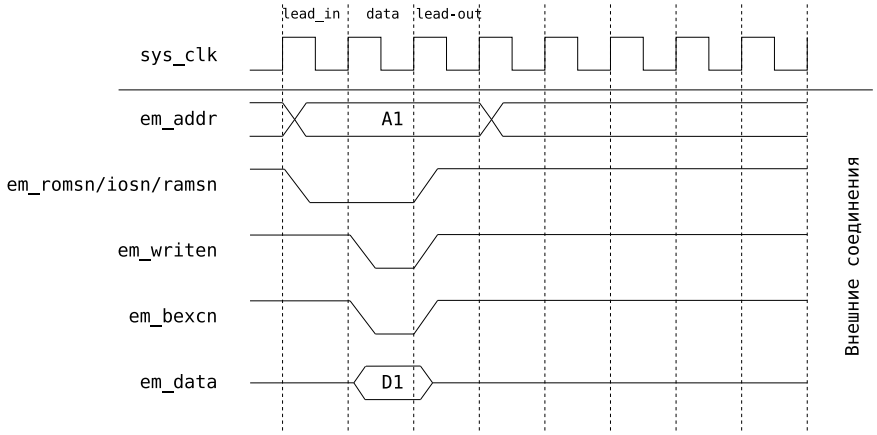

Рис. 28: Диаграмма записи PROM, I/O, RAM с BEXCN (IOSN цикл не устанавливается в lead-in цикле для I/O памяти)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.8 Подключение PROM, SRAM, SDRAM памяти различной разрядности

Ниже представлены схемы подключения внешней памяти:

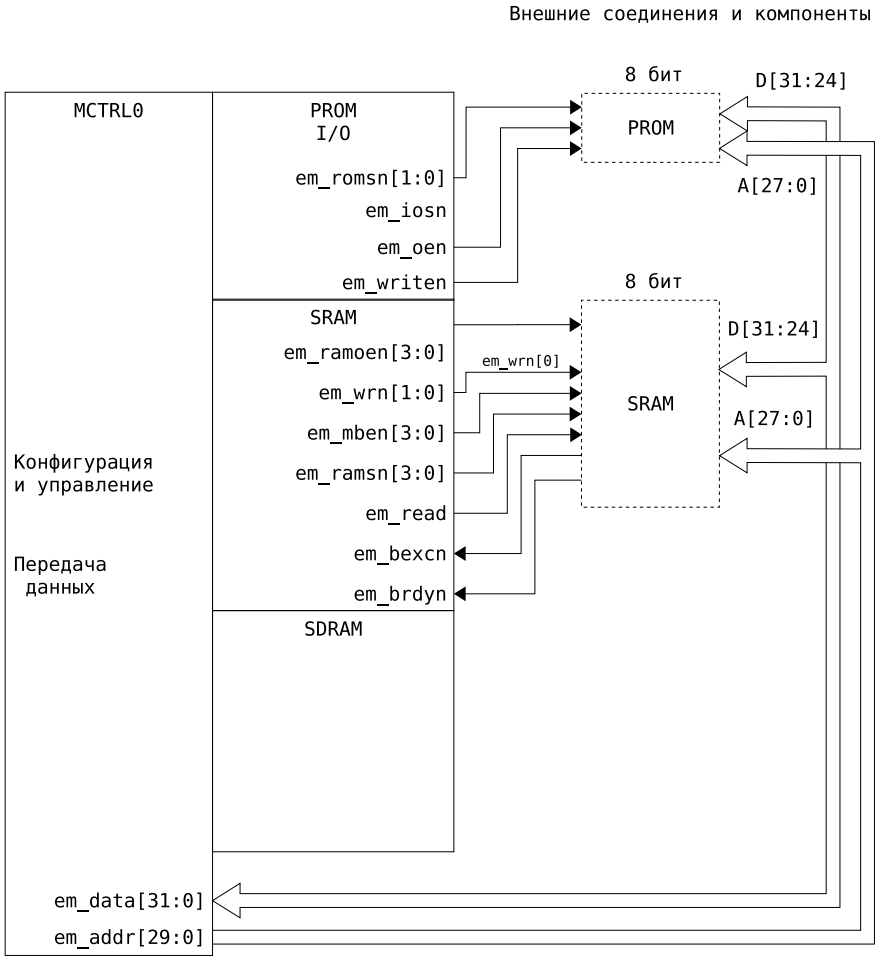

Рис. 29: Схема подключения PROM, SRAM в 8-ми битном формате

Все права защищены © ОАО «Мультиклет» 2010 — 2015

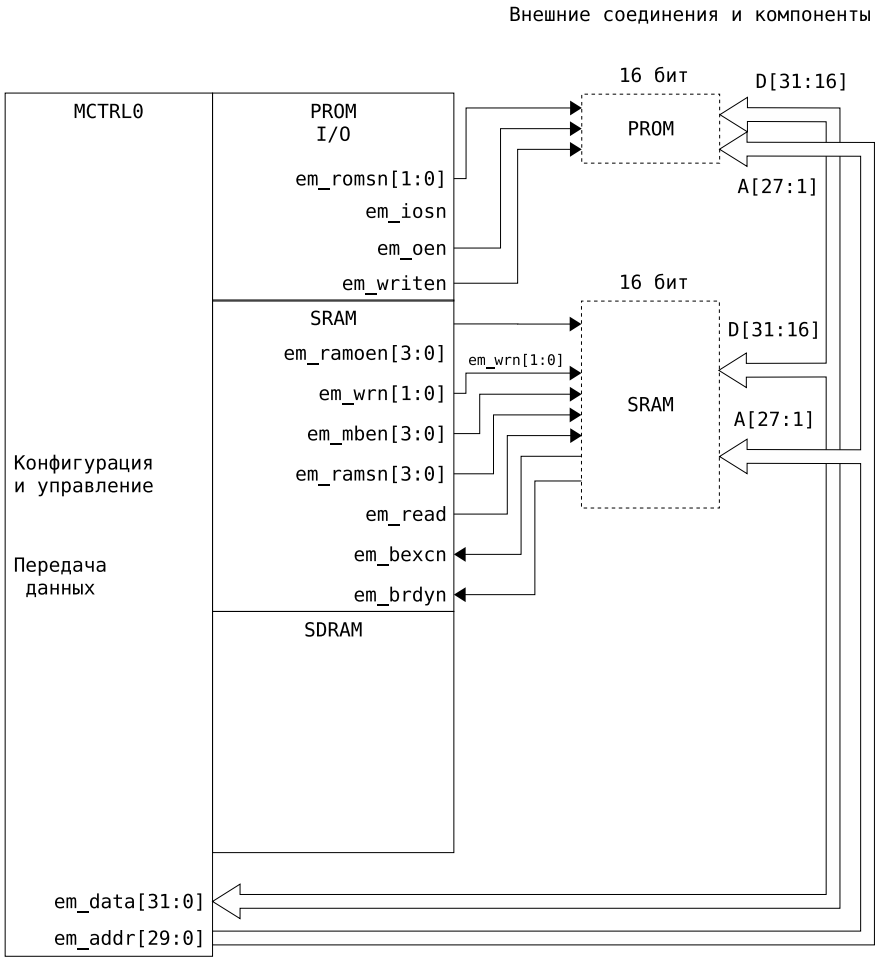

Рис. 30: Схема подключения PROM, SRAM в 16-ти битном формате

Все права защищены © ОАО «Мультиклет» 2010 — 2015

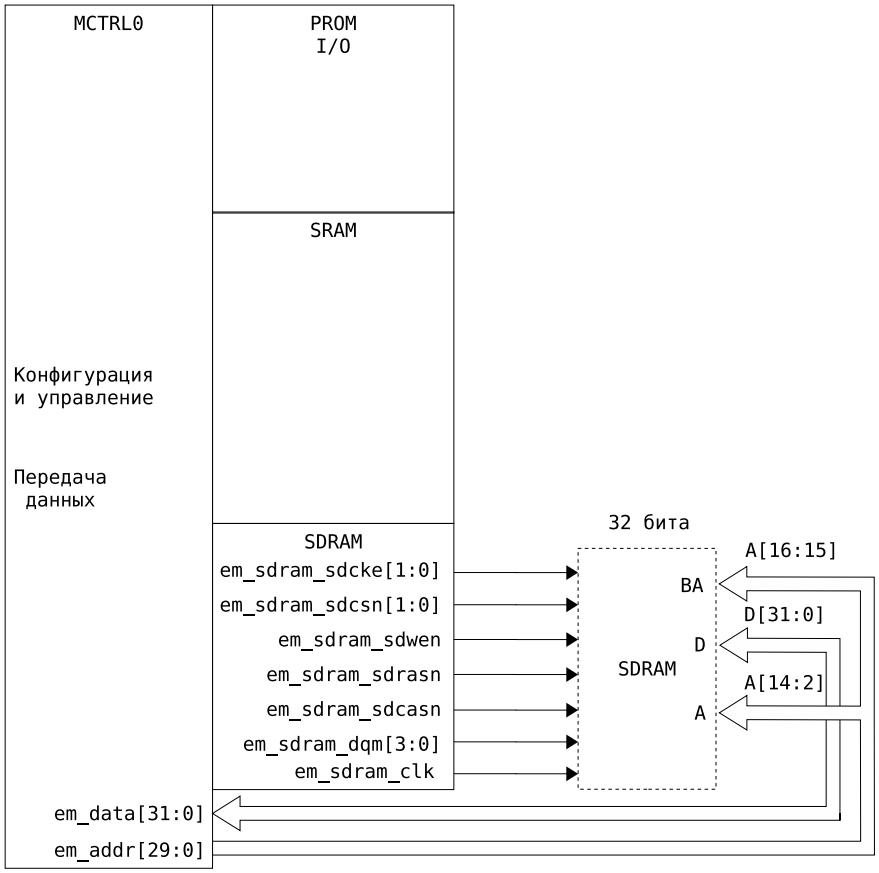

Рис. 31: Схема подключения SDRAM

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.2.9 Описание регистров

Базовый адрес MCFG - 0xC00E E000

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
MCFG1	00h	RW	Регистр конфигурации памяти PROM, IO памяти
MCFG2	04h	RW	Регистр конфигурации памяти SRAM, SDRAM памяти
MCFG3	08h	RW	Период обновления памяти SDRAM
MCFG4	0ch	RW	Регистр управления питанием внешней памяти

MCFG1	Регистр конфигурации памяти PROM, IO памяти																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—			-		0	0	-	1111				-	—							-	-	—		1111				1111			
Описание	—			IOBUSW		IBRDY	BEXCN	-	IO WAITSTATES				IOEN	—							PWEN	-	PROM WIDTH		PROM WRITE WS				PROM READ WS			

29-31	—	*зарезервировано.*
27-28	IOBUSW	Установка разрядности IO памяти (’00’ - 8, ’01’ - 16, ’10’ - 32).
26	IBRDY	Разрешение сигнализации готовности шины для IO памяти(BRDYN) (’0’– запрещено, ’1’– разрешено).
25	BEXCN	Разрешение сигнализации ошибок на шине (’0’– запрещено, ’1’– разрешено).
24	—	*зарезервировано.*
20-23	IO WAITSTATES	Установка количества задержек для работы с памятью IO (’0000’ - 0, ’0001’ - 1, ’0010’ - 2 ... ’1111’ - 15).
19	IOEN	Разрешение доступа к шине памяти IO (’0’– запрещено, ’1’– разрешено).
12-18	—	*зарезервировано.*
11	PWEN	Разрешение записи в память PROM (’0’– запрещено, ’1’– разрешено).
10	—	*зарезервировано.*
8-9	PROM WIDTH	Установка разрядности памяти PROM (’00’ - 8, ’01’ - 16, ’10’ - 32). Начальное значение при подаче питания устанавливается в зависимости от состояния вывода процессора PROM_WID (’0’ - 8, ’1’ - 16).
4-7	PROM WRITE WS	Установка количества задержек для циклов записи памяти PROM (’0000’ - 0, ’0001’ - 1, ’0010’ - 2 ... ’1111’ - 15).
0-3	PROM READ WS	Установка количества задержек для циклов чтения памяти PROM (’0000’ - 0, ’0001’ - 1, ’0010’ - 2 ... ’1111’ - 15).

MCFG2	Регистр конфигурации памяти SRAM, SDRAM памяти																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	-	-	—			-	—			—		—		0	-	-	-	-	-	—				-	-	-	—		—		
Описание	SDRF	TRP	SDRAM TRFC			TCAS	SDRAM BANKSZ			SDRAM COLSZ		SDRAM CMD		D64	-	MS	-	SE	SI	RAM BANK SIZE				-	RBRDY	RMW	RAM WIDTH		RAM WRITE WS		RAM READ WS	

31	SDRF	Разрешение обновления SDRAM (’0’– запрещено, ’1’– разрешено) .
30	TRP	Установка задержки Trp (’0’– 2 такта, ’1’– 3 такта) .
27-29	SDRAM TRFC	Установка задержки Trfc, которая равна (3+значение данного поля) тактов.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

26	TCAS	Установка циклов задержки CAS ('0'– 2 цикла, '1'– 1 цикл). После выбора команды LOAD COMMAND REGISTER будет выполнено в то же время. Также устанавливается RAS/CAS задержка (Trcd).
23-25	SDRAM_BANKSZ	Установка размера банка SDRAM памяти ('000' - 4 Мегабайт, '001' - 8 Мбайт, '010' - 16 Мбайт ... '111' - 512 Мбайт).
21-22	SDRAM_COLSZ	Установка размера столбца SDRAM памяти ('00' - 256 байт, '01' - 512, '10' - 1024, '11' - 4096(если MCFG2(23-25)='111') иначе 2048).
19-20	SDRAM CMD	Установка команды SDRAM ('01' - PRECHARGE, '10' - AUTO REFRESH, '11' - LOAD COMMAND REGISTER). Бит сбрасывается в '0' после выполнения команды.
18	D64	Конфигурация контроллера памяти SDRAM ('0'– 32 бита, '1'– 64 бита).
17	—	*зарезервировано.*
16	MS	Поддержка режима Mobile SDR ('0'– запрещено, '1'– разрешено). Бит доступен только для чтения.
15	—	*зарезервировано.*
14	SE	Разерешение работы SDRAM контроллера ('0'– запрещено, '1'– разрешено).
13	SI	Запрещение работы SRAM если MCFG2(14) = 1.
9-12	RAM BANK SIZE	Установка размера для каждого RAM банка('0000' - 8 Килобайт, '0001' - 16 Килобайт, ... '1111' - 256 Мегабайт).
8	—	*зарезервировано.*
7	RBRDY	Разрешение сигнализации готовности шины для RAM памяти.
6	RMW	Разрешение циклов чтение-модификация-запись для записи полуслов по 16 бит в 32-х битную область с поддержкой строба записи (без строба записи байта).
4-5	RAM WIDTH	Установка разрядности памяти RAM ('00' - 8, '01' - 16, '10' - 32).
2-3	RAM WRITE WS	Установка количества задержек для циклов записи памяти RAM ('00' - 0, '01' - 1, '10' - 2, '11' - 3).
0-1	RAM READ WS	Установка количества задержек для циклов чтения памяти RAM ('00' - 0, '01' - 1, '10' - 2, '11' - 3).

MCFG3	**Период обновления памяти SDRAM**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—					—															—											
Описание	—					SDRAM REFRESH RELOAD VALUE															—											

27-31	—	*зарезервировано.*
12-26	SDRRV	Период обновления SDRAM.
0-11	—	*зарезервировано.*

Период между командами AUTO-REFRESH формируется по следующей формуле:

$$Trefresh = (SDRRV + 1)/Tsysclk$$

MCFG4	**Регистр управления питанием внешней памяти**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—					—															—											
Описание	—					SDRAM REFRESH RELOAD VALUE															—											

31	ME	Установка функционального режима SDRAM ('0'– Standard SDRAM, '1'– Mobile SDRAM).
30	CE	Разрешение тактирования SDRAM ('0'– запрещено, '1'– разрешено). Данный бит доступен только для чтения в случае, если MCFG4(PMODE) не равен '00'.
29	EMR	Если данный бит равен '1', то команда LOAD-COMMAND-REGISTER, выполняемая в регистре MCFG2(SDRAM CMD) будет интерпретирована как LOAD EXTENDED COMMAND REGISTER.
24-28	—	*зарезервировано.*
20-23	TXSR	Установка задержки Txsr, которая будет соответствовать значению данного поля в системных тактах.
19	—	*зарезервировано.*
16-18	PMODE	Установка режима управления питанием:

Все права защищены © ОАО «Мультиклет» 2010 — 2015

		’00’ - нет ’01’ - снижение энергопотребления (данные сохраняются, но нет доступа к памяти) ’10’ - самообновление ’11’ - выключение энергопотребления (данные не сохраняются)
7-15	—	*зарезервировано.*
5-6	DS	Установка нагрузочной способности выходных каскадов: ’00’ - полная ’01’ - наполовину ’10’ - одна четверть ’11’ - три четверти
3-4	TCSR	Резерв для температурной компенсации самообновления: ’00’ - 70°C ’01’ - 45°C ’10’ - 15°C ’11’ - 85°C
0-2	PASR	Установка частичного самообновления SDRAM: ’000’ - полное (банки 0,1,2,3) ’001’ - наполовину (банки 0,1) ’010’ - на четверть (банк 0) ’101’ - на одну восьмую (половина банка 0 с MSB = 0) ’110’ - на одну шестнадцатую (четверть банка 0 с MSB = 00)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.3 Интерфейс UART(UARTx)

9.3.1 Краткие характеристики

- полнодуплексный режим;
- отдельные буферы FIFO глубиной 32 байта для приема и передачи;
- слово данных - 8 бит, фиксированное;
- настраиваемый контроль четности;
- 1 стоп-бит;
- аппаратный контроль потока данных (CTS, RTS);

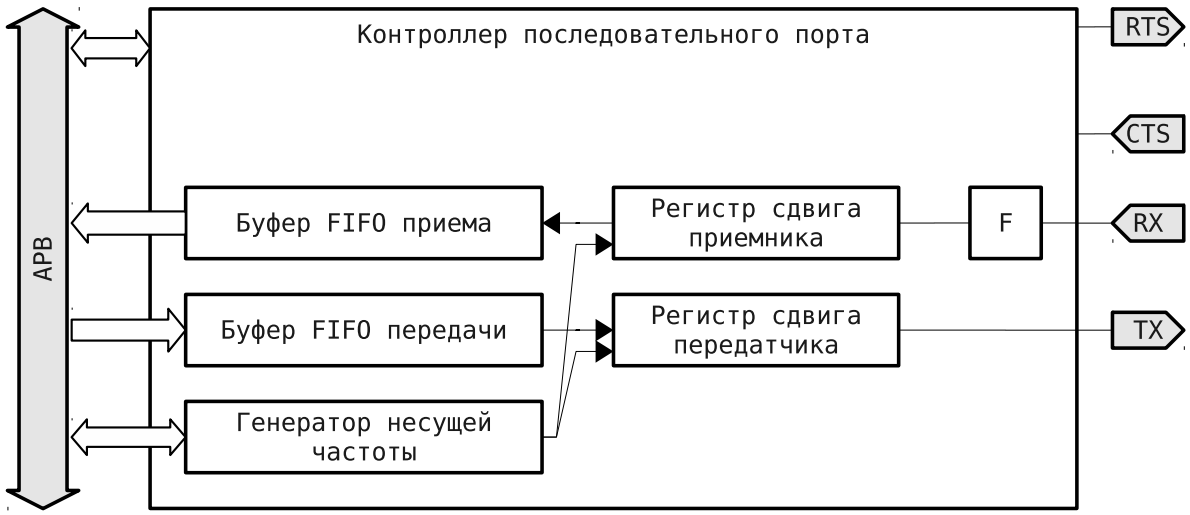

Рис. 32: Блок-схема UARTx

На рис.32 приведена блок-схема контроллера последовательного интерфейса UART.

Контроллер UART автоматически формирует бит контроля четности при передаче и производит контроль четности при приеме данных. Включение и отключение режима контроля четности и его настройка производится в регистре UARTxCR.

9.3.2 Передача данных

Для включения передатчика и разрешения его работы предназначен бит UARTxCR(TE). Данные для передачи записываются в буфер FIFO передачи. Пользователю доступен

Все права защищены © ОАО «Мультиклет» 2010 — 2015

только вход FIFO. Обращение к нему происходит через UARTxDATA. Ширина буфера - 8 бит, глубина - 32 байта.

Из буфера FIFO данные поступают в сдвиговый регистр, из которого они побитно (LSB) появляются на выходе TX. Старт-бит, стоп-бит и бит контроля четности (если разрешен) формируются автоматически. На рис.33 показаны возможные варианты формата передаваемых данных, для данного контроллера.

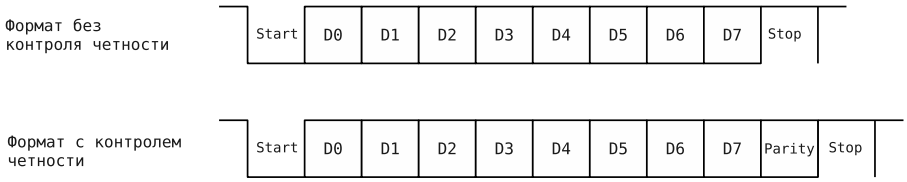

Рис. 33: Формат пакета данных

После завершения передачи последнего слова данных из буфера FIFO, после формирования стоп-бита, выход TX устанавливается в состояние логической '1', устанавливается признак пустого сдвигового регистра UARTxST(TS). После этого бит автоматически установится в состояние логического '0' как только данные появляются в буфере FIFO. Если буфер FIFO пуст, то выставляется бит UARTxST(TE). О заполнении буфера сигнализируют биты UARTxST(TF) - буфер полон и UARTxST(TH) - меньше половины буфера заполнено. Для контроля заполнения буфера FIFO существует счетчик - UARTxST(TCNT).

Существует несколько механизмов формирования запросов прерывания по событиям буфера: если установлен бит UARTxCR(TF), то прерывания вырабатываются если буфер FIFO передатчика заполнен менее чем наполовину; если установлен бит UARTxCR(TI), то будет выработано единичное прерывание в момент, когда буфер FIFO передатчика опустеет.

Так же могут быть разрешены прерывания по пустому сдвиговому регистру передатчика, но в этом случае нужно помнить, что признак пустого сдвигового регистра передатчика устанавливается только в том случае, когда завершена передача очередного символа и в буфере передатчика FIFO не содержится новых данных. Таким образом, прерывание от пустого сдвигового регистра будет генерироваться только если завершена посылка очередного байта и в буфере передатчика FIFO нет новых данных.

При запрете работы передатчика, передача немедленно прекращается, обрывается так же передача текущего слова данных из сдвигового регистра передатчика.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Если разрешен аппаратный контроль потока данных, то данные из сдвигового регистра будут передаваться только если CTS в состоянии логического ’0’. Если в процессе передачи сигнал выставить в состояние логической ’1’, то передача прекратиться, возобновиться только, когда CTS вернуть в состояние логического ’0’.

9.3.3 Прием данных

Для включения приемника и разрешения его работы предназначен бит UARTxCR(RE). Принятые данные записываются в буфер FIFO приема. Пользователю в обычном режиме доступен только выход буфера FIFO. Обращение к нему происходит через UARTxDATA. Ширина буфера - 8 бит, глубина - 32 байта. Принимаемый сигнал проходит цифровой фильтр нижних частот.

Приемник следит за состоянием сигнала на входе и в случае перехода сигнала из состояния логической ’1’ в состояние ’0’ начинается прием данных. Через $\frac{T_{UART}}{2}$ фиксируется состояние входного сигнала, где T_{UART} - период одного бита слова данных. Если не будет зафиксирован стартовый бит, то приемник перейдет обратно в режим ожидания. Если стартовый бит принят, то произойдет прием оставшихся бит слова данных и служебных бит. После приема последнего бита, данные помещаются в буфер FIFO, устанавливается бит UARTxST(DR). В регистре UARTxST устанавливаются биты ошибок приема, если таковые зафиксированы. Биты ошибок очищаются только программно.

В случае, если в сдвиговом регистре находятся принятые данные и буфер FIFO полон, а на входе приемника зафиксирован старт бит, то данные в сдвиговом регистре будут потеряны. При этом будет установлен бит UARTxST(OV).

О заполнении буфера сигнализируют биты UARTxST(RF) - буфер полон и UARTxST(RH) - половина или больше половины буфера заполнено. Для контроля заполнения буфера FIFO существует счетчик - UARTxST(RCNT).

Существует несколько механизмов формирования запросов прерывания для режима приема данных: если установлен бит UARTxCR(RF), то прерывания вырабатываются если буфер FIFO приемника заполнен наполовину или более чем наполовину; если установлен бит UARTxCR(RI), то будет вырабатываться единичное прерывание в моменты, когда принят очередной байт данных.

Если разрешен аппаратный контроль потока данных и буфер FIFO полон, то RTS перейдет в состояние логической ’1’. Как только из буфера будет считано хотя бы одно слово данных, RTS автоматически перейдет в состояние логического ’0’.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.3.4 Установка скорости передачи

Для установки скорости передачи данных существует предделитель системной частоты, коэффициент деления которого задается в регистре UARTxBDR (формула приведена ниже).

$$BRDIV = \frac{F_{sys}}{8 \cdot F_{UART} - 1};$$

9.3.5 Режимы самотестирования

В режимах самотестирования все выходы контроллера переводятся в неактивный режим.

9.3.5.1 Режим самотестирования на уровне интерфейсных линий

В данном режиме внутри микросхемы выход передатчика UART коммутируется со входом приемника, а сигнал CTS коммутируется с RTS. Включение этого режима происходит путем установки бита UARTxCR(LB).

9.3.5.2 Режим самотестирования на уровне данных

В данном режиме разрешается запись в буфер FIFO приема и чтение из буфера FIFO передачи. Чтение и запись осуществляются через регистр UARTxFIFODBG. Включение этого режима происходит путем установки бита UARTxCR(LB).

9.3.6 Формирование прерываний

Запросы прерывания формируются в следующих случаях:

От сдвигового регистра передатчика:

- работа передатчика разрешена: установлен бит UARTxCR(TE);
- прерывания от передатчика разрешены: установлен бит UARTxCR(TI).

От буфера FIFO передачи:

- работа передатчика разрешена: установлен бит UARTxCR(TE);
- прерывания от буфера разрешены: установлен бит UARTxCR(TF);

Все права защищены © ОАО «Мультиклет» 2010 — 2015

От сдвигового регистра приемника:

- работа приемника разрешена: установлен бит UARTxCR(RE);
- прерывания от передатчика разрешены: установлен бит UARTxCR(RI).

От буфера FIFO приема:

- работа приемника разрешена: установлен бит UARTxCR(RE);
- прерывания от буфера разрешены: установлен бит UARTxCR(RF);

9.3.6.1 Режим отложенных прерываний приемника

Режим включается установкой бита UARTxCR(DI). Прерывание от приемника формируется только в случае образование паузы после приема последнего слова данных. Пауза равна времени приема 4,5 слов данных. Если разрешено прерывание от буфера FIFO приема, то прерывание от сдвигового регистра будут очищаться. Активными будут только прерывания от буфера.

Примечание: разрешение данного режима не влияет на формирование запроса прерывания при получении символа прекращения транзакции.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.3.7 Описание регистров

Базовый адрес UART0 - 0xC000 0100
Базовый адрес UART1 - 0xC000 0200
Базовый адрес UART2 - 0xC010 0100
Базовый адрес UART3 - 0xC010 0200

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
UARTxDATA	00h	RW	Регистр данных (FIFO)
UARTxST	04h	R	Регистр состояния
UARTxCR	08h	RW	Регистр управления
UARTxDBR	0ch	RW	Регистр коэффициента делителя тактовой частоты

UARTxDATA	**Регистр данных**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—																															
Описание	—																								DATA							

8-31 — *зарезервировано*
0-7 DATA Регистр данных. При записи является входом 32 байтового буфера FIFO передатчика. При чтении является входом 32 байтового буфера FIFO приемника.

UARTxST	**Регистр состояния**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—						—						—									—	—	—	—	—	—	—	—	—	—	—
Описание	RCNT						TCNT						*зарезервировано*									RF	TF	RH	TH	FE	PE	OV	BR	TE	TS	DR

26-31 RCNT Счетчик данных FIFO приемника
20-25 TCNT Счетчик данных FIFO передатчика
11-19 — *зарезервировано*
10 RF FIFO приемника заполнен
9 TF FIFO передатчика заполнен
8 RH FIFO приемника заполнен наполовину или более чем наполовину
7 TH FIFO передатчика заполнен менее чем наполовину
6 FE Ошибка формата принятых данных
5 PE Ошибка контроля четности принятых данных
4 OV Один или более символов принятых данных потерян из-за переполнения
3 BR Специальный символ завершения обмена получен (BREAK)
2 TE FIFO передатчика пуст
1 TS Сдвиговый регистр передатчика пуст
0 DR Новые символы зафиксированы в регистре приемника

Все права защищены © ОАО «Мультиклет» 2010 — 2015

UARTxCR	Регистр управления UART																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—	—																—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Описание	FA	*зарезервировано*																SI	DI	BI		RF	TF		LB	FL	PE	PS	TI	RI	TE	RE

31	FA	Разрешение использования FIFO приемника и передатчика (’0’- запрещено, ’1’- разрешено)
15-30	—	*зарезервировано*
14	SI	Разрешение прерывания по пустому регистру сдвига передатчика (’0’– запрещено, ’1’– разрешено)
13	DI	Задержанное прерывание приемника (4 символа + 4 бита и нет нового символа) (’0’– запрещено, ’1’– разрешено)
12	BI	Разрешение прерывания по приему символа BREAK (’0’– запрещено, ’1’– разрешено)
11	—	*зарезервировано*
10	RF	Разрешение прерывания по FIFO приемника (’0’– запрещено, ’1’– разрешено)
9	TF	Разрешение прерывания по FIFO передатчика (’0’– запрещено, ’1’– разрешено)
8	—	*зарезервировано*
7	LB	Включение внутренней обратной петли для самотестирования (’0’– отключена, ’1’– включена)
6	FL	Разрешение управления потоком данных (CTS/RTS) (’0’– запрещен, ’1’– разрешен)
5	PE	Разрешение контроля четности (’0’– запрещен, ’1’– разрешен)
4	PS	Выбор типа контроля четности (’0’– на четность, ’1’– на нечетность)
3	TI	Разрешение прерывания передатчика в момент, когда буфер передатчика FIFO опустеет (’0’– запрещено, ’1’– разрешено)
2	RI	Разрешение прерывания приемника по принятию символа (’0’– запрещено, ’1’– разрешено)
1	TE	Разрешение работы передатчика (’0’– запрещена, ’1’– разрешена)
0	RE	Разрешение работы приемника (’0’– запрещена, ’1’– разрешена)

UARTxBDR	Регистр коэффициента делителя тактовой частоты																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—																				—											
Описание	*зарезервировано*																				BRDIV											

12-31	—	*зарезервировано*
0-11	BRDIV	Коэффициент деления системной частоты для формирования требуемой скорости обмена данными BRDIV = Fsys/BR*8, где Fsys в Гц, а BR в бит/с

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.4 Интерфейс SPI(SPIx)

9.4.1 Краткие характеристики

- может работать в режимах «ведущий» или «ведомый»;
- поддерживаются все режимы SPI, а также трехпроводной режим, в котором используется одна двунаправленная линия данных;
- настраиваемая длина слова данных;
- отдельные буферы FIFO глубиной 32 слова для приема и передачи;
- устанавливаемый пользователем формат слова данных — LSB или MSB;
- устанавливаемые пользователем полярность CPOL и фаза CPHA тактового сигнала;
- устанавливаемая пользователем скорость обмена данными.

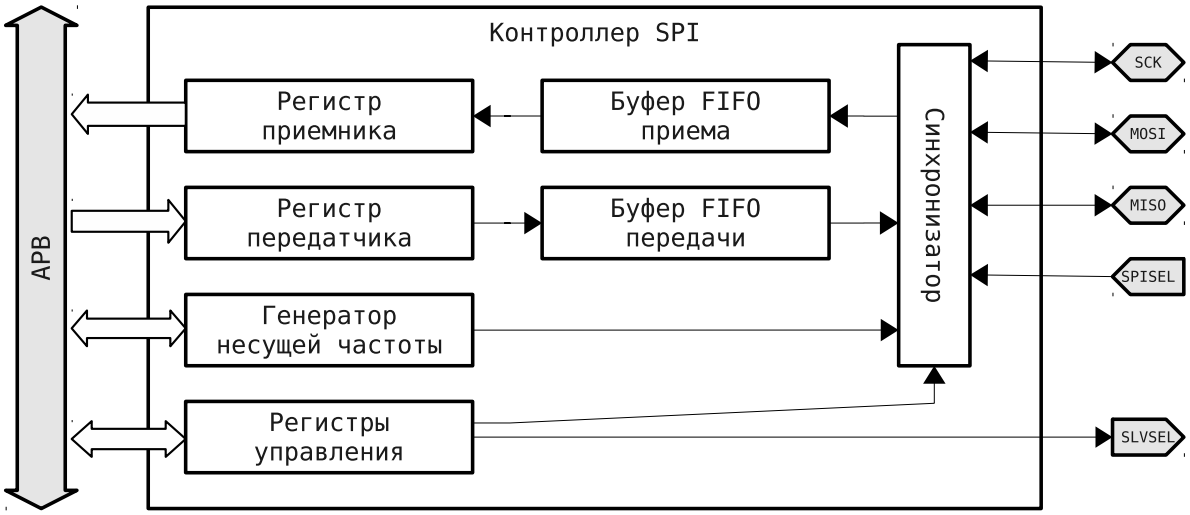

Рис. 34: Блок-схема SPIx

Интерфейс SPI является полно-дуплексным. Передача начинается как только «ведущий» перевел сигнал SLVSEL у соответствующего «ведомого» в активное состояние, так же SCK выведена из неактивного состояния.

Данные передаются «ведущим» по линии MOSI, принимаются по MISO.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

В системе с одним «ведущим» и одним «ведомым» можно не управлять сигналом SLVSEL, он может всегда находиться в активном состоянии.

В системе с несколькими «ведущими» каждый из них производит мониторинг сигнала SPISEL что бы избежать конфликтов с другим «ведущим». Если на входе SPISEL появился активный уровень, то «ведущий», принявший его, выключается.

В процессе приема или передачи, данные меняются при изменении состояния SCK. Значения начального состояния и активного фронта SCK определяют режим работы SPI. На рис.35 приведены диаграммы режимов работы SPI при передаче 0x55 в режиме MSB. Стоит отметить, что данные должны быть готовы в буфере передачи до первой смены состояния SCK.

При работе в режиме «ведомый» передача данных по линии MISO будет задержана так как необходимо синхронизировать передатчик. Смотрите описание в п.9.4.6.

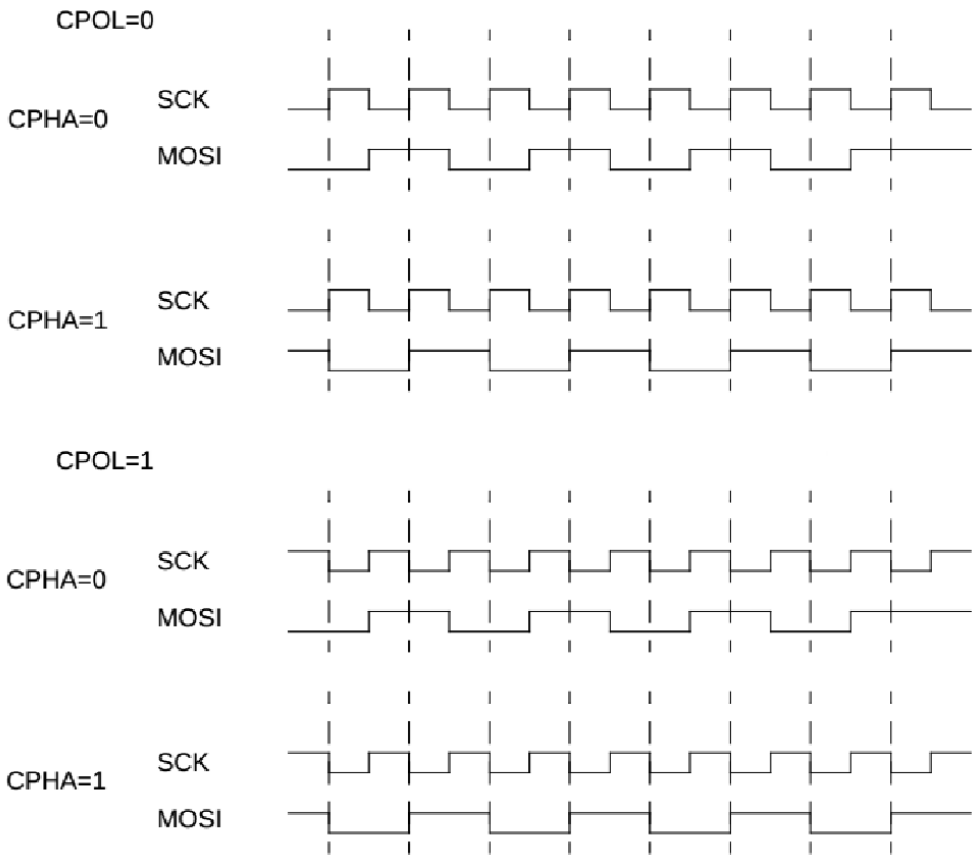

Рис. 35: Режимы работы SPI

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.4.2 Трех-проводный режим

Контроллер интерфейса может быть настроен на работу в трехпроводном режиме (смотрите регистр SPIxCR(TWEN)). Работа будет происходить в полудуплексном режиме. В данном режиме используется одна двунаправленная линия приема-передачи данных вместо двух однонаправленных для приема и передачи.

Для обмена данными используется линия MOSI, MISO в данном режиме не задействуется. Направление обмена данными выбирается в регистре SPIxCR(TTO).

Смена данных на линии приема-передачи в соответствии с полярностью и фазой сигнала тактирования так же как и в четырех-проводном режиме работы.

9.4.3 Прием и передача данных

Контроллер интерфейса имеет отдельные регистры и буферы FIFO для приема и передачи. Разрядность буфера FIFO - 32 бита, глубина - 32 слова. Разрядность слова данных определяется в SPIxCR(TWEN). Если в буфере передачи есть свободное место, то устанавливается бит в регистре SPIxST(NF), в буфер можно посылать данные. Если в буфере приема содержится хотя бы одно полностью принятое слово, то устанавливается бит в регистре SPIxST(NE). Если возникла ситуация, что принято 33 и более слов данных, то устанавливается бит SPIxST(OV). При работе в режиме «ведомого» контроллер интерфейса может детектировать ситуацию, когда он был выбран «ведущим» (SPISEL принял значение логический '0'), а в буфере передачи нет данных. В этом случае устанавливается бит в регистре SPIxST(UN).

9.4.4 Тактовый сигнал SCK

Генерировать сигнал SCK контроллер интерфейса может только при работе в режиме «ведущего» . Параметры для генератора SCK задаются в регистре SPIxCR.

$$F_{SCK} = \frac{F_{sys}}{(4-(2\cdot FACT))(PM+1)}, \quad DIV16 = 0;$$

$$F_{SCK} = \frac{F_{sys}}{16(4-(2\cdot FACT))(PM+1)}, \quad DIV16 = 1;$$

9.4.5 Работа в режиме «ведущий»

В данном режиме как только данные появляются в буфере FIFO передачи, они сразу передаются. Если данные переданы и буфер передачи пуст, то SCK не подается.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Если во время работы в данном режиме сигнал SPISEL примет значение логический ’0’, контроллер интерфейса прекратит передачу данных и установит бит в регистре SPIxST(MME). Бит разрешения работы в режиме «ведущий» в регистре SPIxCR(MS) сбрасывается.

Поведение контроллера интерфейса при изменении сигнала SPISEL определяется в регистре SPIxCR(IGSEL).

9.4.6 Работа в режиме «ведомый»

В данном режиме контроллер интерфейса не управляет линями интерфейса пока сигнал SPISEL от «ведущий» не примет значение логический ’0’. Как только это произошло, MISO конфигурируется как выход и этот выход принимает состояние соответствующее первому биту данных буфера FIFO передачи. Если контроллер интерфейса работает в трех-проводном режиме, то ожидается окончание приема слова по линии MOSI и после этого MOSI конфигурируется как выход. Если буфер передачи пуст, то линия передачи принимает состояние логической ’1’.

Частота SCK в данном режиме должна удовлетворять следующему условию:

$$F_{SCK} \leq \frac{F_{sys}}{8};$$

Передатчик контроллера интерфейса синхронизируется от внешнего SCK, поэтому новые данные на линии MISO появятся только через 2 периода F_{sys} после фронта SCK.

Так же контроллер интерфейса может применять внутренний фильтр для SCK, это управляется регистром SPIxCR(PM). SPIxCR(PM) определяет какое время, выраженное в периодах F_{sys} сигнал SCK должен быть стабилен. При каждом увеличении PM на 1 задержка выдачи очередных данных на линию MISO увеличивается на 2 периода F_{sys}. Так же необходимо увеличивать период SCK на это же значение, рассчитанной из условий, упомянутых выше.

Если буфер передатчика пуст, а сигнал SPISEL в активном уровне и «ведущий» подает сигнал тактирования, то на выходе передатчика будет постоянно выдаваться ’1’.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.4.7 Описание регистров

Базовый адрес SPI0 - 0xC010 2000
Базовый адрес SPI1 - 0xC010 2100.

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра. Биты в статусных регистрах очищаются записью в них ’1’ если очистка бита возможна.

Регистр	Смещение адреса	Доступ	Описание
SPIxCFG	00h	RW	Регистр установок конфигурации
SPIxCR	20h	RW	Регистр управления
SPIxST	24h	RW	Регистр состояния
SPIxMSK	28h	RW	Регистр маски
SPIxCMD	2Ch	RW	Регистр команд
SPIxTX	30h	W	Регистр передаваемых данных
SPIxRX	34h	R	Регистр принимаемых данных
SPIxSS	38h	RW	Регистр выбора ведомого устройства

SPIxCFG	Регистр конфигурации																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0x3								0				0	0		0	0x20								0							
Описание	SSSZ								MAXWLEN				TWEN	-		SSEN	FDEPTH								—							

24-31	SSSZ	число линий выборок ведомого устройства
20-23	MAXWLEN	максимальная поддерживаемая длина слова данных (0-32)
19	TWEN	разрешение трёхпроводного режима (’1’ - разрешено, ’0’ - запрещено)
17-18	—	*зарезервировано*
16	SSEN	разрешение сигналов выбора ведомого устройства (’1’ - разрешено, ’0’ - запрещено)
8-15	FDEPTH	глубина FIFO RX, TX
0-7	—	*зарезервировано*

SPIxCR	Регистр управления																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	0	0	0	0				0				0	0	0	0	0					0			0	0		
Описание	-	LOOP	CPOL	CPHA	DIV16	REV	MS	EN	LEN				PM				TWEN	-	FACT	-	CG					—			TTO	—		

31	—	*зарезервировано*
30	LOOP	режим самотестирования (’1’ - разрешено, ’0’ - запрещено)
29	CPOL	состояние SCK в режиме ожидания (’1’ - логическая 1 , ’0’ - логический 0)
28	CPHA	настройка фазы синхросигнала (’1’ - данные будут прочитаны на втором переходе состояния SCK, ’0’ - данные будут прочитаны при первом переходе состояния SCK)
27	DIV16	разрешение делителя на 16 (только в режиме ведущего) (’1’ - разрешено, ’0’ - запрещено)
26	REV	направление передачи (’1’ - MSB, ’0’ - LSB)
25	MS	выбор режима (’1’ - ведущий, ’0’ - ведомый)
24	EN	разрешение работы (’1’ - разрешено, ’0’ - запрещено)
20-23	LEN	длина слова данных 0x0 - длина слова 32 бита 0x1-0x2 - недопустимые значения 0x3-0xf - 4-16 бит соответственно
16-19	PM	режим предделителя (только в режиме ведущего): если DIV16 = 0: $F_{sck} = \frac{F_{sys}}{(4-(2 \cdot FACT)) \cdot (PM+1)}$ если DIV16 = 1: $F_{sck} = \frac{F_{sys}}{16 \cdot (4-(2 \cdot FACT)) \cdot (PM+1)}$

Все права защищены © ОАО «Мультиклет» 2010 — 2015

15	TWEN	трёхпроводной режим (’1’ - разрешено, ’0’ - запрещено)
14	—	*зарезервировано*
13	FACT	режим предделителя частоты (1 - совместимость с MCP83xx)
12	—	*зарезервировано*
7-11	CG	прекращение подачи сигнала SCK после передачи каждого слова данных на N периодов (только в режиме ведущего)
4-6	—	*зарезервировано*
3	TTO	порядок передачи при работе по трёхпроводной линии (’1’ - ведомый передаёт первый, ’0’ - ведущий передаёт первый)
0-2	—	*зарезервировано*

SPIxST	**Регистр состояния**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0																0	0	0	0	0	0	0	0							
Описание	TIP	—																LT	-	OV	UN	MME	NE	NF	—							

Для всех разрядов регистра: (’1’ - наличие признака, ’0’ - отсутствие признака)

31	TIP	передаётся слово данных. Этот бит не сбрасывается пока не будут переданы все данные из буфера передатчика
15-30	—	*зарезервировано*
14	LT	последнее слово данных передано, буфер передатчика пуст и в SPIxCMD записан бит LST (бит очищается записью ’1’)
13	—	*зарезервировано*
12	OV	буфер приемника заполнен, новые данные игнорируются (бит очищается записью ’1’)
11	UN	признак запроса от «ведущего» (только в режиме «ведомого»). Бит равен ’1’ если сигнал SPISEL в активном состоянии и подается сигнал тактирования от ведущего
10	MME	ошибка при работе в системе с несколькими ведущими (возникает, когда в режиме ведущего появляется сигнал SPISEL)
9	NE	буфер приемника содержит данные
8	NF	буфер передатчика имеет свободное место
0-7	—	*зарезервировано*

SPIxMSK	**Регистр маски**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0																0	0	0	0	0	0	0	0							
Описание	TIPE	—																LTE	-	OVE	UNE	MMEE	NEE	NFE	—							

Для всех разрядов регистра: (’1’ - прерывание разрешено , ’0’ - прерывание запрещено)

31	TIPE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(TIP) переключается из ’0’ в ’1’
15-30	—	*зарезервировано*
14	LTE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(LT) переключается из ’0’ в ’1’
13	—	*зарезервировано*
12	OVE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(OV) переключается из ’0’ в ’1’
11	UNE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(UN) переключается из ’0’ в ’1’
10	MMEE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(MME) переключается из ’0’ в ’1’
9	NEE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(NE) переключается из ’0’ в ’1’
8	NFE	если установлен этот бит, то генерируется прерывание когда бит SPIxST(NF) переключается из ’0’ в ’1’
0-7	—	*зарезервировано*

Все права защищены © ОАО «Мультиклет» 2010 — 2015

SPIxCMD	Регистр команд																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0									0	0																					
Описание	—									LST	—																					

23-31 — *зарезервировано*

22 LST После того, как этот бит будет установлен в единицу, будет разрешена установка бита SPIxST(LT) когда слово передано и буфер передатчика пуст. Если разрешена работа в трехпроводном режиме, бит SPIxST(LT) установится в единицу после завершения передачи всех данных. Этот бит автоматически очищается, когда бит SPIxST(LT) устанавливается в единицу и всегда вычитывается как '0'

0-21 — *зарезервировано*

SPIxTX	Регистр передаваемых данных																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	TDATA																															

0-31 TDATA данные для передачи, ширина и порядок следования битов определяется в регистре SPIxCR. Данные действительны, если SPIxST(NF) = '1'. для REV = '0' SPIxCR – LSB размещается в бите 0, для REV = '1' SPIxCR – MSB размещается в бите 31 При 8 битном слове байт 0xAB на передачу получит следующее расположение: для REV = '0' - 0x000000AB, для REV = '1' - 0xAB000000

SPIxRX	Регистр принимаемых данных																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	RDATA																															

0-31 RDATA принятые данные, ширина и порядок следования битов определяются в регистре SPIxCR. Данные действительны, если NE = '1' в регистре SPIxST. для REV = '0' SPIxCR – MSB размещается в бите 15 (при ширине слова 4-16 бит), для REV = '1' SPIxCR – LSB размещается в бите 16 (при ширине слова 4-16 бит). При 8 битном слове байт 0xAB получит следующее расположение: для REV = '0' - 0x0000AB00, для REV = '1' - 0x00AB0000

SPIxSS	Регистр выбора ведомого устройства																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																													0		
Описание	—																													SLVSEL		

3-31 — *зарезервировано*

0-2 SLVSEL номер подчиненного устройства, с которым необходимо произвести обмен данными

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.5 Интерфейс I^2C «ведущий» (I2C0)

9.5.1 Краткие характеристики

- работает в режиме «ведущий»;
- совместим со стандартом Philips I^2C;
- поддерживает 7-ми и 10-ти битную адресацию;
- скорость обмена - 100кбит/с и 400кбит/с;
- требуется установка внешних подтягивающих резисторов на линии SCL и SDA.

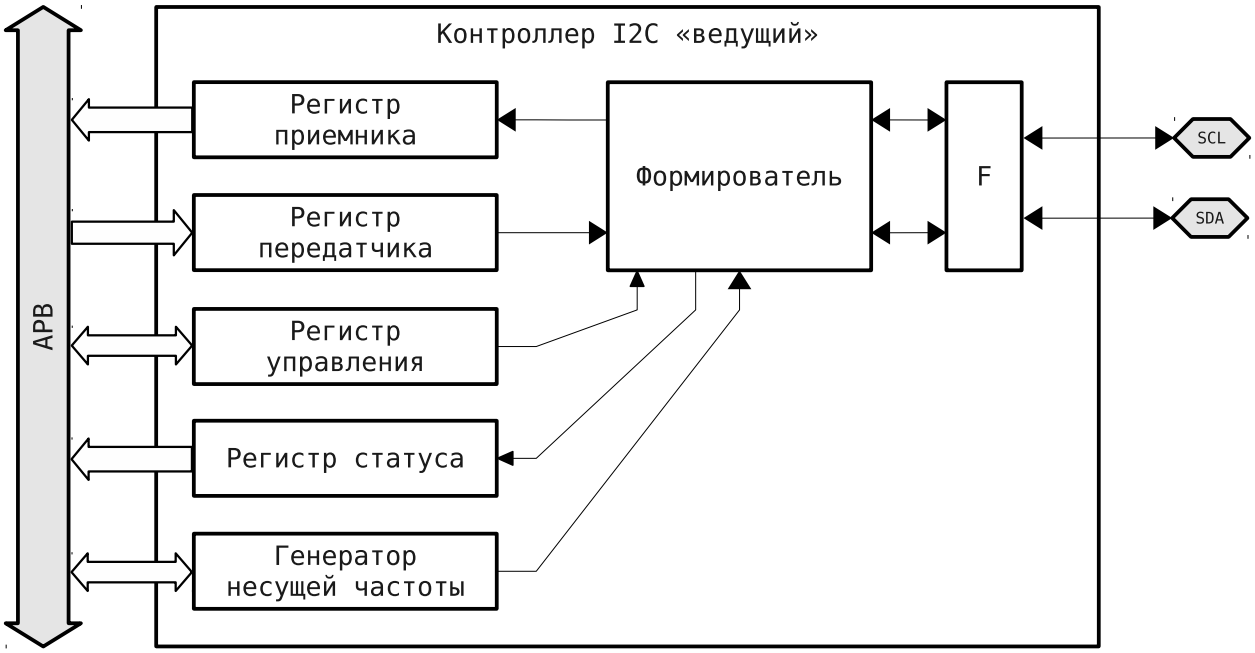

Рис. 36: Блок-схема I2C0 («ведущий»)

В МП I2C0 работает только в режиме «ведущий».

I^2Cпростой двухпроводной последовательный интерфейс с возможностью работы нескольких «ведущих» на одной физической линии. Интерфейс обеспечивает детектирование коллизий и арбитраж. I^2C имеет 2 физических линии SDA (serial data line) и SCL (serial clock line).

На рис.36 приведена блок-схема описываемого контроллера интерфейса. На входе внешних линий интерфейса установлен цифровой фильтр низких частот.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.5.2 Общее описание протокола приема-передачи

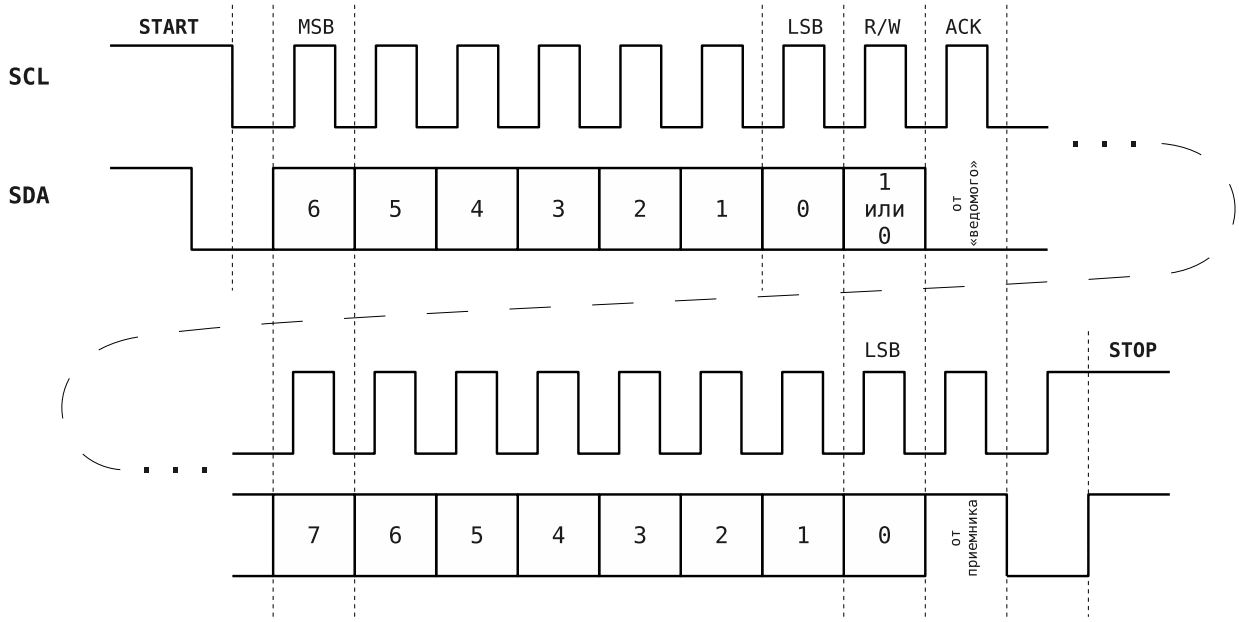

Рис. 37: Транзакция на шине I^2C

Прием-передача данных осуществляется побайтно.

Начало транзакции по шине I^2C определяется состоянием «START» на линиях SDA, SLC: переход линии SDA из состояния ’1’ в состояние ’0’ при состоянии SCL - ’1’.

Окончание транзакции определяется состоянием «STOP» на линиях SDA, SLC: переход линии SDA из состояния ’0’ в состояние ’1’ при состоянии SCL - ’1’.

Состояния «START», «STOP» формируются только «ведущим» шины. После формирования состояния «START» «ведущим» шины, шина считается занятой и освобождается только после формирования, занявшим ее «ведущим» состояния «STOP». Время между «STOP» и «START» определяется стандартом I2C и зависит от текущей скорости ее работы.

На рис. 37 приведены примеры состояния линий SDA, SLC при транзакциях. «Ведущий» шины формирует состояние «START» и передает 7-ми битный адрес «ведомого» устройства. После адреса следует бит $R/\overline{W}$ определяющий направление передачи данных (’1’ - чтение из «ведомого» устройства, ’0’ - запись в «ведомое» устройство). После передачи адреса и бита $R/\overline{W}$, «ведущий» шины освобождает линию SDA, а «ведомый» должен привести линию SDA в состояние ’0’. Если этого не произошло, то считается, что не принят сигнал ACK(подтверждение) от «ведомого». «Ведущий» шины мо-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

жет сформировать состояние «STOP» и повторить транзакцию с этим «ведомым» или предпринять другие действия, заложенные в алгоритм его работы.

Если от «ведомого» принят сигнал ACK, то начинается передача данных, направление которой было опеределено битом $R/\overline{W}$. Данные могут передаваться до тех пор, пока приемник на каждый переданный байт информации отвечает ACK. Т.е. после передачи каждого байта данных, передатчик освобождает на один период SLC линию SDA, что бы приемник мог сообщить принят или не принят им байт информации или готов или не готов он принимать следующий. После ответа NAK (во время ожидания ACK подана '1') «ведущий» формирует состояние «STOP». «Ведущий» так же может прервать транзакцию, сформировав состояние «STOP».

9.5.3 Генерация несущей частоты

Контроллер I^2C формирует две частоты: для внешнего тактирования по линии SCL и для тактирования внутренних блоков в пять раз превышающую частоту на линии SCL. Для рассчета значения коэффициента деления предделителя частоты тактирования контроллера (I2CxPSC(PSC)) используется следующая формула:

$$PSC = \frac{F_{sys}}{5 \cdot F_{SCL}} - 1$$

Коэффициента деления предделителя может быть изменен только при отключенном контроллере I^2C (бит I2CxCR(EN)).

Минимально рекомендуемое значение коэффициента равно 3, что бы соблюдались допуски протокола по синхронизации. Это так же накладывает ограничение на минимальную частоту тактирования контроллера. При скорости обмена 100кбит/с минимальная рекомендуемая частота тактирования будет равной 2МГц. Но при скорости обмена 400кбит/с частота 2МГц будет недостаточной для соблюдения требования ко времени установки данных. Исходя из этого частота тактирования контроллера должна быть не менее 20МГц.

9.5.4 Алгоритм работы с интерфейсом

Для разрешения работы контроллера надо записать в I2CxPSC(PSC) необходимое значение и установить бит I2CxCR(EN) = '1'. Прерывания разрешаются битом I2CxCR(IEN).

Ниже описаны примеры взаимодействия с «ведомым» устройством. При работе с реальными устройствами внимательно читайте их документация и описание протокола взаимодействия с ними, они могут отличаться от описанных ниже.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.5.4.1 Запись данных

Для передачи байта данных в «ведомое» устройство, «ведущий» I^2C должен сформировать состояние «START» на линиях SDA, SCL, послать адрес «ведомого» устройства и признак направления $R/\overline{W}$ = '0'. «Ведомое» устройство должно ответить посылкой ACK. Затем «ведущий» посылает байт данных, ждет сигнала ACK и формирует состояние «STOP».

- записать байт данных содержащий адрес «ведомого» и $R/\overline{W}$ = '0' в I2CxTX;
- сформировать состояние «START» на линиях SDA, SCL, записав биты I2CxCMD(WR) = '0' и I2CxCMD(STA) = '1';
- подождать пока бит I2CxST(TIP) примет значение '0';
- прочитать бит I2CxST(RxACK). Если бит равен '0', то «ведомый» принял информацию, можно продолжать дальше транзакцию. Если бит равен '1', повторить пункты сначала, «ведомый» по каким-то причинам не принял информацию;
- записать данные для передачи в I2CxTX;
- сформировать состояние «STOP» I2CxCMD(WR) = '1' и I2CxCMD(STO) = '1';
- подождать пока бит I2CxST(TIP) примет значение '0';
- прочитать бит I2CxST(RxACK). Если бит равен '0', то «ведомый» принял данные.

9.5.4.2 Чтение данных

Для чтения байта данных с произвольного адреса в «ведомом» устройстве, «ведущий» I^2C должен сформировать состояние «START» на линиях SDA, SCL, послать адрес «ведомого» устройства и признак направления $R/\overline{W}$ = '0'. За тем «ведущий» посылает байт(ы) содержащие внутренний адрес для «ведомого» устройства. Повторно формирует состояние «START» на линиях SDA, SCL, посылает адрес «ведомого» устройства и признак направления $R/\overline{W}$ = '1'. Принимает байт(ы) данных от «ведомого», отвечая ACK. После принятия последнего байта данных отвечает NACK. Формирует состояние «STOP».

Стоит помнить, что регистр I2CxRX перезаписывается каждый раз, когда принимается новый байт данных.

- записать байт данных содержащий адрес «ведомого» и $R/\overline{W}$ = '0' в I2CxTX;

Все права защищены © ОАО «Мультиклет» 2010 — 2015

- сформировать состояние «START» на линиях SDA, SCL, записав биты I2CxCMD(WR) = ’1’ и I2CxCMD(STA) = ’1’;
- подождать пока бит I2CxST(TIP) примет значение ’0’;
- прочитать бит I2CxST(RxACK). Если бит равен ’0’, то «ведомый» принял информацию, можно продолжать дальше транзакцию. Если бит равен ’1’, повторить пункты сначала, «ведомый» по каким-то причинам не принял информацию;
- записать данные для передачи в I2CxTX и установить бит I2CxCMD(WR) = ’1’;
- подождать пока бит I2CxST(TIP) примет значение ’0’;
- прочитать бит I2CxST(RxACK). Если бит равен ’0’, то «ведомый» принял информацию, можно продолжать дальше транзакцию. Если бит равен ’1’, повторить пункты сначала, «ведомый» по каким-то причинам не принял информацию;
- повторить предыдущие 3 пункта до тех пор, пока не будут переданы все байты, содержащие внутренний адрес в «ведомом»;
- записать байт данных содержащий адрес «ведомого» и $R/\overline{W}$ = ’1’ в I2CxTX;
- сформировать состояние «STOP» I2CxCMD(WR) = ’1’ и I2CxCMD(STO) = ’1’;
- сформировать состояние «START» на линиях SDA, SCL, записав биты I2CxCMD(WR) = ’1’ и I2CxCMD(STA) = ’1’;
- подождать пока бит I2CxST(TIP) примет значение ’0’;
- прочитать бит I2CxST(RxACK). Если бит равен ’0’, то «ведомый» принял информацию, можно продолжать дальше транзакцию. Если бит равен ’1’, повторить пункты сначала, «ведомый» по каким-то причинам не принял информацию;
- прочитать байт данных из «ведомого», установив I2CxCMD(RD) = ’1’, I2CxCMD(STO) = ’1’, I2CxCMD(ACK) = ’1’;
- подождать пока бит I2CxST(TIP) примет значение ’0’;
- прочитать данные из регистра I2CxRX, сохранить в ПД.
- если требуется прочитать несколько байт данных из «ведомого» то необходимо повторять предыдущие 3 пункта но не устанавливать I2CxCMD(STO) = ’1’, I2CxCMD(ACK) = ’1’ до тех пор, пока не будет считано требуемое количество байт данных;

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.5.5 Описание регистров

Базовый адрес I2C0 - 0xC000 1000.

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
I2CxPSC	00h	RW	Регистр предделителя тактовой частоты
I2CxCR	04h	RW	Регистр управления
I2CxTX	08h	W	Регистр передаваемых данных
I2CxRX	08h	R	Регистр принимаемых данных
I2CxCMD	0Ch	W	Регистр команд
I2CxST	0Ch	R	Регистр состояния

I2CxPSC	**Регистр предделителя**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																PSC															

16-31 — *зарезервировано*
0-15 PSC значение предделителя

I2CxCR	**Регистр управления**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								EN	IEN	—					

8-31 — *зарезервировано*
7 EN разрешение работы контроллера (’1’ - разрешено, ’0’ - запрещено)
6 IEN разрешение прерывания по завершению передачи (’1’ - разрешено, ’0’ - запрещено)
0-5 — *зарезервировано*

I2CxTX	**Регистр передаваемых данных**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								TDATA							RW

8-31 — *зарезервировано*
7 TDATA старшие 7 бит передаваемых данных
0-6 RW бит $R/\overline{W}$ при передачи адреса «ведомого», в остальных случаях – младший бит данных

I2CxRX	**Регистр принимаемых данных**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								RDATA							

8-31 — *зарезервировано*
0-7 RDATA последний принятый байт данных

Все права защищены © ОАО «Мультиклет» 2010 — 2015

I2CxCMD	Регистр команд																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								STA	STO	RD	WR	ACK	—	IACK	

8-31	—	*зарезервировано*
7	STA	сформировать последовательность START(RESTART) ('1' — сформировать)
6	STO	сформировать последовательность STOP ('1' — сформировать)
5	RD	чтение из ведомого устройства ('1' — считать)
4	WR	запись в ведомое устройство ('1' — записать)
3	ACK	подтверждение получения данных ('0' – ACK, '1' – NACK)
1-2	—	*зарезервировано*
0	IACK	сброс бита I2CxST(IF) ('1' — сбросить)

I2CxST	Регистр состояния																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								RxACK	BUSY	AL	—			TIP	IF

8-31	—	*зарезервировано*
7	RxACK	принят ACK
6	BUSY	шина занята (обнаружено состояние START, сбрасывается при обнаружении STOP)
5	AL	потеря контроля над шиной
2-4	—	*зарезервировано*
1	TIP	признак передачи данных, и формирования STOP
0	IF	байт передан или потерян контроль над линией. Если бит I2CxCR(IEN) = '1', то будут возникать запросы прерывания. Новые прерывания будут возникать даже если этот бит не был очищен

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.6 Интерфейс I^2C «ведомый» (I2C1)

9.6.1 Краткие характеристики

- работает в режиме «ведомый»;
- совместим со стандартом Philips I^2C;
- поддерживает 7-ми битную адресацию;
- скорость обмена - 100кбит/с и 400кбит/с;
- требуется установка внешних подтягивающих резисторов на линии SCL и SDA.

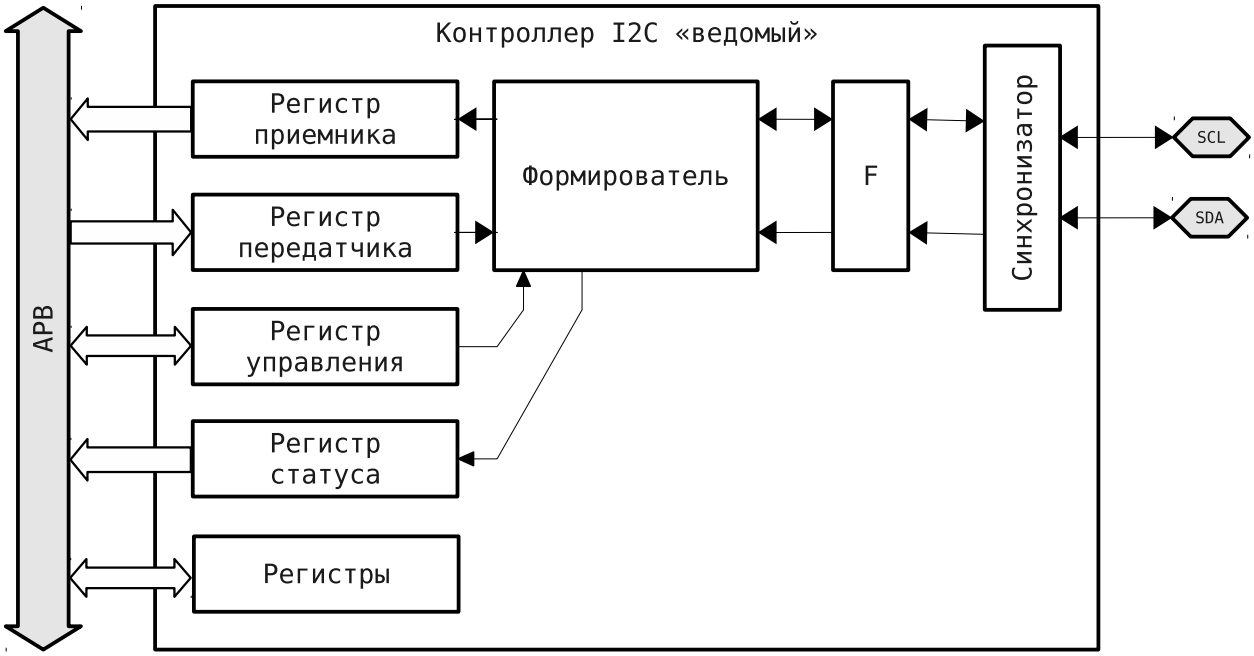

Рис. 38: Блок-схема I2C1 («ведомый»)

В МП I2C1 работает только в режиме «ведомый».

I^2Cпростой двухпроводной последовательный интерфейс с возможностью работы нескольких «ведущих» на одной физической линии. Интерфейс обеспечивает детектирование коллизий и арбитраж. I^2C имеет 2 физических линии SDA (serial data line) и SCL (serial clock line).

На рис.38 приведена блок-схема описываемого контроллера интерфейса. На входе внешних линий интерфейса установлен цифровой фильтр низких частот.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.6.2 Общее описание протокола приема-передачи

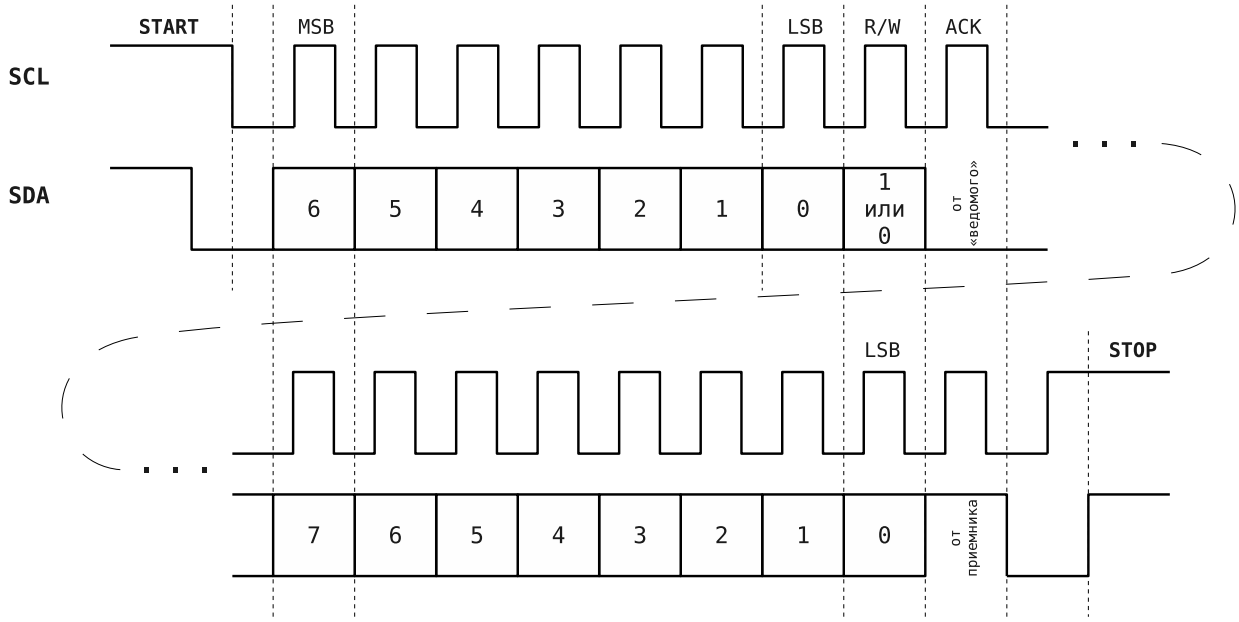

Рис. 39: Транзакция на шине I^2C

Прием-передача данных осуществляется побайтно.

Начало транзакции по шине I^2C определяется состоянием «START» на линиях SDA, SCL: переход линии SDA из состояния ’1’ в состояние ’0’ при состоянии SCL - ’1’.

Окончание транзакции определяется состоянием «STOP» на линиях SDA, SLC: переход линии SDA из состояния ’0’ в состояние ’1’ при состоянии SCL - ’1’.

Состояния «START», «STOP» формируются только «ведущим» шины. После формирования состояния «START» «ведущим» шины, шина считается занятой и освобождается только после формирования, занявшим ее «ведущим» состояния «STOP». Время между «STOP» и «START» определяется стандартом I2C и зависит от текущей скорости ее работы.

На рис. 39 приведены примеры состояния линий SDA, SLC при транзакциях. «Ведущий» шины формирует состояние «START» и передает 7-ми битный адрес «ведомого» устройства. После адреса следует бит $R/\overline{W}$ определяющий направление передачи данных (’1’ - чтение из «ведомого» устройства, ’0’ - запись в «ведомое» устройство). После передачи адреса и бита $R/\overline{W}$, «ведущий» шины освобождает линию SDA, а «ведомый» должен привести линию SDA в состояние ’0’. Если этого не произошло, то считается, что не принят сигнал ACK(подтверждение) от «ведомого». «Ведущий» шины мо-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

жет сформировать состояние «STOP» и повторить транзакцию с этим «ведомым» или предпринять другие действия, заложенные в алгоритм его работы.

Если от «ведомого» принят сигнал ACK, то начинается передача данных, направление которой было определено битом $R/\overline{W}$. Данные могут передаваться до тех пор, пока приемник на каждый переданный байт информации отвечает ACK. Т.е. после передачи каждого байта данных, передатчик освобождает на один период SLC линию SDA, что бы приемник мог сообщить принят или не принят им байт информации или готов или не готов он принимать следующий. После ответа NAK (во время ожидания ACK подана '1') «ведущий» формирует состояние «STOP». «Ведущий» так же может прервать транзакцию, сформировав состояние «STOP».

9.6.3 Генерация несущей частоты

Контроллер I^2C I2C1 тактируется из вне. Сигналы SDA и SCL проходят через синхронизатор и ФНЧ. Их состояния синхронизируются с системной частотой F_{sys}. Это накладывает ограничение на минимальное значение частоты F_{sys}, она должна не менее, чем в 8 раз превышать частоту обмена по шине. Рекомендуется для скорости 100кбит/с $F_{sys} \geq 2MHz$, для скорости 400кбит/с $F_{sys} \geq 6MHz$.

9.6.4 Алгоритм работы с интерфейсом

Интерфейс имеет четыре режима, которые определяются в регистре I2CxCR. Они определяют поведение контроллера после приема или передачи байта данных. Состояния регистрируются в регистре I2CxST. Они же являются источниками запросов прерываний от контроллера.

9.6.4.1 Прием данных от «ведущего»

После приема байта данных контроллер будет отвечать NAK на все последующие, до тех пор пока не будет считан регистр I2CxRX. ACK автоматически формируется при считывании I2CxRX. Байты данных, которые не были подтверждены ACK, не сохраняются в I2CxRX.

Поведение контроллера после приема байта данных задается битом I2CxCR(RMOD).

Если I2CxCR(RMOD) = '0', то контроллер ожидает реакции «ведущего» шины. Будет принимать данные в сдвиговый регистр и отвечать на каждый байт данных NAK, до

Все права защищены © ОАО «Мультиклет» 2010 — 2015

тех пор пока не будет считан регистр I2CxRX.

Если I2CxCR(RMOD) = '1', то контроллер блокирует линию SCL и удерживает ее в состоянии '0' до тех пор пока не будет считан регистр I2CxRX и очищен бит I2CxST(REC) (очищается автоматически при чтении I2CxRX).

9.6.4.2 Передача данных «ведущему»

Передача данных «ведущему» управляется битом I2CxCR(TV). Если бит равен '1', то после принятия адреса контроллер подтвердит его состоянием ACK и начнет передавать данные расположенные в регистре I2CxTX. После того как байт данных был передан, биту I2CxCR(TV) присваивается значение I2CxCR(TAV). Это позволяет отправлять один и тот же байт на все запросы, без траты процессорного времени.

Поведение контроллера после передачи байта данных «ведущему» и получение от него ACK. Если «ведущий» ответил NAK, то контроллер перейдет в состояние ожидания состояния START на линиях SLC, SDA. Бит I2CxST(NAK) примет значение '1'.

Если I2CxCR(TMOD) = '0', то после ответа ACK «ведущим», контроллер продолжает ждать действий от «ведущего». Если он продолжит операции считывания данных из «ведомого», то контроллер будет отправлять данные, которые будут находиться в регистре I2CxTX.

Если I2CxCR(TMOD) = '1', то после ответа ACK «ведущим», контроллер блокирует линию SCL и удерживает ее в состоянии '0' до тех пор пока бит I2CxCR(TV) не будет равен '1'. Пользоваться этим режимом следует осторожно т.к. «ведущий» шины не имеет возможности управлять сигналом SCL, а так же работа шины зависит от программного алгоритма написанного пользователем для системы в которой работает «ведомое» устройство.

Если I2CxCR(TAV) равен '1', то поведения в контроллера при любом значении бита I2CxCR(TMOD) будет одинаковым.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.6.5 Описание регистров

Базовый адрес I2C1 - 0xC000 1100.

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра.

Регистр	Смещение адреса	Доступ	Описание
I2CxSAD	00h	RW	Регистр адреса «ведомого»
I2CxCR	04h	RW	Регистр управления
I2CxST	08h	RW	Регистр состояния
I2CxMSK	0Ch	RW	Регистр маски
I2CxRX	10h	R	Регистр принимаемых данных
I2CxTX	14h	W	Регистр передаваемых данных

I2CxSAD	Регистр адреса «ведомого»																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																									SLVADDR						

7-31 — *зарезервировано*
0-6 SLVADDR 7-ми битный адрес «ведомого» устройства

I2CxCR	Регистр управления																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																											U				0
Описание	—																											RMOD	TMOD	TV	TAV	EN

5-31 — *зарезервировано*
4 RMOD режим приема данных ('1' - «ведомый» принимает данные и удерживает SCL в '0' до тех пор пока данные из I2CxRX не будут считаны (ACK будет послан сформирован), '0' - «ведомый» принимает данные и формирует NAK для данного байта информации и для всех последующих, до тех пор пока не будет считан регистр I2CxRX)
3 TMOD режим передачи данных ('1' - «ведомый» передает один и тот же байт данных и формирует NAK на все запросы после, до тех пор пока I2CxCR(TV) = '0', '0' - «ведомый» передает один байт и удерживает SCL в '0' до тех пор пока I2CxCR(TV) = '0')
2 TV подтверждение передачи ('1' - подтверждает факт передачи данных (после передачи байта данных автоматически принимает значение '0'), '0' - формируется NAK или удерживается SCL в '0' в зависимости от I2CxCR(TMOD))
1 TAV передача данных всегда подтверждена ('1' - разрешено, '0' - запрещено)
0 EN разрешение работы контроллера ('1' - разрешено, '0' - запрещено, линии SCL, SDA в 3-м состоянии)

I2CxST	Регистр состояния																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																													REC	TRA	NAK

3-31 — *зарезервировано*
2 REC байт принят ('1' - принят (автоматически очищается, когда считан регистр I2CxRX), '0' - не принят)
1 TRA байт передан ('1' - передан (очищается записью '1' в I2CxST(TRA)))

Все права защищены © ОАО «Мультиклет» 2010 — 2015

0 NAK сформирован NAK на запрос (’1’ - сформирован NAK)). Если адрес «ведомого» не совпадает с I2CxSAD, то отправка NAK не имеет влияния на этот бит

I2CxMSK	**Регистр маски**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	U																															
Описание	—																													RECE	TRAE	NAKE

3-31 — *зарезервировано*

2 RECE разрешение формирования прерывание по I2CxST(REC) (’1’ - разрешено, ’0’ - запрещено)

1 TRAE разрешение формирования прерывание по I2CxST(TRA) (’1’ - разрешено, ’0’ - запрещено)

0 NAKE разрешение формирования прерывание по I2CxST(NAK) (’1’ - разрешено, ’0’ - запрещено)

I2CxTX	**Регистр передаваемых данных**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								TDATA							

8-31 — *зарезервировано*

0-7 TDATA старшие 7 бит передаваемых данных

I2CxRX	**Регистр принимаемых данных**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																								RDATA							

8-31 — *зарезервировано*

0-7 RDATA последний принятый байт данных

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.7 Контроллер I^2S(I2Sx)

9.7.1 Краткие характеристики

- реализует приемник данных, работающий в режиме «ведущий»;
- совместим со стандартом Philips I^2S;
- выбор разрядности сэмпла: от 16 до 32 бит;
- предделитель синхронизирующей частоты;

9.7.2 Общее описание шины I^2S

Шина предназначена только для обработки звуковых данных, в то время как другие сигналы, такие как субкодирования и управления, передаются отдельно. Для сведения к минимуму количества требуемых контактов используется последовательная шина, состоящая из 3 линий, которая содержит линию для передачи данных двух каналов с временным уплотнением, селекционную линию и линию синхронизации.

Поскольку передатчик и приемник имеют одинаковые тактовые сигналы для передачи данных, передатчик, как ведущее устройство, должен генерировать сигнал синхронизации, сигнал данных и селекционный сигнал. Однако, в сложных системах, может быть несколько передатчиков и приемников и это затрудняет определение ведущего устройства. В таких системах, как правило, существуют ведущие системы управления цифровыми звуковыми потоками данных между различными устройствами. Тогда, передатчики, должны генерировать данные под управлением внешнего синхросигнала и выступать в качестве ведомых устройств. В общем случае I^2S интерфейс состоит из двух раздельных ядер - передатчика и приемника. Оба могут работать в режиме или ведущего или ведомого. Для передачи звука по I^2S в одну сторону требуется как минимум 3 линии:

- Bit clock — SCK (тактирование);
- Word select — WS (линия выбора канала);
- Data line – SD (линия передачи аудио данных).

Сигналы WS и SCK создает только ведущее устройство (рис. 40).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

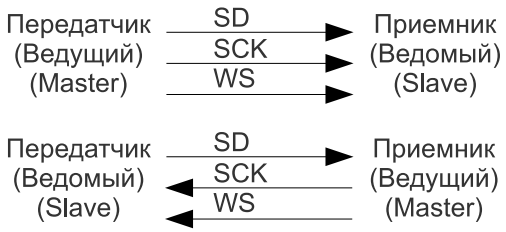

Рис. 40: Направление сигналов I^2S

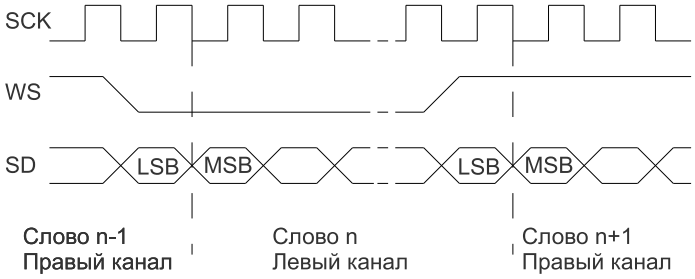

Рис. 41: Временная диаграмма сигналов I^2S

Временная диаграмма сигналов I^2S представлена на рис. 41:

Линия WS указывает, данные какого канала сейчас передаются, низкий уровень ('0') соответствует левому каналу, высокий ('1') — правому, изменение WS происходит на отрицательном фронте SCK. Передатчик изменяет значение линии данных SD при отрицательном фронте сигнала SCK, приемник считывает при положительном. Старший бит слова передается на втором положительном фронте сигнала SCK после изменения сигнала WS.

9.7.3 Особенности текущей реализации I^2S

На передачу контроллер I^2S функционирует в полном объеме. На прием в текущей версии процессора недоступен буфер приема на чтение. Поэтому в текущей версии процессора необходимо программно определять входное значение. Т.е. с помощью GPIOxIN определять текущий уровень тактирующего сигнала и снимать значение с линии данных(все сигналы кроме линии данных на прием работают штатно).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.7.4 Описание регистров

Базовый адрес I^2S- 0xC010 2000

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
I2SxCFG	00h	RW	Регистр настройки приемника
I2SxMSK	04h	RW	Регистр маски прерываний
I2SxINT	08h	RW	Регистр прерываний
I2SxDATA	0Ch-8Ch	W	Регистр данных (32 строки)

I2SxCFG	**Регистр настройки приемника**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0										0						0								0					0	0	0
Описание	—										RES						PSC								—					SWAP	INTEN	RXEN

22-31	—	*зарезервировано*
16-21	RES	количество бит в записываемых аудио данных (размер сэмпла)(16-32 бит)
8-15	PSC	значение делителя передающей частоты
3-7	—	*зарезервировано*
2	SWAP	установка записи левого канала (’1’ - в нечетные адреса, ’0’ - в четные адреса)
1	INTEN	разрешение прерываний (’1’ - разрешены, ’0’ - запрещены)
0	RXEN	разрешение работы (’1’ - разрешены, ’0’ - запрещены)

I2SxMSK	**Регистр маски прерываний**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																													0	0	0
Описание	—																													HSBF	LSBF

Для всех разрядов регистра: (’1’ - разрешение, ’0’ - запрет)

2-31	—	*зарезервировано*
1	HSBF	верхний буфер аудио данных заполнен
0	LSBF	нижний буфер аудио данных заполнен

I2SxINT	**Регистр прерываний**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																													0	0	0
Описание	—																													**HSBF_ST**	**LSBF_ST**

Для всех разрядов регистра: (’1’ - разрешение, ’0’ - запрет)

2-31	—	*зарезервировано*
1	HSBF_ST	верхний буфер аудио данных заполнен
0	LSBF_ST	нижний буфер аудио данных заполнен

Все права защищены © ОАО «Мультиклет» 2010 — 2015

I2SxTX	**Регистр передачи**																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	**TX_OUT**																															

0-31 TX_OUT данные на передачу

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.8 Таймер общего назначения(GPTIMx)

9.8.1 Краткие характеристики

- представляет собой декрементирующий 32-х битный счетчик;
- предделитель 16 бит;
- однократный и непрерывный режимы счета.

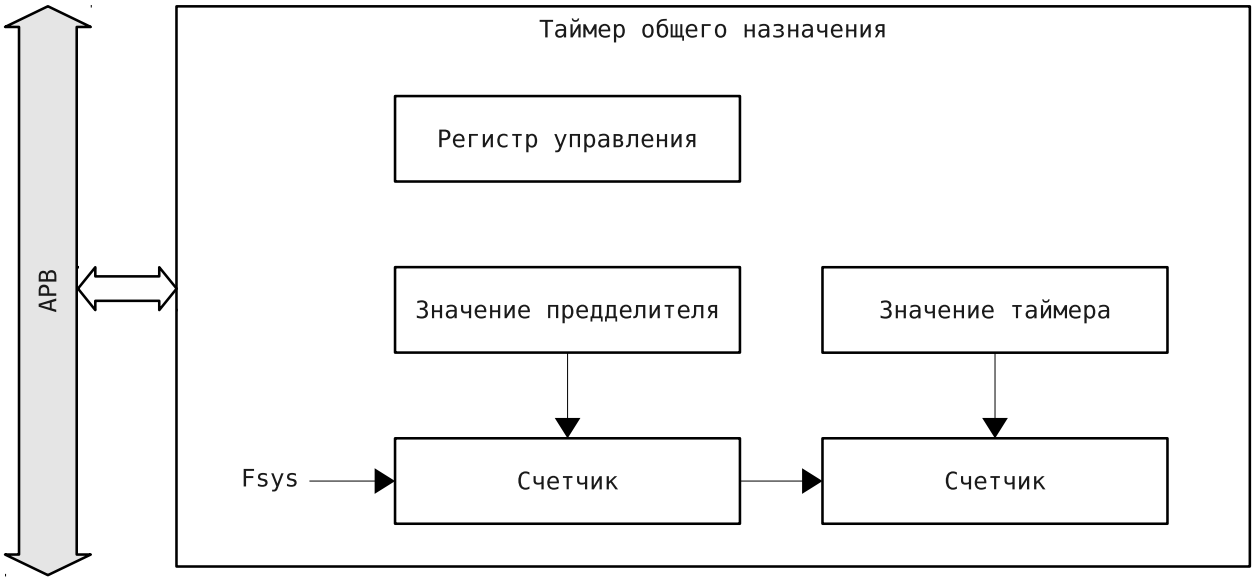

Рис. 42: Блок-схема GPTIMx

Исполнительная часть таймера состоит из предделителя и таймера. Предделитель и таймер представляют собой декрементирующие счетчики с регистрами начальных значений, из которых оно загружается в счетчик после достижения им значение −1. На рис.42 приведена блок-схема GPTIMx.

9.8.2 Алгоритм работы

Таймер начинает счет после установки бита TIMxCR(EN) = ’1’. Внутренний тактовый сигнал после предделителя подается на счетчик таймера. Как только его значение становится −1, формируется запрос обработки прерывания - бит TIMxCR(IP) принимает значение = ’1’, значение TIMxCNTPER(CNTPER) загружается в регистр текущего

Все права защищены © ОАО «Мультиклет» 2010 — 2015

значения счетчика TIMxCNTVAL(CNTVAL). Если установлен непрерывный режим работы счетчика (бит TIMxCR(RS) = '1'), то эти события периодически повторяются. Если установлен однократный режим работы (бит TIMxCR(RS) = '0'), то возобновление счета не происходит, счетчик не декрементируется.

В любой момент таймер может быть перезагружен его начальным значением, при установке бита TIMxCR(LD) = '1'.

Период таймера можно вычислить по следующей формуле:

$$T_{GPTIM} = T_{sys} \cdot (PSCPER + 1) \cdot CNTPER,$$

$$PSCPER \geq 2(GPTIM1, GPTIM3) \quad PSCPER \geq 4(GPTIM0, GPTIM2)$$

Следует особо обратить внимание на то, что значение TIMxPSCPER(PSCPER) не должно быть меньше 2 (или 4).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.8.3 Описание регистров

Базовые адреса регистров GPTIMx:
GPTIM0 - 0xC001 0000
GPTIM1 - 0xC001 0100
GPTIM2 - 0xC011 0000
GPTIM3 - 0xC011 0100
Для получения реального адреса регистра необходимо к базовому (начальному) адресу на шине прибавить смещение адреса регистра. GPTIM0 - 4 канала, GPTIM1 - 2 канала, GPTIM2 - 4 канала, GPTIM3 - 2 канала. Для каждого следующего канала у таймера общего назначения вводятся свои регистры TIMxCNTVAL, TIMxCNTPER, TIMxCR. Например для канала 2 необходимо к адресу каждого из трех регистров добавить смещение 10h(итоговые адреса будут 20h, 24h, 28h).

Регистр	Смещение адреса	Доступ	Описание
TIMxPSCVAL	00h	RW	Регистр текущего значения предделителя
TIMxPSCPER	04h	RW	Регистр начального значения предделителя
TIMxCNTVAL	10h	RW	Регистр текущего значения таймера
TIMxCNTPER	14h	RW	Регистр начального значения таймера
TIMxCR	18h	RW	Регистр управления

TIMxPSCVAL	**Регистр текущего значения предделителя**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																PSCVAL															

16-31 — *зарезервировано*
0-15 PSCVAL текущее значение счетчика предделителя

TIMxPSCPER	**Регистр начального значения предделителя**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																PSCPER															

31-16 — *зарезервировано*
15-0 PSCPER начального значение предделителя (период предделителя)

TIMxCNTVAL	**Регистр текущего значения таймера**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	CNTVAL																															

31-0 CNTVAL текущее значение счетчика таймера

Все права защищены © ОАО «Мультиклет» 2010 — 2015

TIMxCNTPER	**Регистр начального значения таймера**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	CNTPER																															

31-0 CNTPER начального значение счетчика (период счетчика)

TIMxCR	**Регистр управления**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	—																											IP	IE	LD	RS	EN

5-31	—	*зарезервировано*
4	IP	признак сформированного прерывания (’1’ - сформировано, ’0’ - нет запроса прерывания), очищается записью ’1’ в этот бит
3	IE	разрешение формирования прерывания (’1’ - разрешено, ’0’ - запрещено)
2	LD	перезагрузка таймера (’1’ - загрузить TIMxCNTPER(CNTPER) в TIMxCNTVAL(CNTVAL)
1	RS	режим работы таймера (’1’ - непрерывный, ’0’ - однократный)
0	EN	разрешение работы таймера (’1’ - разрешено, ’0’ - запрещено)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9 Контроллер Ethernet(Ethernet0)

9.9.1 Краткие характеристики

- поддерживает скорость 10/100МБит/с
- полнодуплексный, полудуплексный режимы работы;
- прямой канал доступа к ОЗУ;
- поддержка интерфейсов RMII;
- имеет интерфейс MDIO;
- соответствует стандарту IEEE 802.3-2002 и IEEE 802.3Q-2003

Контроллер Ethernet0 состоит из функциональных модулей:

- контроллер прямого доступа к памяти (КПДП)
- MDIO

КПДП используется для передачи данных между внутренней памятью МП и контроллером Ethernet0. Все принятые и сформированные для передачи пакеты данных хранятся во внутренней памяти МП. Приемник и передатчик имеют раздельные КПДП.

MDIO используется для конфигурации и управления внешним преобразователем среды (PHY).

Ethernet0 поддерживает следующие стандарты: IEEE 802.3-2002 и IEEE 802.3Q- 2003(опционально, см. таблицу комплектации МП). Контроллер не поддерживает пакеты типа 0x8808, они не будут приниматься.

Приемник и передатчик Ethernet0 связаны с внешним преобразователем среды по интерфейсу Reduced Media Independent Interface (RMII).

Размер таблиц дескрипторов приемника и передатчика - 1КБ.

9.9.2 Тактирование

Приемник и передатчик Ethernet0 тактируются внешним преобразователем среды, для приемника и передатчика независимые тактовые сигналы и являются частью интерфейса RMII. Внутренние структуры управления тактируются системной частотой Fsys. Контроллером поддерживаются полудуплексные и полнодуплексные режимы работы, которые могут работать на скоростях передачи 10 и 100МБит/с. Минимальная Fsys

Все права защищены © ОАО «Мультиклет» 2010 — 2015

необходимая для корректной работы на скорости 10МБит/с - 2.5МГц, для 100МБит/с - 18МГц. Значения Fsys ниже требуемых может привести к потере пакетов.

9.9.3 КПДП передатчика

Передатчик использует дескрипторы размещенные во внутренней памяти МП. Нельзя изменять дескриптор во время передачи.

9.9.3.1 Установка дескриптора

Дескриптор имеет указание на адрес по которому размещается блок данных и размер блока. Так же там содержится управляющая информация. Адрес блока данных должен быть выровнен на 32 бита. Дескриптор не должен изменяться пока контроллером Ethernet0 не будет установлен TD0(EN) = '0'.

TD0	**Ethernet0 дескриптор, часть 0 (смещение адреса 0x0)**																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	—																AL	UE	IE	WR	EN	LENGTH										

31-16	—	*зарезервировано*
15	AL	пакет не передан, т.к. число попыток превысило максимальное
14	UE	пакет передан некорректно, т.к. FIFO был заполнен не полностью
13	IE	разрешение прерывания по завершении передачи пакета, вне зависимости от того был ли он передан корректно или нет, при условии, что установлен бит ETHxCR(TI)
12	WR	разрешение указателю таблицы дескрипторов принять значение 0 после передачи данного пакета ('1' - разрешено, '0' - запрещено). Если WR=0, то указатель таблицы дескрипторов инкрементируется на 8 и примет значение 0 только после того, как достигнет конца таблицы дескрипторов.
11	EN	разрешение операций с дескриптором ('1' - разрешено, '0' - запрещено)
10-0	LENGTH	размер блока данных для передачи в байтах

TD1	**Ethernet0 дескриптор, часть 1 (смещение адреса 0x4)**																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	ADDRESS																														—	

31-2	ADDRESS	указатель на начальный адрес в памяти, где размещены данные для передачи
1-0	—	*зарезервировано*

9.9.3.2 Подготовка данных для передачи

Весь пакет данных кроме CRC должен быть размещен в памяти, начальный адрес должен быть указан в дескрипторе. Длинна пакета, указанная в дескрипторе, не должна превышать максимально возможную 1514 байт, иначе пакет не будет передан.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9.3.3 Передача данных

Для начала передачи данных, необходимо установить указатель на адрес таблицы дескрипторов и установить бит TD0(EN) в соответствующем дескрипторе. Адрес таблицы дескрипторов должен быть выровнен на 1КБ. Биты 31..10 содержат базовый адрес таблицы дескрипторов, 9..3 - указатель на конкретный дескриптор (в байтах). Указатель будет установлен в 0 как только он превысит значение 1КБ. В случае если в каком-то дескрипторе установлен бит TD0(WR)=’1’, то указатель на дескриптор примет значение 0 когда дойдет до этого дескриптора.

После установки адреса необходимо разрешить передачу данных CR(TX_EN)=’1’. Это означает, что все дескрипторы готовы, можно начать передачу данных.

9.9.3.4 Работа с дескриптором после окончания передачи данных

Соответствующие статусные биты будут записаны в TD0 после завершения передачи пакета, описанного дескриптором. Пакет считается переданным успешно, если TD0(UE) и TD0(AL) имеют значение ’0’. TD0(UE)=’1’, если во время передачи FIFO передатчика оказывался пустым. TD0(AL)=’1’ устанавливается, если во время передачи возникло коллизий больше, чем предусмотрено протоколом. Все остальные биты TD0 устанавливаются = ’0’ после завершения передачи пакета. TD1 остается без изменений. Бит TD0(EN) может быть использован как индикатор того, что дескриптор готов к использованию, т.к. контроллер Ethernetx автоматически устанавливает его в ’0’ после окончания передачи пакета. Помимо отображения информации в дескрипторе, в контроллер так же имеются биты статуса передатчика: ETHxST(TE) - ошибка передачи, ETHxST(TI) - запрос обработки прерывания, устанавливается каждый раз как передача была завершена успешно. ETHxST(TA) - ошибка обмена данными через периферийную шину. В этом случае передатчик будет остановлен.

9.9.4 КПДП приемника

Приемник использует дескрипторы размещенные во внутренней памяти МП. КПДП приемника предназначен для приема данных по сети Ethernet.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9.4.1 Установка дескриптора

Дескриптор имеет указание на адрес по которому размещается блок данных и размер блока. Так же там содержится управляющая информация. Адрес блока данных должен быть выровнен на 32 бита.

TD0	**Ethernet0 дескриптор, часть 0 (смещение адреса 0x0)**																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	—					MC	—							LE	OE	CE	FT	AE	IE	WR	EN	LENGTH										

31-27	—	*зарезервировано*
26	MC	адрес назначения пакета является групповым (не транслируется)
25-19	—	*зарезервировано*
18	LE	ошибка, значение поля "длина пакета"не соответствует текущему числу принятых байтов
17	OE	ошибка, кадр был принят некорректно из-за переполнения буфера приемника
16	CE	ошибка CRC в кадре
15	FT	ошибка, принят кадр больше максимального размера, лишняя часть отброшена
14	AE	ошибка, принято нечётное количество полубайтов
13	IE	разрешение прерываний (’1’ - разрешено, ’0’ - запрещено). Прерывания буду вырабатываться после приёма пакета (бит ETHxCR(RI) должен быть в ’1’), Прерывания будут вырабатываться вне зависимости от того завершился ли приём пакета успешно или произошла ошибка.
12	WR	разрешение указателю таблицы дескрипторов принять значение 0 после передачи данного пакета (’1’ - разрешено, ’0’ - запрещено). Если WR=0, то указатель таблицы дескрипторов инкрементируется на 8 и примет значение 0 только после того, как достигнет конца таблицы дескрипторов.
11	EN	разрешение операций с дескриптором (поле устанавливается последним) (’1’ - разрешено, ’0’ - запрещено)
10-0	LENGTH	размер блока данных для приема в байтах

TD1	**Ethernet0 дескриптор, часть 1 (смещение адреса 0x4)**																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	ADDRESS																														—	

31-2	ADDRESS	указатель на начальный адрес в памяти, где будут размещены принятые данные
1-0	—	*зарезервировано*

9.9.4.2 Прием данных

Для начала приема данных, необходимо установить указатель на адрес таблицы дескрипторов и установить бит TD0(EN) в соответствующем дескрипторе. Адрес таблицы дескрипторов должен быть выровнен на 1КБ. Биты 31..10 содержат базовый адрес таблицы дескрипторов, 9..3 - указатель на конкретный дескриптор (в байтах). Указатель будет установлен в 0 как только он превысит значение 1КБ. В случае если в каком-то дескрипторе установлен бит TD0(WR)=’1’, то указатель на дескриптор примет значение 0 когда дойдет до этого дескриптора.

После установки адреса необходимо разрешить приём данных ETHxCR(RE)=’1’. Это означает, что все дескрипторы готовы, можно начать передачу данных.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9.4.3 Работа с дескриптором после окончания приема данных

После завершения приема, бит TD0(EN) имеет значение ’0’. Биты TD0(WR) и TD0(IE) также имеют значение ’0’. Количество принятых байт отображается в TD0(LENGTH). Части Ethernet кадра содержат адрес назначения, адрес источника, тип и поля данных. Биты 17..14 в TD0 сигнализируют об ошибках приёма. После успешного приёма все 4 бита должны иметь значение ’0’. Пакет меньше минимального размера в 64 байта не является принятым и отбрасывается. Текущий регистр приёма запрещается изменять до приёма первого пакета с допустимым размером. Бит ETHxST(TS) сигнализирует об ошибке приёма пакета меньше минимально размера. Бит ETHxST(IA) сигнализирует о приёме пакета с недопустимым MAK адресом. Бит TD0(FT) сигнализирует о приёме пакета больше максимально допустимого размера. Поле TDO(LENGTH) не гарантирует правильного приёма данных. Количество пустых байт до максимального размера пакета после слова, содержащего последний байт, записывается в память.

9.9.4.4 Ошибки приема на шине AHB

Если произойдет ошибка на шине AHB в момент чтения дескриптора или записи данных, то бит ETHxST(RA) будет установлен в "1"и приемник отключится. Текущий прием будет отклонен. Для включения приемника необходимо будет установить бит ETHxCR(RE) в "1".

9.9.4.5 Допустимые MAC адреса

По умолчанию контроллер Ethernet принимает пакеты с любым одноадресным MAC или широковещательным. Многоадресный прием пакетов также может быть активирован. MAC адрес устанавливается в регистрах ETHxMACMSB и ETHxMACLSB. В случае использования многоадресной рассылки решение о приеме пакете принимается следующим образом: 1) считается ethernet crc от MAC адреса назначения пакета; 2) 6 младших бит полученного значения используются в качестве индекса бита в регистрах ETHxHASHMSB и ETHxHASHLSB; 3) если значение бита определенного на предыдущем шаге равно "1 то пакет принимается, в противном случае не принимается.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9.5 MDIO интерфейс

MDIO интерфейс обеспечивает доступ к регистрам настройки и статуса внешней микросхемы PHY через двухпроводной интерфейс. MDIO интерфейс может быть использован для доступа от 1 до 32 PHY, содержащих от 1 до 32 16-ти битных регистров. Чтение из PHY происходит путем установки адреса PHY и адреса считываемого регистра и установкой бита чтения в регистре ETHxMDIO(RD) в "1". Операция чтения завершится после того как бит ETHxMDIO(BU) станет равен "0". Если операция прошла успешно, то ETHxMDIO(LF) равен "0" и поле данных будет актуальным, в противном случае данный бит будет равен "1" и поле данных будет неопределенным значением. Операция записи осуществляется записью в ETHxMDIO 16-ти бит данных, установкой адреса PHY и адреса записываемого регистра и установкой ETHxMDIO(WR) в "1". Операция завершается после того как ETHxMDIO(BU) стал равен "0"и считается успешной, если ETHxMDIO(LF) равен "0".

9.9.5.1 Прерывания PHY

Контроллер Ethernet поддерживает прерывания от PHY. Прерывания вырабатываются при установке "0" на линии MDINT. Изменение состояния PHY отображается в ETHxST(PS) установкой в "1"каждый раз при детектировании событий на этой линии.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.9.6 Описание регистров

Базовый адрес Ethernet0 - 0xC000 5000

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
ETHxCR	00h	RW	Регистр установок конфигурации
ETHxST	04h	RW	Регистр состояния
ETHxMACMSB	08h	RW	MAC адрес старшая часть
ETHxMACLSB	0Ch	RW	MAC адрес младшая часть
ETHxMDIO	10h	RW	Регистр MDIO
ETHxTDP	14h	RW	Указатель дескриптора передачи
ETHxRDP	18h	RW	Указатель дескриптора приема
ETHxHASHMSB	20h	RW	Хэш таблица, старшая часть
ETHxHASHLSB	24h	RW	Хэш таблица, младшая часть

ETHxCR	**Регистр управления**																											
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0						0	0													0	0	0		0	0	0	0	0	0	0	0
Описание	—						MC	—													ME	PI	-		SP	RS	PM	FD	RI	TI	RE	TE

26-31	—	*зарезервировано*
25	MC	статус многоадресного режима (’1’ - разрешено, ’0’ - запрещено)
12-24	—	*зарезервировано*
11	ME	разрешить прием многоадресных пакетов (’1’ - разрешено, ’0’ - запрещено)
10	PI	разрешить прерывания при изменении статуса внешнего PHY (’1’ - разрешено, ’0’ - запрещено)
8-9	—	*зарезервировано*
7	SP	скорость (’1’ - 100 Мбит/с, ’0’ - 10 Мбит/с)
6	RS	сброс, бит будет очищен после окончания сброса контроллера, никакие другие операции не следует производить с контроллером пока бит равен ’1’ (’1’ - инициировать сброс контроллера Ethernet)
5	PM	прием всех пакетов несмотря на адрес устройства назначения (’1’ - разрешено, ’0’ - запрещено)
4	FD	полнодуплексный режим (’1’ - разрешено, ’0’ - запрещено)
3	RI	разрешение прерываний приемника (’1’ - разрешены, ’0’ - запрещены)
2	TI	разрешение прерываний передатчика (’1’ - разрешены, ’0’ - запрещены)
1	RE	разрешение приема, бит автоматически сбрасывается в ’0’ после завершения приема пакета. Устанавливать бит следует только после записи дескриптора приема (’1’ - разрешено, ’0’ - запрещено)
0	TE	разрешение передачи, бит автоматически сбрасывается в ’0’ после завершения передачи пакета. Устанавливать бит следует только после записи дескриптора передачи (’1’ - разрешено, ’0’ - запрещено)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ETHxST	Регистр состояния																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																							0	0	0	0	0	0	0	0	0
Описание	—																							PS	IA	TS	TA	RA	TI	RI	TE	RE

Для всех разрядов регистра: ('1' - наличие признака, '0' - отсутствие признака)

9-31	—	*зарезервировано*
8	PS	изменения статуса PHY
7	IA	принят пакет с адресом не соответствующим MAC. Очищается записью '1'
6	TS	принят пакет данных меньше минимального размера. Очищается записью '1'
5	TA	ошибка передатчика при работе по каналу DMA. Коллизии на системной шине или при доступе к памяти. Очищается записью '1'
4	RA	ошибка приемника при работе по каналу DMA. Коллизии на системной шине или при доступе к памяти. Очищается записью '1'
3	TI	пакет передан без ошибок. Очищается записью '1'
2	RI	пакет принят без ошибок. Очищается записью '1'
1	TE	передача пакета прервалась ошибкой. Очищается записью '1'
0	RE	прием пакета прервался ошибкой. Очищается записью '1'

ETHxMACMSB	MAC адрес, старшая часть																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																0															
Описание	—																47..32 биты MAC															

16-31	—	*зарезервировано*
0-15	MACMSB	два самых старших байта MAC адреса

ETHxMACLSB	MAC адрес, младшая часть																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	31..0 биты MAC																															

16-31	—	*зарезервировано*
0-15	MACLSB	младшие байты MAC адреса

ETHxMDIO	Регистр MDIO																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—																—					—					-	-	-	-	-	-
Описание	DATA																PHYADDR					REGADDR					-	NV	BU	LF	RD	WR

31-16	DATA	поле данных, которое может содержать принятые данные или данные на передачу
15-11	PHYADDR	адрес PHY для доступа на чтение или запись
10-6	REGADDR	адрес регистра для которого может быть осуществлена операция чтения или записи
5	—	*зарезервировано*
4	NV	после завершения операции (ETHxMDIO(BU) = 0) данный бит показывает корректность данных (1 - данные некорректны, 0 - данные корректны)
3	BU	когда выполняется операция данный бит устанавливается в "1". После того как операция выполнена и есть функциональная связь данный бит будет установлен в "0"
2	LF	после завершения операции (ETHxMDIO(BU) = 0) данный бит устанавливается в "1" если отсутствует функциональная связь
1	RD	операция чтения данных
0	WR	операция записи данных

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ETHxTDP	Указатель дескриптора передачи																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																						0							0		
Описание	BASEADDR																						DESCPNT							-		

10-31 BASEADDR базовый адрес дескрипторов. Необходимо выставлять адрес 0xE0200000 + реальный адрес в ПД, записываются только старшие биты 31..10

3-9 DESCPNT указатель на дескриптор, автоматически инкрементируется при получении нового пакета данных

0-2 — *зарезервировано*

ETHxRDP	Указатель дескриптора приема																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																						0							0		
Описание	BASEADDR																						DESCPNT							-		

10-31 BASEADDR базовый адрес дескрипторов. Необходимо выставлять адрес 0xE0200000 + реальный адрес в ПД, записываются только старшие биты 31..10

3-9 DESCPNT указатель на дескриптор, автоматически инкрементируется при получении нового пакета данных

0-2 — *зарезервировано*

ETHxHASHMSB	Хэш таблица, старшая часть																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	Hash table 64..32																															

31-0 HASHMSB Старшая часть хэш таблицы для многоадресных пакетов.

ETHxHASHLSB	Хэш таблица, младшая часть																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																															
Описание	Hash table 31..0																															

31-0 HASHLSB Младшая часть хэш таблицы для многоадресных пакетов.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.10 Контроллер USB(USBx)

9.10.1 Краткие характеристики

- 4 канала «IN»;
- 4 канала «OUT»;
- поддержка передач «Control», «Bulk», «Interrupt» и «Isochronous»;
- требуется внешний приёмопередатчик, подключаемый по интерфейсу ULPI;
- обмен данными с системной памятью по шине AHB в режиме «Master»;
- внутренние буферы FIFO на приём и передачу снижают нагрузку на системную шину.

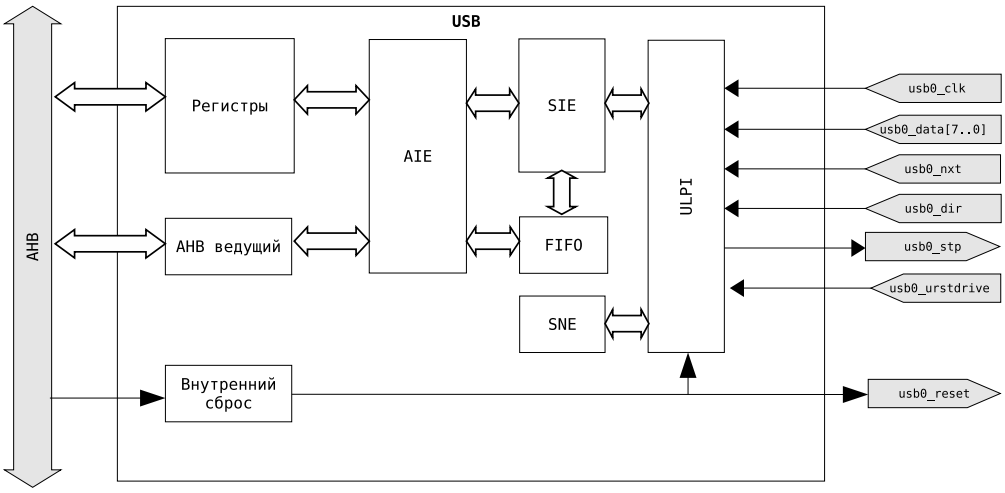

Рис. 43: Блок-схема USBx

Встроенный контроллер USB 2.0 поддерживает только функцию «Device» и реализует обмен в режиме Full Speed (12 Мбит/с) и High Speed (480 Мбит/с) с использованием внешнего приёмопередатчика, подключаемого по интерфейсу ULPI.

Для связи с шиной USB USBx использует внешний приёмопередатчик, подключаемый по ULPI (UTMI+ Low Pin Interface). Промежуточное хранение данных обеспечивают внутренние буферы FIFO общим объёмом 8 Кбайт (по 1 Кбайт на каждый канал). Принимаемые или передаваемые данные размещаются в буферах, находящихся в системной памяти. Информация о буферах хранится в таблице дескрипторов данных, также размещающейся в системной памяти. Доступ к памяти осуществляется по шине AHB в режиме «ведущего».

Все права защищены © ОАО «Мультиклет» 2010 — 2015

При инициализации блока USB в ULPI передаётся последовательность команд, устанавливающих внешний приёмопередатчик в исходное состояние. Поскольку USBx инициализируется автоматически при запуске МК, а выводы МК после сброса устанавливаются в режим GPIO, в USBx введена задержка его инициализации относительно системного сброса.

Таким образом, если программа будет использовать USB, выводы МК, использующиеся для подключения ULPI, необходимо переключить на альтернативную функцию (с помощью регистра GPIOxBPS) сразу после запуска программы (т.е. выхода МК из сброса).

Передаваемые и принимаемые блоком USB данные хранятся в системной памяти.

Допустимо изменение следующих параметров:

- скорость работы USB;
- полярность USB;
- USB адрес(по умолчанию 0);
- установка глобального разрешения работы конечной точки;

9.10.2 Обзор системы

SNE (Speed Negotiation Engine) детектирует подключение мониторингом сигнала VBUS на USB соединителе. После обнаружения устойчивых 5 Вольт SNE ждет сброса и затем начинает высокоскоростное согласование (High-speed negotiation). После завершения согласования по скорости и процедуры сброса выбранный режим скорости работы (high-speed или full-speed) устанавливается в SIE (Serial Interface Engine), который начинает свою работу. SNE также детектирует и обрабатывает приостановку(suspend) и операции с режимами работы.

SIE включается после уведомления от SNE, что процедура сброса завершена. Затем ожидает прибывающих пакетов и обрабатывает их по стандарту USB 2.0. Данные записываются во внутренние буфера принадлежащие конечной точке приемнику.

AIE (AHB Interface Engine) предназначен для передачи данных по USB от внутренних буферов конечной точки на AHB шину, используя дескриптор ПДП через интерфейс AHB "ведущий".

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.10.3 PHY интерфейс

Контроллер поддерживает в качестве внешнего уровня PHY микросхемы с интерфейсом ULPI(UTMI+ Low Pin Interface).

9.10.4 Speed Negotiation Engine(SNE)

9.10.5 Serial Interface Engine(SIE)

9.10.6 Буферы конечной точки

Каждая конечная точка содержит по два буфера, в которые происходит запись пакетов. Контроллер автоматически управляет ими, когда пакет принимается или передается так, что данные из одного буфера могут быть переданы по шине AHB, пока новый пакет принимается или передается по USB от(в) второго буфера.

Состояние буферов влияет на согласование посылаемое хосту в конце транзакции. В high-speed режиме пакетные (BULK) конечные точки на выход и управляющие (CONTROL) конечные точки на выход, которые находятся не в состоянии SETUP поддерживают PING протокол. Это означает, что в конце транзакции на выход к одной из этих конечных точек, устройство должно вернуть ACK, если возможен прием текущих данных и имеется место для другого пакета. Для USB контроллера это выполняется, когда второй буфер конечной точки пуст по завершению транзакции.

Если второй буфер не пуст, то взамен будет отправлен NYET. Если текущие данные не могут быть приняты (оба буфера не пусты когда пришел пакет), то будет отправлен NAK.

Для других типов конечных точек в высокоскоростном режиме и всех типов конечных точек в полноскоростном режиме вернется ACK, если данные приняты и не был получен NACK. Буфер конечной точки может быть сконфигурирован больше максимального значения по количеству данных для данной конечной точки. Для конечных точек типа IN при попытке записи данных объем, которых превышает размер буфера, будет сгенерировано несколько пакетов. Для конечных точек типа OUT большие буферы используются только для высоконагруженных конечных точек, где может произойти более одной транзакции на кадр. В этом случае данные из все пакетов в течение одного микрокадра записываются поочередно как прибыли в одиночный буфер и передаются через AHB интерфейс. Все не высокозагруженные конечные точки также записывают один пакет в буфер.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.10.7 AMBA Interface Engine(AIE)

9.10.8 Синхронизация

В контроллере имеется две области тактирования: AHB и USB. AHB область тактирования работает с такой же частотой, как и шина AHB, тогда как USB область тактируется от ULPI. Граница между областями тактирования проходим между AIE, SIE и буферами конечных точек.

9.10.9 Формирование сброса

Основной сброс (AMBA сброс) сбрасывает AHB регистры, синхронизацию между USB и AHB областями тактирования, USB PHY и USB SIE регистрами. Регистры конечной точки в SIE, согласно USB протоколу, сбрасываются когда получен USB сброс.

9.10.10 ПДП операции

Каждая конечная точка типа IN и каждая конечная точка типа OUT имеет выделенный ПДП канал, который передают данные из(в) внутренних буферов конечной точки, используя дескриптор автономного ПДП. Каждое направление (IN и OUT) имеет свой блок ПДП, который запрашивает AHB ведущий интерфейс в соответствии с направлением. Также каждая конечная точка одного направления соперничает с остальными конечными точками такого же напралвения за использование ПДП блока. Арбитраж происходит по кругу для конечных точек, которые включены и имеют данные на прием или передачу.

Операции в обоих направлениях идентичны и соответствующие свойства описаны в данной секции, тогда как различия описываются в двух следующих подсекциях ниже.

ПДП операции основаны на связном списке в виде дескрипторов, размещенных в памяти. Каждая конечная точка имеет свой связный список. Первое слово дескриптора - управляющее и содержит бит разрешения работы, который определяет активен ли дескриптор или нет, и другие управляющие биты. Следующее слово - указатель на буфер в памяти из которого должны быть считаны данные или записаны для текущего дескриптора. Последнее слово - указатель на область памяти, содержащей следующий дескриптор. Бит управляющего слова определяет действителен ли указатель на следующий дескриптор. Если дескриптор не действителен происходит остановка после обработки текущего дескриптора и ПДП канал отключается.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

ПДП операции начинаются после первой установки списка дескрипторов в памяти и записи указателя на первый дескриптор в регистр указателя дескриптора конечной точки и установки бита доступности дескриптора. Регистр указателя дескриптора обновляется после того, как список дескрипторов пройден и может быть прочитан через AHB интерфейс. Когда список заканчивается дескриптором с битом доступности следующего дескриптора установленным в '0' запрещается вносить корректировки в список, до того как контроллер закончит обработку всего списка и отключит канал. При не соблюдении данной рекомендации поведение контроллера будет неопределенным.

Другой способ использования списка связности это установка у каждого дескриптора бита доступности, но при этом убедиться, что последний дескриптор выключен. Данный способ позволяет добавлять новые дескрипторы и включать их "на лету". На рис. 44 изображен пример структуры ПДП списка связности дескриптора в памяти:

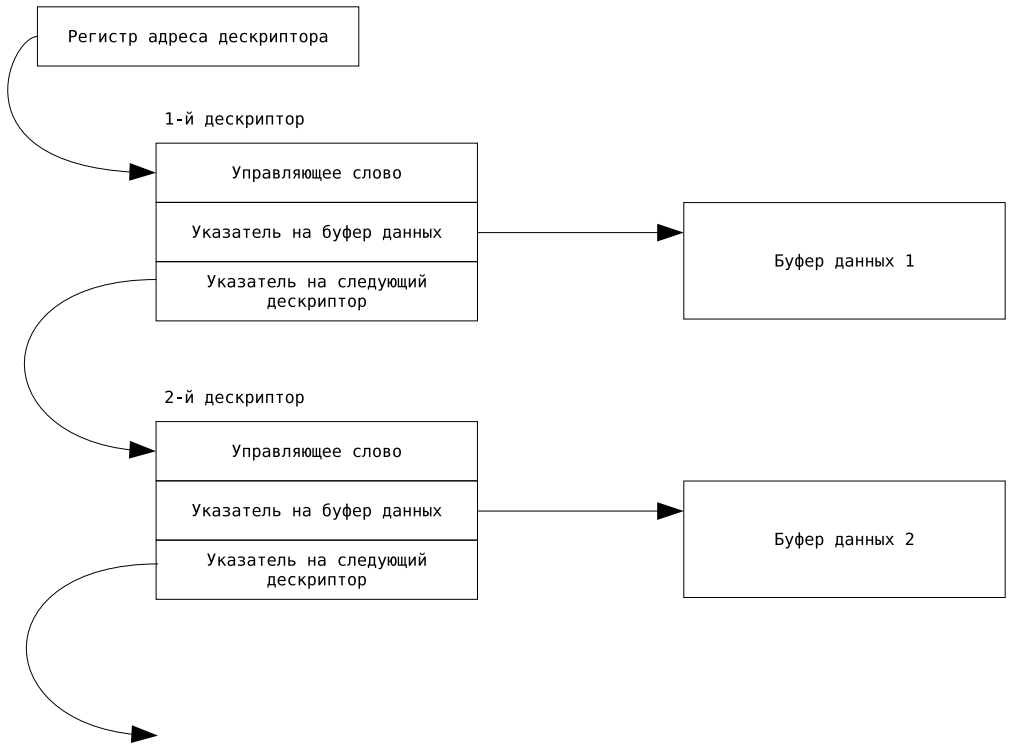

Рис. 44: Структура ПДП списка связности дескриптора в памяти

9.10.11 Конечные точки типа OUT

ПДП операции для конечных точек типа OUT формируются в соответствии с основным описанием представленным ранее. Небольшие отличия заключаются в индивидуальных битах и значении поля размер. Содержание всех слов дескриптора представлено в таблицах ниже.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Когда разрешена работа дескриптора он выбирается контроллером, если бит доступности дескриптора установлен в "1"и как только буфер для соответствующей конечной точки содержит данные, полученные по USB, они будут записаны в память, начиная с адреса указанного в слове дескриптора - указатель на буфер данных. Содержимое одиночного внутреннего буфера всегда записывается в одиночный буфер дескриптора. Это всегда соответствует одиночному пакету USB, за исключением высоконагруженных изохронных и конечных точек по прерываниям. Количество записанных байт указывается в поле размер первого слова дескриптора, когда запись завершена, что отображается путем установки бита разрешения работы в "0". Затем SETUP бит также будет действителен. Когда бит разрешения работы установлен в "0"область памяти может быть использована снова.

Прерывание формируются (если разрешены) после заврешения записи в память. Конечная точка также может быть настроена на формирование прерываний немедленно после приема пакета во внутренние буферы. Это не может быть настроено на конкретный пакет, т.к. контроллер не может ассоциировать полученный пакет с дескриптором заранее. Прерывания разрешаются в регистре управления конечной точки.

Когда данные выбраны из внутреннего буфера он может использоваться SIE снова для приема нового пакета.

OUT0	**USB OUT дескриптор, часть 0 (смещение адреса 0x0)**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	—														SE	-	IE	NX	EN	LENGTH												

31-18	—	*зарезервировано*
17	SE	данные были приняты из пакета SETUP взамен OUT.
16	—	*зарезервировано*
15	IE	разрешение прерываний (’1’ - разрешено, ’0’ - запрещено). Прерывания буду вырабатываться после того как пакет для данного дескриптора считан во внутренние буферы и обработан через SIE. Но это не означает, что пакет полностью передан.
14	NX	указатель на следующий дескриптор действителен (’1’ - действителен, ’0’ - не действителен).
13	EN	разрешение операций с дескриптором (поле устанавливается последним) (’1’ - разрешено, ’0’ - запрещено)
12-0	LENGTH	размер блока принятых данных (данное поле действительно после выставления контроллером бита EN в ’0’)

OUT1	**USB OUT дескриптор, часть 1 (смещение адреса 0x4)**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	ADDRESS																														—	

31-2	ADDRESS	указатель на начальный адрес в памяти, где будут размещены принятые данные
1-0	—	*зарезервировано*

Все права защищены © ОАО «Мультиклет» 2010 — 2015

OUT2	USB OUT дескриптор, часть 2 (смещение адреса 0x8)																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	NDP																														—	

31-2	NDP	указатель на следующий дескриптор
1-0	—	*зарезервировано*

9.10.12 Конечные точки типа IN

ПДП операции для конечных точек типа IN формируются в соответствии с основным описанием представленным ранее. Небольшие отличия заключаются в индивидуальных битах и значении поля размер. Содержание всех слов дескриптора представлено в таблицах ниже.

Когда разрешена работа дескриптора он выбирается контроллером, если бит доступности дескриптора установлен в "1"контроллер начинает обрабатывать дескриптор и выбирает количество байт, указанных в поле LENGTH, во внутренние буферы, принадлежащие конечной точке, как только бит доступности дескриптора установлен в "1".

Прерывание формируются (если разрешены) после заврешения записи во внутренние буферы и установки статусных битов. Пакет при этом может быть еще не передан по USB. Доступны отдельные прерывания, которые формируются когда пакет данных полностью передан по USB. Данные прерывания необходимо разрешать в регистре управления конечной точки а также в дескрипторе, используя бит PI для каждого пакета при передаче которого необходимо формировать прерывания.

Дескриптор с нулевым полем LENGTH будет преобразован в пакет с нулевой длиной, тогда как пакет с длингой большей максимально допустимого значения будет преобразован в несколько пакетов, каждый из которых, кроме последнего, будет иметь максимальную длину. Последняя транзакция может быть меньше или равна максимальному значению. Если поле LENGTH будет больше, чем внутренний буфер, то данные не будут записаны во внутренний буфер и установится бит ошибки.

В случае установки MORE бита данные из текущего дескриптора записываются во внутренний буфер и затем осуществляется переход к следующему дескриптору без разрешения работы буфера на передачу. Данные следующего дескриптора также будут считаны в текущий буфер и данная ситуация продолжится до тех пор пока не встретится дескриптор в котором не установлен бит MORE.

Если итоговое количество байт больше, чем внутренний буфер, то пакет не посылается(данные из внутреннего буфера выбрасываются) и устанавливается в "1"бит ML для

Все права защищены © ОАО «Мультиклет» 2010 — 2015

последнего дескриптора. Затем выборка начинается снова.

IN0	**USB IN дескриптор, часть 0 (смещение адреса 0x0)**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	—														SE	-	IE	NX	EN	LENGTH												

31-19	—	*зарезервировано*
18	MO	данные из следующего дескриптора должны быть считаны в текущий внутренний буфер (’1’ - разрешено, ’0’ - запрещено)
17	PI	разрешение формирования прерываний, когда пакет полностью передан по USB (’1’ - разрешено, ’0’ - запрещено)
16	ML	индикация установкой данного бита в "1"попытки передачи данных размером больше буфера
15	IE	разрешение прерываний (’1’ - разрешено, ’0’ - запрещено). Прерывания буду вырабатываться после того как пакет для данного дескриптора считан во внутренние буферы и обработан через SIE. Но это не означает, что пакет полностью передан.
14	NX	указатель на следующий дескриптор действителен (’1’ - действителен, ’0’ - не действителен).
13	EN	разрешение операций с дескриптором (поле устанавливается последним) (’1’ - разрешено, ’0’ - запрещено)
12-0	LENGTH	размер блока данных на передачу

IN1	**USB IN дескриптор, часть 1 (смещение адреса 0x4)**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	ADDRESS																														—	

31-2	ADDRESS	указатель на начальный адрес в памяти, где будут размещены данные на передачу
1-0	—	*зарезервировано*

IN2	**USB IN дескриптор, часть 2 (смещение адреса 0x8)**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	NDP																														—	

31-2	NDP	указатель на следующий дескриптор
1-0	—	*зарезервировано*

9.10.13 Конечные точки

9.10.14 Управляющие конечные точки

9.10.15 Пакетные конечные точки

9.10.16 Конечные точки по прерываниям

9.10.17 Изохронные конечные точки

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.10.18 Описание регистров

Базовый адрес USB - 0xFFFF 0000.

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра. Для конечных точек 1,2,3 необходимо использовать смещение для каждого регистра на 10h, 20h, 30h соответсвтенно. Например, для конечной точки 2 регистр USBxEN2_CR_OUT будет иметь смещение адреса 20h, а регистр USBxEN2_ST_IN смещение 12Ch.

Регистр	**Смещение адреса**	**Доступ**	**Описание**
USBxEP0_OUT_CR	00h	RW	Регистр управления конечной точки 0 на выход
USBxEP0_OUT_DMA_CR	04h	RW	Регистр управления ПДП конечной точки 0 на выход
USBxEP0_OUT_DM_ADDR	08h	RW	Регистр адреса ПДП конечной точки 0 на выход
USBxEP0_IN_CR	100h	RW	Регистр управления конечной точки 0 на вход
USBxEP0_IN_DMA_CR	104h	RW	Регистр управления ПДП конечной точки 0 на вход
USBxEP0_IN_DMA_ADDR	108h	RW	Регистр адреса ПДП конечной точки 0 на вход
USBxCR	200h	RW	Регистр управления
USBxST	204h	RW	Регистр состояния

Все права защищены © ОАО «Мультиклет» 2010 — 2015

EP0_OUT_CR	Регистр управления конечной точки 0 на выход																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0											0	0	0	0											0		0		0	0	0
Описание	—											PI	CB	CS	MAXPL											NT		TT		EH	ED	EV

31-21	—	*зарезервировано*
20	PI	разрешение прерываний на каждый полученный пакет по USB для конечной точки(пакет был записан во внутренний буфер) ('1' – разрешено, '0' - запрещено).
19	—	*зарезервировано*
18	CS	разрешение возврата STALL для этапов с данными и состоянием в управляющих передачах('1' – разрешено, '0' - запрещено). Автоматически бит очищается при приеме следующего маркера(token) SETUP. Используется для конечных точек, сконфигурированных как управляющие.
17-7	MAXPL	установка максимального размера одиночного USB пакета посылаемого или принимаемого для данной конечной точки. Максимально возможный размер одиночного пакета равен 1024 байта.
6-5	NT	установка количества дополнительных транзакций на микрокадр для высокоскоростных конечных точек и на кадр для полноскоростных конечных точек. Данные биты предназначены только для изохронных конечных точек.
4-3	TT	установка типа работы конечной точки: "00 управляющая(CTRL), "01 изохронная(ISOCH), "10 пакетная(BULK), "11 по прерываниям(INTERRUPT). Только конечные точки типа выход (OUT) должны устанавливаться в тип CTRL. В этом случае конечная точка на вход (IN) с этим же номером будет автоматически задействована.
2	EH	остановка конечной точки ('1' – остановка включена, '0' - остановка выключена). Если данный бит установлен в "1 то все трансферы к данной точке будут получать ответ STALL.
1	ED	отключение конечной точки ('1' – отключено, '0' - не отключено). Если данный бит установлен в "1 то все трансферы к данной точке будут получать ответ NAK.
0	EV	разрешение работы конечной точки ('1' – разрешено, '0' - запрещено). Если работа запрещена, то все трансферы к данной точке буду проигнорированы без согласования.

EP0_OUT_DM_CR	Регистр управления ПДП конечной точки 0 на выход																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	-																					-	—						0	-	-	-
Описание	—																					AE	—						AD	AI	IE	DA

31-11	—	*зарезервировано*
10	AE	Бит сигнализации ошибки на шине AHB для данной конечной точки('0' – нет ошибки, '1' – есть ошибка).
9-4	—	*зарезервировано*
3	AD	Отключение обработки дескрипторов(установка бита DA в '0') и отклонение текущей передачи(приема) ПДП('0' – запрещено, '1' – разрешено).
2	AI	Разрешение формирования прерываний, когда происходят ошибки на шине AHB.
1	IE	Разрешение прерываний ПДП('0' – запрещено, '1' – разрешено). Каждый раз при приеме или передаче данных через дескриптор с установленным битом разрешения прерываний, прерывания будут формироваться когда установлен текущий бит в '1'.
0	DA	Бит индикации включения(готовности) одного или нескольких дескрипторов ('0' – не готовы, '1' – готовы).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

EP0_OUT_DM_ADDR	Регистр адреса ПДП конечной точки 0 на выход																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																														-	
Описание	DESCADDR																														-	

31-2 DESCADDR Адрес следующего дескриптора

0-1 — *зарезервировано*

EP0_IN_CR	Регистр управления конечной точки 0 на вход																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0											0	0	0	0											0		0		0	0	0
Описание	—											PI	CB	CS	MAXPL											NT		TT		EH	ED	EV

31-21 — *зарезервировано*

20 PI разрешение прерываний на каждый полученный пакет по USB для конечной точки(пакет был записан во внутренний буфер) ('1' – разрешено, '0' - запрещено).

19 — *зарезервировано*

18 CS разрешение возврата STALL для этапов с данными и состоянием в управляющих передачах('1' – разрешено, '0' - запрещено). Автоматически бит очищается при приеме следующего маркера(token) SETUP. Используется для конечных точек, сконфигурированных как управляющие.

17-7 MAXPL установка максимального размера одиночного USB пакета посылаемого или принимаемого для данной конечной точки. Максимально возможный размер одиночного пакета равен 1024 байта.

6-5 NT установка количества дополнительных транзакций на микрокадр для высокоскоростных конечных точек и на кадр для полноскоростных конечных точек. Данные биты предназначены только для изохронных конечных точек.

4-3 TT установка типа работы конечной точки: "00 управляющая(CTRL), "01 изохронная(ISOCH), "10 пакетная(BULK), "11 по прерываниям(INTERRUPT). Только конечные точки типа выход (OUT) должны устанавливаться в тип CTRL. В этом случае конечная точка на вход (IN) с этим же номером будет автоматически задействована.

2 EH остановка конечной точки ('1' – остановка включена, '0' - остановка выключена). Если данный бит установлен в "1 то все трансферы к данной точке будут получать ответ STALL.

1 ED отключение конечной точки ('1' – отключено, '0' - не отключено). Если данный бит установлен в "1 то все трансферы к данной точке будут получать ответ NAK.

0 EV разрешение работы конечной точки ('1' – разрешено, '0' - запрещено). Если работа запрещена, то все трансферы к данной точке буду проигнорированы без согласования.

EP0_IN_DM_ADDR	Регистр адреса ПДП конечной точки 0 на вход																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																														-	
Описание	DESCADDR																														-	

31-2 DESCADDR Адрес следующего дескриптора

0-1 — *зарезервировано*

USBxCR	Регистр управления																														
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0	0	0	0	0	—											-	0	0	0	-			0	-							-
Описание	SI	UI	VI	SP	FI	—											FT	EP	DH	RW	TS			TM	UA							SU

31 SI разрешение прерываний при изменении состояния режима приостановки(suspend) ('0' – запрещено, '1' – разрешено).

30 UI разрешение прерываний при обнаружении сигнала сброса USB ('0' – запрещено, '1' – разрешено).

29 VI разрешение прерываний при изменении состояния VBUS ('0' – запрещено, '1' – разрешено).

28 SP разрешение прерываний при изменении скоростного режима ('0' – запрещено, '1' – разрешено).

27 FI разрешение прерываний при получении маркера начала кадра(SOF) ('0' – запрещено, '1' – разрешено).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

26-16	—	*зарезервировано*
15	FT	разрешение функционального тестового режима, который сокращает все счетчики, такие как сброс и таймеры до 8 системных циклов. (’0’ – запрещено, ’1’ – разрешено).
14	EP	разрешение включения подтяжки на линии D+ подключенной к хосту (’0’ – запрещено, ’1’ – разрешено).
13	DH	управление скоростными режимами (’0’ – высокоскоростной режим, ’1’ – полноскоростной режим).
12	RW	запуск индикации удаленного сигнала wakeup. Данный бит переходит в состояние ’0’ после завершения передачи удаленного сигнала wakeup в режиме приостановки(suspend) и немедленно если не в режиме приостановки.
11-9	TS	выбор типа тестового режима(’001’ – Test_J, ’010’ - Test_K, ’011’ - Test_SE0_NAK, ’100’ - Test_Packet).
8	TM	разрешение режима тестирования, который нельзя отключить без сброса питания ядра (’0’ – разрешено, ’1’ – запрещено)
7-1	UA	адрес назначенный устройству на USB шине.
0	SU	разрешение установки адреса USB из поля UA(’0’ – запрещено, ’1’ – разрешено).

USBxST	**Регистр состояния**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—				—				-	—					-	-	-	-	-			—									
Описание	NEPI				NEPO				-	—					SU	UR	VB	SP	AF			FN									

31-28	NEPI	количество реализованных конечных точек типа IN.
27-24	NEPO	количество реализованных конечных точек типа OUT.
23-18	—	*зарезервировано*
17	SU	индикация режима приостановки(suspend) (’0’ – режим включен, ’1’ – режим отключен).
16	UR	индикация сигнала сброса USB (’0’ – нет сброса, ’1’ – есть сброс). Бит очищается записью ’1’.
15	VB	индикация корректного уровня напряжения на линии VBUS(’0’ – нет напряжения, ’1’ – есть напряжение).
14	SP	текущий скоростной режим шины (’0’ – высокоскоростной(high speed), ’1’ – полноскоростной(full speed))
13-11	AF	количество дополнительных кадров, полученных с текущим номером кадра.
10-0	FN	значение последнего принятого маркера SOF.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.11 Контроллер PWM(PWMx)

9.11.1 Краткие характеристики

- режим генерации одиночного импульса;
- возможность изменения периода счетчика во время его работы (при определенных условиях);

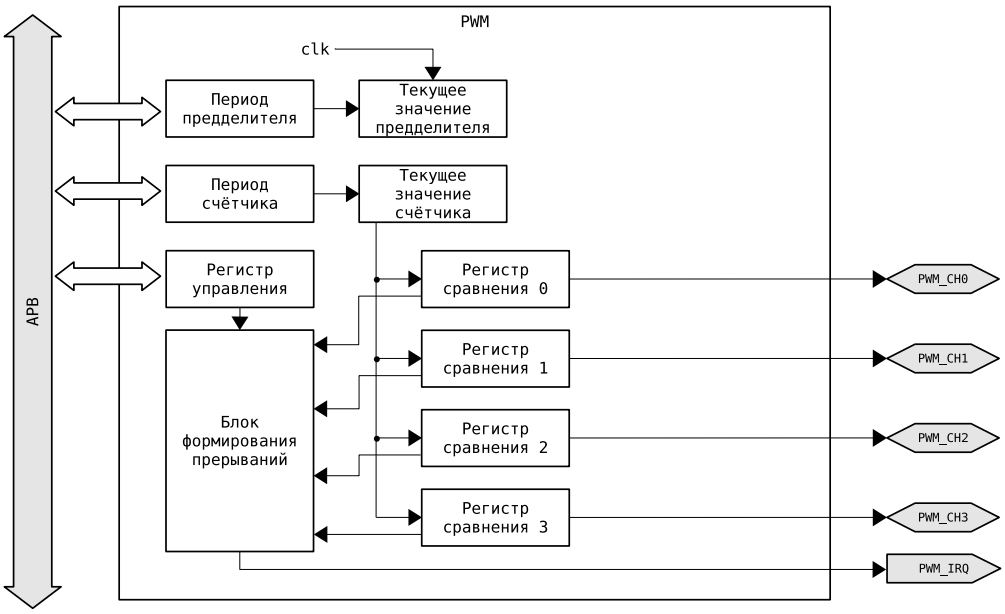

Рис. 45: Блок-схема PWM

Контроллер ШИМ (PWM) предназначен для генерирования широтно-модулированных импульсов. PWM является 4-х канальным и имеет режим генерации одиночного импульса, а также допускает возможность изменения периода счётчика во время его работы (при соблюдении условий, описываемых в соответствующем пункте).

9.11.2 Инициализация ШИМ

Для инициализации ШИМ необходимо задать в регистре PWM_CR режим работы интерфейса и счётчика, а также выбрать каналы, работу которых мы хотим разрешить. Следующим шагом устанавливаем активный/неактивный уровень включённого/-выключенного канала. Кроме того возможно разрешить прерывания соответствующего канала, а также возможно разрешение прерываний по переполнению счётчика.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.11.3 Режимы работы ШИМ

Генерация импульсов ШИМ возможна в однократном или периодическом режиме, установка производится в бите PWM_CR(PULSE_MODE). Счётчик запускается в трёх режимах: инкрементирующий, декрементирующий и режим увеличения до максимального значения, а потом уменьшения до нуля. Установка режима работы счётчика производится в бите PWM_CR(AUTO_RELOAD). Разрешается активировать возможность изменения периода счётчика во время его работы в бите PWM_CR(CNT_MODE), но только при определённых условиях.

9.11.4 Прерывания ШИМ

Прерывания бывают двух типов: по переполнению счётчика и по достижению счётчиком определённого канала значения регистра сравнения. Прерывания разрешаются в регистре управления, а запрос прерываний фиксируется в регистре PWM_INT.

9.11.5 Длительность импульса ШИМ

Длительность активного уровня импульса ШИМ будет определяться как разность значения периода счётчика PWM_CNT и значения регистра сравнения PWM_CMPCHn, умноженная на произведение длительности одного такта процессора и значения предделителя PWM_PSC.

$$T_{ACT} = (T_{cnt} - T_{cmpch}) \cdot T_{sys} \cdot (PSC + 1)$$

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.11.6 Описание регистров

Базовый адрес PWM - 0xC001 2000.

Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Примечание: Регистры сравнения при n=[0;3] будут иметь следующие адреса смещения:

PWMxCMPCH0 – 0x20

PWMxCMPCH1 – 0x24

PWMxCMPCH2 – 0x28

PWMxCMPCH3 – 0x2C

Регистр	Смещение адреса	Доступ	Описание
PWMxCR	00h	RW	Регистр управления
PWMxINT	04h	RW	Регистр прерываний
PWMxCNTVAL	08h	RW	Регистр текущего значения счётчика
PWMxPSC	0Ch	RW	Регистр значения предделителя
PWMxCNT	10h	RW	Регистр периода счётчика
PWMxCMPCHn	0x20-0x2Ch	RW	Регистр сравнения счётчика

PWMxCR	**Регистр управления**																												
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0		0	0
Описание	—								CH3MODE	CH2MODE	CH1MODE	CH0MODE	EN_CH3_IRQ	EN_CH2_IRQ	EN_CH1_IRQ	EN_CH0_IRQ	OUT_LVL_3	OUT_LVL_2	OUT_LVL_1	OUT_LVL_0	CH_EN_3	CH_EN_2	CH_EN_1	CH_EN_0	—			OVF_IRQ	CNT_MODE		AUTO_RELOAD	PULSE_MODE

24-31	—	*зарезервировано*
23	CH3_MODE	Установка активного уровня включенного канала 3 (’0’ – логический 0, ’1’ – логическая 1)
22	CH2_MODE	Установка активного уровня включенного канала 2 (’0’ – логический 0, ’1’ – логическая 1)
21	CH1_MODE	Установка активного уровня включенного канала 1 (’0’ – логический 0, ’1’ – логическая 1)
20	CH0_MODE	Установка активного уровня включенного канала 0 (’0’ – логический 0, ’1’ – логическая 1)
19	EN_CH3_IRQ	Разрешение прерывания канала 3 (’0’ – запрещено, ’1’ – разрешено)
18	EN_CH2_IRQ	Разрешение прерывания канала 2 (’0’ – запрещено, ’1’ – разрешено)
17	EN_CH1_IRQ	Разрешение прерывания канала 1 (’0’ – запрещено, ’1’ – разрешено)
16	EN_CH0_IRQ	Разрешение прерывания канала 0 (’0’ – запрещено, ’1’ – разрешено)
15	OUT_LVL_3	Установка уровня выключенного канала 3 (’0’ – логический 0, ’1’ – логическая 1)
14	OUT_LVL_2	Установка уровня выключенного канала 2 (’0’ – логический 0, ’1’ – логическая 1)
13	OUT_LVL_1	Установка уровня выключенного канала 1 (’0’ – логический 0, ’1’ – логическая 1)
12	OUT_LVL_0	Установка уровня выключенного канала 0 (’0’ – логический 0, ’1’ – логическая 1)
11	CH_EN3	Разрешение работы канала 3 (’0’ – запрещено, ’1’ – разрешено)
10	CH_EN2	Разрешение работы канала 2 (’0’ – запрещено, ’1’ – разрешено)
9	CH_EN1	Разрешение работы канала 1 (’0’ – запрещено, ’1’ – разрешено)
8	CH_EN0	Разрешение работы канала 0 (’0’ – запрещено, ’1’ – разрешено)
5-7	—	*зарезервировано*

Все права защищены © ОАО «Мультиклет» 2010 — 2015

4	OVF_IRQ	Разрешение прерывания по переполнению счётчика (’0’ – запрещено, ’1’ – разрешено)
2-3	CNT_MODE	Установка режима работы счетчика: 10 – счетчик увеличивается до максимального значения, а потом уменьшается до нуля 01 – декрементирующий 00 – инкрементирующий
1	AUTO_RELOAD	Разрешение изменения периода счетчика во время его работы (’0’ – запрещено, ’1’ – разрешено)
0	PULSE_MODE	Разрешение работы в однократном режиме (’0’ – запрещено, ’1’ – разрешено)

PWMxINT	**Регистр прерываний**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																												0	0	0	0
Описание	—																												CH3_IRQ	CH2_IRQ	CH1_IRQ	CH0_IRQ

4-31	—	*зарезервировано*
3	CH3_IRQ	Счетчик канала 3 достиг значения регистра сравнения
2	CH2_IRQ	Счетчик канала 2 достиг значения регистра сравнения
1	CH1_IRQ	Счетчик канала 1 достиг значения регистра сравнения
0	CH0_IRQ	Счетчик канала 0 достиг значения регистра сравнения

PWMxCNTVAL	**Регистр текущего значения счётчика**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																0															
Описание	—																CNT_VAL															

16-31	—	*зарезервировано*
0-15	CNT_VAL	Текущее значение счётчика

PWMxPSC	**Регистр значения предделителя**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																0															
Описание	—																PSC															

16-31	—	*зарезервировано*
0-15	PSC	Значение предделителя

PWMxCNT	**Регистр периода счётчика**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																0															
Описание	—																CNT_PER															

16-31	—	*зарезервировано*
0-15	CNT_PER	Значение периода счётчика

PWMxCMPCHn	**Регистр сравнения счётчика**																															
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0																0															
Описание	—																CMP_VAL															

16-31	—	*зарезервировано*
0-15	CMP_VAL	Значение регистра сравнения канала

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.12 Часы реального времени RTC

Часы реального времени предназначены только для работы при полном питании процессора. Работа часов реального времени от батарейки не предусмотрена.

9.12.1 Описание регистров

Базовый адрес RTC - 0xC001 F000
Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.13 Сторожевой таймер(WDT)

9.13.1 Краткие характеристики

- включен при старте процессора;
- счетчик периода 32 разряда;

Сторожевой таймер(WDT) включается автоматически при старте процессора. После того, как декрементирующий счетчик сторожевого таймера досчитает до нуля будет выработан сигнал сброса процессора. Сторожевой таймер может быть отключен, сброшен, а также может быть произведено изменение периода его работы. Для изменения периода необходимо: 1)Выключить WDT 2)Разрешить установку нового периода 3)Установить новый период 4)Включить WDT

9.13.2 Описание регистров

Базовый адрес WDT - 0xC00E 0000
Для получения реального адреса регистра надо к базовому (начальному) адресу на шине прибавить смещение адреса регистра

Регистр	Смещение адреса	Доступ	Описание
WDTxCNT	00h	RW	Период счетчика
WDTxKEY	04h	RW	Регистр переключения режимов
WDTxST	0Ch	RW	Регистр состояния

WDTxCNT	**Период счетчика**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	PERIOD																															

31-0 PERIOD период счетчика WDT

WDTxKEY	**Регистр переключения режимов**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	KEY_VAL																															

31-0 — *зарезервировано*
15-0 KEY_VAL регистр переключения режимов WDT:
0x5555 - включить WDT
0x3333 - выключить WDT
0xAAAA - сброс WDT
0xCCCC - разрешение записи периода счетчика

Все права защищены © ОАО «Мультиклет» 2010 — 2015

WDTxST	**Регистр состояния счетчика**																													
Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Описание	PERIOD																															EN

31-1	—	*зарезервировано*
0	EN	текущее состояние WDT (0 - не работает, 1 - работает)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.14 Аналого-цифровой преобразователь (ADCx)

9.14.1 Краткие характеристики

- четыре независимых канала с дельта-сигма преобразователями;
- номинальная частота тактирования: 12 МГц;
- максимальная частота выборок: 48 кГц;
- разрешение: 16 бит;
- программируемые усилители в каждом канале.

На рис.46 приведена блок-схема АЦП.

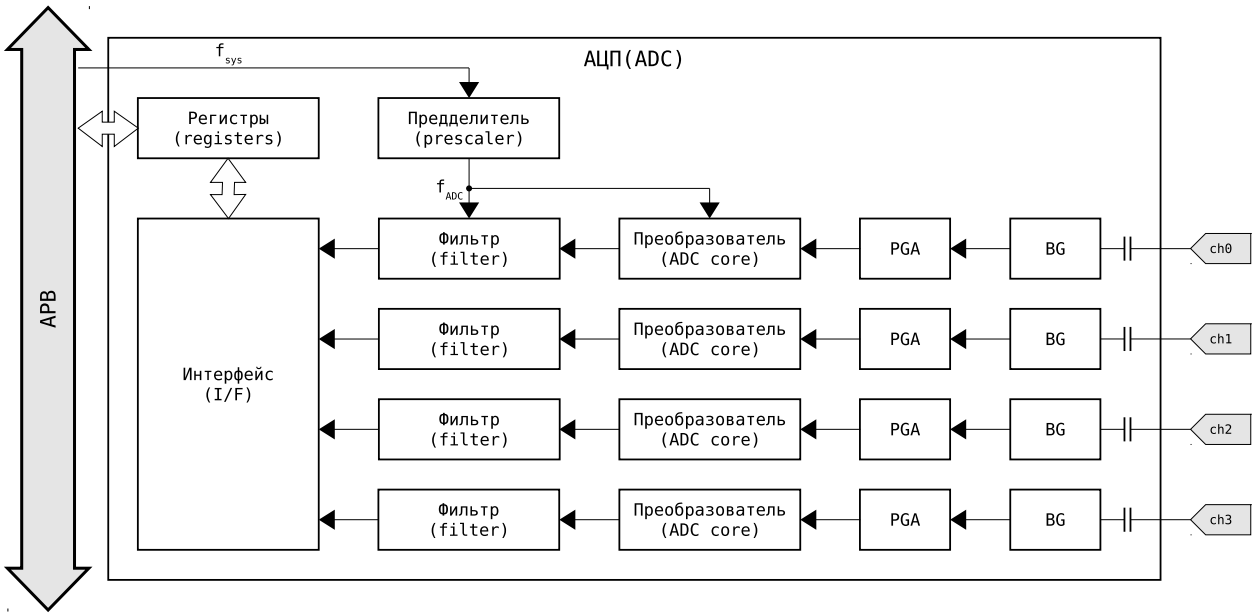

Рис. 46: Блок-схема АЦП

АЦП содержит 4 независимых канала сигма-дельта преобразования. Сигнал на вход каждого преобразователя подаётся через цепочку усилителей с изменяемыми коэффициентами усиления: усилители PGA (Programmable-gain amplifier) позволяют плавно изменять коэффициент усиления/ослабления. Усилители BG (Boost gain) позволяют внести дополнительное усиление в слабый сигнал (например, от микрофона) до 21 дБ. Значение коэффициентов описано в регистрах `ADCxCHy(PGA)` и `ADCxCHy(BG)`.

Входы каналов АЦП связаны с выводами процессора только по переменному току. Таким образом автоматически подавляется постоянная составляющая входного сигнала.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Каждый канал содержит схему автоматического “приглушения” (mute). Данная схема анализирует выходное значение канала преобразования, отбрасывая младшие биты (их число задаётся полем `ADCxCR(MUTE_SEL)`). Если получающееся значение на протяжении нескольких выборок равно нулю, в интерфейс передаётся результат преобразования, в точности равный нулю. Работа схемы автоматического приглушения разрешается битом `ADCxCR(MUTE_EN)`.

Тактовая частота АЦП f_{ADC} вырабатывается из системной тактовой частоты f_{sys} предделителем частоты с коэффициентом деления, определяемым полем `ADCxCR(PRCR)`:

$$f_{ADC} = \begin{cases} f_{sys} & \text{если } PRCR \leq 1 \\ \frac{f_{sys}}{PRCR'+2} & \text{если } PRCR > 1 \end{cases}$$

где $PRCR'$ — значение поля `ADCxCR(PRCR)` с младшим битом, равным ’0’, т.е. наибольшее *чётное* число, не превышающее `ADCxCR(PRCR)`.

Не допускается превышать номинальную тактовую частоту АЦП. Допускается тактировать АЦП частотой, меньшей номинальной, но при этом пропорционально изменятся частоты выборок, и, возможно, ухудшатся “аналоговые” характеристики.

Частота выборок может принимать значения из фиксированного набора, они приведены в описании поля `ADCxCR(SAMPL)`.

Не допускается подавать на вход АЦП сигналы с уровнями ниже напряжения avss_adc и выше напряжения avdd_adc.

Питанием каждого канала можно управлять отдельно с помощью битов `ADCxCR(PD_CHy)`.

Момент установки ADCxCHy(LRCK) в ’1’ совпадает с фиксированием очередного значения входного сигнала на входе сигма-дельта преобразователя. Также в этот момент вырабатывается прерывание АЦП. Через 8 периодов тактового сигнала АЦП частотой f_{ADC} в регистре `ADCxCHy(DATA)` становится доступно значение очередной выборки сигнала.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.14.2 Описание регистров

Базовый адрес ADC0: `0xC01D 0100`

Базовый адрес ADC1: `0xC01D 0200`

Для получения реального адреса регистра необходимо к базовому (начальному) адресу прибавить смещение адреса регистра.

Регистр	Смещение адреса	Доступ	Описание
ADCxCR	00h	RW	Регистр управления
ADCxCH0	04h	RW	Регистр 0 канала преобразователя
ADCxCH1	08h	RW	Регистр 1 канала преобразователя
ADCxCH2	0ch	RW	Регистр 2 канала преобразователя
ADCxCH3	10h	RW	Регистр 3 канала преобразователя

ADCxCR — **Регистр управления**

Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Начальное состояние	—												1111b			
Описание	—												PRCR			

Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	00b		10b		0000b				0	0	0	0	—	0	0	0
Описание	MUTE_SEL		PM		SAMPL				PD_CH3	PD_CH2	PD_CH1	PD_CH0	—	PD_BIAS	SUSPEND	MUTE_EN

31...20	—	*зарезервировано*
19...16	PRCR	предделитель системной частоты;
15...14	MUTE_SEL	порог срабатывания автоматического приглушения: 00 — отбрасывается младший бит, 01 — отбрасываются два младших бита, 10 — отбрасываются три младших бита, 11 — отбрасываются четыре младших бита;
13...12	PM	уровень мощности питания аналоговых каскадов: 00 — 60%, 01 - 80%, 10 — 100%, 11 —120%;
11...8	SAMPL	частота выборки: 0000 — 8 кГц, 0001 — 11,025 кГц, 0010 — 12 кГц, 0011 — 16 кГц, 0100 — 22,050 кГц, 0101 — 24 кГц, 0110 — 32 кГц, 0111 — 44,100 кГц, 1000 — 48 кГц;
7	PD_CH3	отключение канала преобразователя ('1' — выключен);
6	PD_CH2	отключение канала преобразователя ('1' — выключен);
5	PD_CH1	отключение канала преобразователя ('1' — выключен);
4	PD_CH0	отключение канала преобразователя ('1' — выключен);
3	—	*зарезервировано*
2	PD_BIAS	отключение тока смещения преобразователей ('1' — выключен);
1	SUSPEND	отключение АЦП ('1' — выключен);
0	MUTE_EN	разрешение работы схем автоматического приглушения ('1' — работа схемы разрешена).

ADCxCHy — **Структура регистров каналов преобразователя (y = 0,1,2,3)**

Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Начальное состояние	—			00000b					000b			0				
Описание	—			PGA					BG			LRCK				

Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0															
Описание	DATA															

31...29	—	*зарезервировано*
28...24	PGA	установка коэффициента передачи PGA (шаг 1,5 дБ): 11111 — 21 дБ, 11110 — 19,5 дБ, ..., 10001 — 0 дБ, ..., 00001 — -24 дБ, 00000 — -25.5 дБ;
22...20	BG	установка коэффициента усиления BG: 000 — 0 дБ, 001 — 3 дБ, 010 — 6 дБ, 011 — 9 дБ, 100 — 12 дБ, 101 — 15 дБ, 110 — 18 дБ, 111 — 21 дБ;

Все права защищены © ОАО «Мультиклет» 2010 — 2015

16	LRCK	признак готовности очередной выборки;
15...0	DATA	результат преобразования (двоичное *знаковое* значение).

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.15 Цифро-аналоговый преобразователь (DACx)

9.15.1 Краткие характеристики

- один независимый канал;
- разрешение: 12 бит;
- максимальная частота выборок: 200 МГц;
- тип выхода: токовый однополярный;
- максимальный вытекающий ток: 16 мА.

На рис.?? приведена блок-схема ЦАП. ЦАП относится к преобразователям с токовым переключением. Вытекающий ток I_{DAC} зависит от значения D, подаваемого на вход ЦАП, следующим образом:

$$I_{DAC} = D \cdot \frac{V_{ref}}{R_{ref}}$$

где V_{ref} — напряжение внутреннего источника опорного напряжения ЦАП (1,2 В), R_{ref} — сопротивление токозадающего резистора, подключенного между выводами rref и avss_dac процессора.

Сопротивление токозадающего резистора следует выбирать таким образом, чтобы в процессе выходной ток ЦАП не превышал максимально допустимого значения.

Если необходимо получить выход по напряжению, то между выходом ЦАП и линией avss_dac необходимо включить резистор сопротивлением 50...100 Ом. Кроме того, сопротивление данного резистора должно выбираться таким образом, чтобы напряжение на выходе ЦАП не превышало 1,4 В.

Для начала работы необходимо установить бит DACxCR(PD) в состояние '0'.

Примечание:
Оба канала ЦАП должны объединяться в один на плате, выходной ток вытекает из точки объединения. В итоге получается один канал. В одноимённые регистры обоих каналов ЦАП необходимо записывать идентичные данные. Только в этом случае гарантируется стабильная работа ЦАП.

Все права защищены © ОАО «Мультиклет» 2010 — 2015

9.15.2 Описание регистров

Базовый адрес DAC0: `0xC01D 1000`

Базовый адрес DAC1: `0xC01D 1100`

Для получения реального адреса регистра необходимо к базовому (начальному) адресу прибавить смещение адреса регистра.

Регистр	Смещение адреса	Доступ	Описание
DACxCR	00h	RW	Регистр управления
DACxDATA	04h	RW	Регистр данных преобразователя

DACxCR — **Регистр управления**

Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Начальное состояние	—															
Описание	—															

Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	0															0
Описание	—															PD

31...1 — *зарезервировано*

0 PD режим пониженного энергопотребления ('1' — ЦАП выключен)

DACxDATA — **Структура регистра данных преобразователя**

Номер бита	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Начальное состояние	—															
Описание	—															

Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Начальное состояние	—				0											
Описание	—				DATA											

31...12 — *зарезервировано*

11...0 DATA данные в прямом двоичном коде (0x000 — минимальный ток на выходе)

Все права защищены © ОАО «Мультиклет» 2010 — 2015

10 Назначение выводов процессора

10.1 Назначение выводов процессора в корпусе QFP256

Условные обозначения	
s	подсоединение к линиям электропитания
i	вход
o	выход
a	вход/выход аналоговых сигналов
nc	не подсоединен
vdd	1.8В ядро
vss	gnd питания ядра
dvss	3.3В порты ввода/вывода
dvdd	gnd питания портов
vdd_pll	1.8В ядро PLL
vss_pll	gnd питания ядра PLL
avdd_pll	1.8В аналоговые блоки PLL
avss_pll	gnd питания аналоговых блоков PLL
avss_dac	gnd питания аналоговых блоков DAC
avdd_dac	3.3В аналоговые блоки DAC
vdd_conv	1.8В ядра конверторов (АЦП/ЦАП)
vss_conv	gnd питания ядер конверторов (АЦП/ЦАП)
avss_adc	gnd питания аналоговых блоков DAC
avdd_adc	3.3В аналоговые блоки DAC

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
1	i/o	GPIOF	0	usb0_data[0]	
2	i/o		1	usb0_data[1]	
3	i/o		2	usb0_data[2]	
4	i/o		3	usb0_data[3]	
5	i/o		4	usb0_data[4]	
6	i/o		5	usb0_data[5]	
7	s	dvss			
8	i/o	GPIOF	6	usb0_data[6]	
9	i/o		7	usb0_data[7]	
10	i/o		8	usb0_nxt	
11	i/o		9	usb0_dir	
12	i/o		10	usb0_stp	
13	i/o		11	usb0_urstdrive	
14	i/o		12	usb0_reset	
15	i/o	GPIOE	27	em_bexcn	
16	i/o		26	em_brdyn	
17	i/o		25	em_sdram_dqm[3]	
18	i/o		24	em_sdram_dqm[2]	
19	i/o		23	em_sdram_dqm[1]	
20	i/o		22	em_sdram_dqm[0]	
21	i/o		0	em_ramsn[0]	
22	i/o	GPIOD	31	em_data[31]	
23	i/o	GPIOE	21	em_writen	
24	s	vdd			
25	s	vss			
26	s	dvss			
27	s	dvdd			
28	s	vdd_pll			
29	s	vss_pll			

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
30	s	avdd_pll			
31	s	avss_pll			
32	s			pad_a	
33	s	avss_tpll			
34	s	avdd_tpll			
35	s			pad_b	
36	s	avss_tpll			
37	i/o	GPIOE	20	em_read	
38	i/o		19	em_oen	
39	i/o		18	em_iosn	
40	i/o		17	em_romsn[1]	
41	i/o		16	em_romsn[0]	
42	i/o		15	em_mben[3]	
43	s	vdd			
44	s	vss			
45	i/o	GPIOE	14	em_mben[2]	
46	i/o		13	em_mben[1]	
47	i/o		12	em_mben[0]	
48	i/o		11	em_wrn[3]	
49	i/o		10	em_wrn[2]	
50	i/o		6	em_ramoen[2]	
51	i/o		5	em_ramoen[1]	
52	i/o		4	em_ramoen[0]	
53	i/o		3	em_ramsn[3]	
54	i/o		2	em_ramsn[2]	
55	i/o		1	em_ramsn[1]	
56	s	dvss			
57	s	dvdd			
58	s	vdd			
59	s	vss			
60	i/o	GPIOD	30	em_data[30]	
61	i/o		29	em_data[29]	
62	i/o		28	em_data[28]	
63	i/o		27	em_data[27]	
64	s	dvss			
65	i/o	GPIOE	9	em_wrn[1]	
66	i/o		8	em_wrn[0]	
67	i/o		7	em_ramoen[3]	
68	i/o	GPIOD	26	em_data[26]	
69	i/o		25	em_data[25]	
70	i/o		24	em_data[24]	
71	s	dvss			
72	i/o	GPIOD	23	em_data[23]	
73	i/o		22	em_data[22]	
74	i/o		21	em_data[21]	
75	i/o		20	em_data[20]	
76	i/o		19	em_data[19]	
77	i/o		18	em_data[18]	
78	i/o		17	em_data[17]	
79	i/o		16	em_data[16]	

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
80	i/o		15	em_data[15]	
81	i/o		14	em_data[14]	
82	s	vdd			
83	s	vss			
84	s	dvss			
85	s	dvdd			
86	i/o	GPIOD	13	em_data[13]	
87	i/o		12	em_data[12]	
88	i/o		11	em_data[11]	
89	i/o		10	em_data[10]	
90	i/o		9	em_data[9]	
91	i/o		8	em_data[8]	
92	i/o		7	em_data[7]	
93	i/o		6	em_data[6]	
94	s	vdd			
95	s	vss			
96	i/o	GPIOD	5	em_data[5]	
97	i/o		4	em_data[4]	
98	i/o		3	em_data[3]	
99	i/o		2	em_data[2]	
100	i/o		1	em_data[1]	
101	i/o		0	em_data[0]	
102	s	dvss			
103	s	dvdd			
104	s	vdd			
105	s	vss			
106	i/o	GPIOC	27	i2s_ws	
107	i/o		26	i2s_sd	
108	i/o		25	i2s_sck	
109	i/o		22	uart2_txd	
110	i/o		21	uart2_rxd	
111	i/o		20	em_sdram_casn	
112	i/o		19	em_sdram_rasn	
113	i/o		18	em_sdram_sdwen	
114	i/o		17	em_sdram_sdcsn[1]	
115	i/o		16	em_sdram_sdcsn[0]	
116	i/o		15	em_sdram_sdcke[1]	
117	i/o		14	em_sdram_sdcke[0]	
118	i/o		11	uart0_txd	
119	i/o		10	uart0_rxd	
120	i/o		9	clk_out	
121	i/o		8	spi0_ssl[1]	
122	s	dvss			
123	i/o	GPIOC	7	spi0_ssl[0]	
124	i/o		6	spi0_aready	
125	i/o		5	spi0_astart_o	
126	i/o		4	spi0_astart_i	
127	i/o		3	spi0_spisel	
128	i/o		2	spi0_miso	
129	i/o		1	spi0_mosi	

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
130	i/o		0	spi0_sck	
131	i/o	GPIOB	29	em_addr[29]	
132	i/o		28	em_addr[28]	
133	i/o		27	em_addr[27]	
134	i/o		26	em_addr[26]	
135	s	dvss			
136	i/o	GPIOB	25	em_addr[25]	
137	i/o		24	em_addr[24]	
138	i/o		23	em_addr[23]	
139	i/o		22	em_addr[22]	
140	i/o		21	em_addr[21]	
141	i/o		20	em_addr[20]	
142	i/o		19	em_addr[19]	
143	i/o		18	em_addr[18]	
144	i/o		17	em_addr[17]	
145	i/o		16	em_addr[16]	
146	s	vdd			
147	s	vss			
148	s	dvss			
149	s	dvdd			
150	i/o	GPIOB	15	em_addr[15]	
151	i/o		14	em_addr[14]	
152	i/o		13	em_addr[13]	
153	i/o		12	em_addr[12]	
154	i/o		11	em_addr[11]	
155	i/o		10	em_addr[10]	
156	i/o		9	em_addr[9]	
157	i/o		8	em_addr[8]	
158	i/o		7	em_addr[7]	
159	i/o		6	em_addr[6]	
160	s	vdd			
161	s	vss			
162	i/o	GPIOB	5	em_addr[5]	
163	i/o		4	em_addr[4]	
164	i/o		3	em_addr[3]	
165	i/o		2	em_addr[2]	
166	i/o		1	em_addr[1]	
167	i/o		0	em_addr[0]	
168	i/o	GPIOA	29	pwm_out[3]	
169	i/o		28	pwm_out[2]	
170	i/o		27	pwm_out[1]	
171	i/o		26	pwm_out[0]	
172	i/o		25	uart1_rtsn	
173	i/o		24	uart1_ctsn	
174	i/o		23	uart1_txd	
175	i/o		22	uart1_rxd	
176	i/o		21	i2c0_sda	
177	i/o		20	i2c0_scl	
178	s	dvss			
179	s	dvdd			

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
180	s	vdd			
181	s	vss			
182	i/o	GPIOA	19	eth0_reset	
183	i/o		18	eth0_mdint	
184	i/o		17	eth0_mdc	
185	i/o		16	eth0_mdio	
186	s	dvss			
187	i/o	GPIOA	15	eth0_rx_clk	
188	i/o		14	eth0_rxd[3]	
189	i/o		13	eth0_rxd[2]	
190	i/o		12	eth0_rxd[1]	
191	i/o		11	eth0_rxd[0]	
192	i/o		10	eth0_rx_er	
193	i/o		9	eth0_rx_dv	
194	i			en_rcfg	
195	s	dvss			
196	s	dvdd			
197	i			trst	JTAG (IEEE 1149.1)
198	i			tck	
199	i			tms	
200	i			st[0]	
201	o			cr[0]	
202	s	dvss			
203	i			tdi	JTAG (IEEE 1149.1)
204	o			tdo	
205	i			wakeup	
206	i			prom_width[0]	
207	i			nmi	
208	i			reset	
209	i			rref	
210	o			vref	
211	o			dac[0]	
212	o			dac[1]	
213	s	avss_dac			
214	s	avdd_dac			
215	s	vdd_conv			
216	s	vss_conv			
217	o			vref_adc	
218	i			adc[0]	
219	i			adc[1]	
220	i			adc[2]	
221	s	avss_adc			
222	s	avdd_adc			
223	i			adc[3]	
224	i			adc[4]	
225	i			adc[5]	
226	i			adc[6]	
227	i			adc[7]	
228	i			start_addr[1]	
229	i/o		8	eth0_rx_crs	

GPIOA

Все права защищены © ОАО «Мультиклет» 2010 — 2015

№	Тип	Порт	Бит порта	Альтернативная функция	Примечания
230	i/o		7	eth0_tx_clk	
231	i/o		6	eth0_txd[3]	
232	i/o		5	eth0_txd[2]	
233	i/o		4	eth0_txd[1]	
234	i/o		3	eth0_txd[0]	
235	s	dvss			
236	s	dvdd			
237	s	vdd			
238	s	vss			
239	i			osc_main_in	
240	o			osc_main_out	
241	s	dvss			
242	i/o		2	eth0_tx_er	
243	i/o	GPIOA	1	eth0_tx_en	
244	i/o		0	eth0_rx_col	
245	i			osc32k_in	
246	o			osc32k_out	
247	i			usb_clk	
248	i/o		29	rtc0_a	
249	i/o		28	i2c1_sda	
250	i/o		27	i2c1_scl	
251	i/o		20	spi1_spisel	
252	i/o	GPIOF	19	spi1_miso	
253	i/o		18	spi1_mosi	
254	i/o		17	spi1_sck	
255	i/o		14	uart3_txd	
256	i/o		13	uart3_rxd	

Все права защищены © ОАО «Мультиклет» 2010 — 2015

10.2 Диаграмма выводов процессора в корпусе QFP256

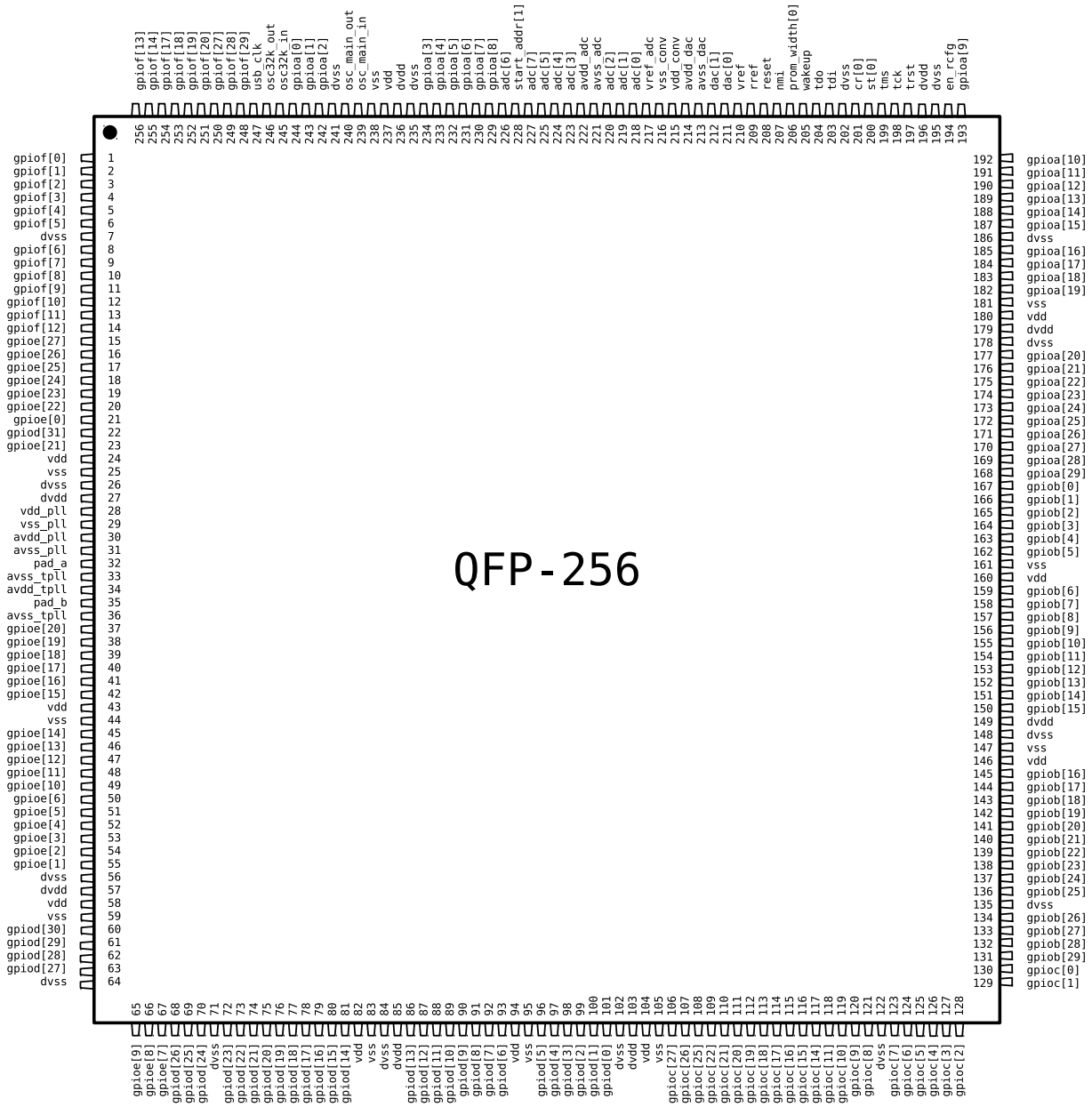

Рис. 47: Диаграмма выводов процессора

Все права защищены © ОАО «Мультиклет» 2010 — 2015

11 Электрические параметры

11.1 Электрические характеристики портов ввода-вывода

Таблица 32: Электрические характеристики портов ввода-вывода

Параметр		Условия	Мин.	Тип.	Макс.
V_{IL}	Низкий уровень входного напряжения, В	CMOS, LVTTL	$-0,3$		$0,8$
V_{IH}	Высокий уровень входного напряжения, В		$2,0$		5.5
I_{IL}	Низкий уровень входного тока, мкА	$V_{in} = V_{SS}$	-10		10
I_{IH}	Высокий уровень входного тока, мкА	$V_{in} = DVDD$	-10		10
V_{OL}	Низкий уровень выходного напряжения, В	$I_{OL} = -12$мА	0		$0,4$
V_{OH}	Высокий уровень выходного напряжения, В	$I_{OH} = 12$мА	2.4		3.6
	Подтягивающий резистор к высокому уровню, кОм		$68,2$		$118,1$
	Подтягивающий резистор к низкому уровню, кОм		$30,2$		$80,6$

Таблица 33: Электрические характеристики ЦАП

Параметр		Условия	Мин.	Тип.	Макс.
	Диапазон рабочих температур, ℃		-55	-	$+125$
avss_dac	Напряжение питания аналоговых частей ЦАП, В		$-2,97$	$3,3$	$3,63$
vdd_conv	Напряжение питания цифровых частей ЦАП, В		$1,62$	1.8	1.98
I_{DACPD}	Ток потребления в неактивном режиме, мкА		-	-	10
I_{omax}	Максимальный выходной ток, мА		-	- 16	24
$ENOB$	Эффективное разрешение, бит		-	-	$9,5$
INL	Интегральная нелинейность, бит		-	-	-
DNL	Дифференциальная нелинейность, бит		-	-	-

Таблица 34: Электрические характеристики АЦП

Параметр		Условия	Мин.	Тип.	Макс.
	Диапазон рабочих температур, ℃		-55	-	$+125$
avss_adc	Напряжение питания аналоговых частей АЦП, В		$-2,97$	$3,3$	$3,63$
vdd_conv	Напряжение питания цифровых частей АЦП, В		$1,62$	1.8	1.98
I_{ADCPD}	Ток потребления в неактивном режиме по цепи vdd_conv, мкА		-	-	10
I_{ADCC}	Ток потребления в активном режиме по цепи vdd_conv, мА	ADCxCR(PD) = 10	-	-	7
I_{ADCAPD}	Ток потребления в неактивном режиме по цепи avss_adc, мкА		-	-	10
I_{ADCA}	Ток потребления в активном режиме по цепи avss_adc, мА		-	-	5
	Динамический диапазон, дБ		-	85	-
	Входное сопротивление, кОм		-	30	-

Все права защищены © ОАО «Мультиклет» 2010 — 2015

Все права защищены © ОАО «Мультиклет» 2010 — 2015

A Приложение 1. Руководство по применению сложно-функциональных блоков(СФБ) «GAISLER»

ПРИЛОЖЕНИЕ НАХОДИТСЯ В РАЗДЕЛЕ САЙТА «ПОДДЕРЖКА → ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ» (www.multiclet.com/docs/grip.pdf)

Все права защищены © ОАО «Мультиклет» 2010 — 2015