



Процессоры ADSP-BF53... серии

Краткое описание

ADSP-BF53... функциональная блок диаграмма

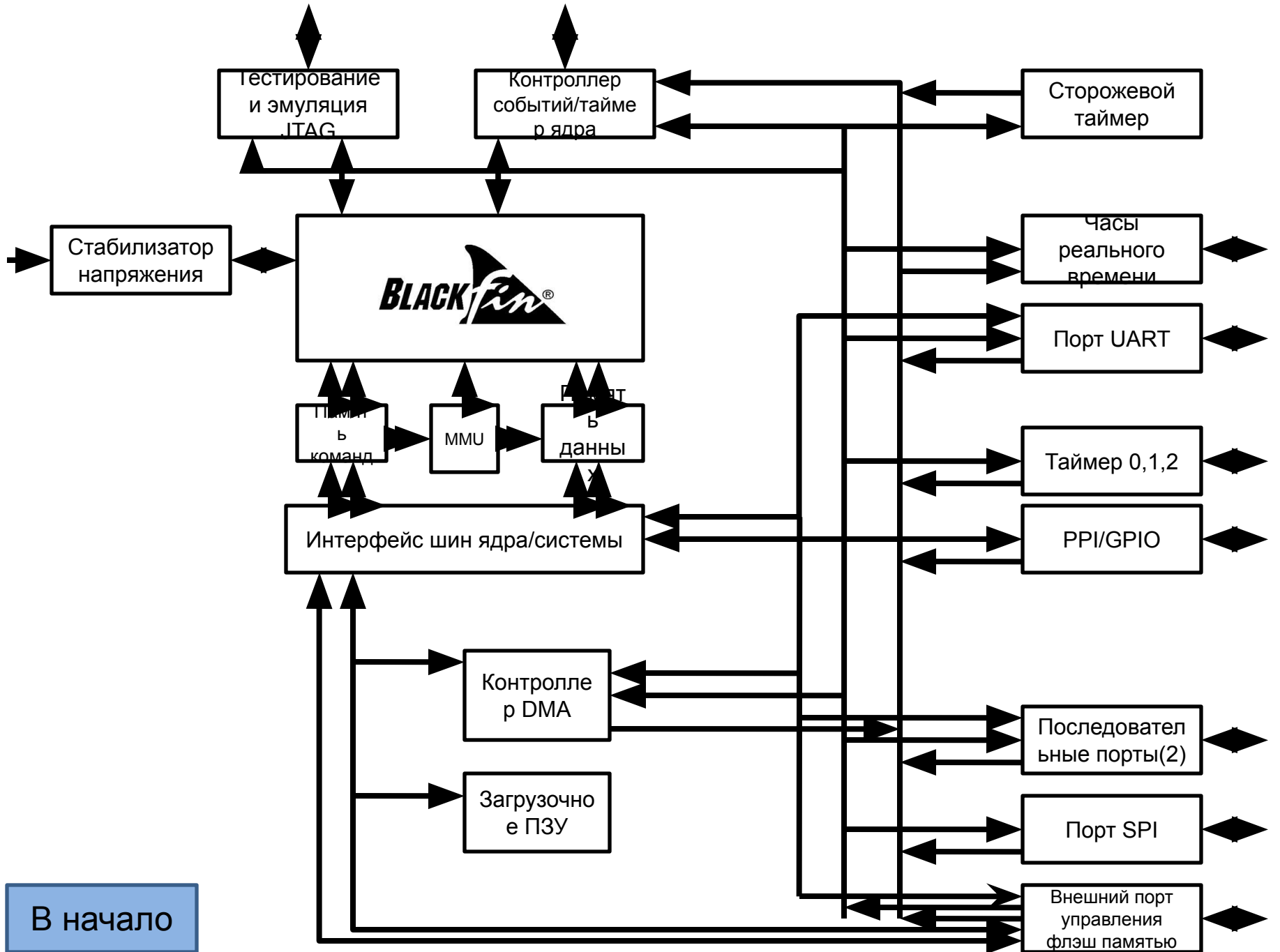
Программа переведена студентом МТУСИ.
Не для продажи, распространяется
бесплатно

Процессоры ADSP-BF531, ADSP-BF532 и ADSP-BF533 являются представителями семейства процессоров Blackfin с расширенными возможностями, которые обладают значительно большей производительностью и меньшей потребляемой мощностью по сравнению с предыдущими процессорами семейства Blackfin при сохранении простоты использования и совместимости кода. Три новых процессора полностью совместимы по выводам и различаются только производительностью и размером внутренней памяти, что позволяет избежать многих трудностей, возникающих при разработке новых продуктов.

Архитектура ядра процессора Blackfin является архитектурой с единым набором команд, включающей ядро обработки сигналов со сдвоенным блоком умножения-накопления, имеющей ортогональный набор команд, характерный для RISC-микропроцессоров, обладающей гибкостью команд типа SIMD и мультимедийными возможностями.

Особенностью продуктов семейства Blackfin является динамическое управление питанием. Возможность изменения как напряжения питания, так и рабочей частоты позволяет оптимизировать потребление мощности в соответствии с конкретной задачей.





В начало

Тестирование и эмуляция

JTAG Эмуляторы производства Analog Devices используют порт доступа тестирования JTAG IEEE 1149.1 процессора для наблюдения и управления процессором на отлаживаемой плате при эмуляции. Эмулятор обеспечивает полноскоростную эмуляцию, позволяющую отслеживание состояния и модификацию памяти, регистров и стеков процессора.

Внутрисхемная эмуляция без вмешательства в работу процессора обеспечивается использованием интерфейса JTAG – эмулятор не влияет на нагрузку или временные характеристики тестируемой системы.

Стабилизация напряжения

Процессор имеет внутренний стабилизатор напряжения, способный формировать уровни внутреннего напряжения (от 0,8 В до 1,2 В) из внешнего напряжения питания (от 2,25 В до 3,6 В). На рис. 1 показаны типичные внешние компоненты, необходимые для завершения системы управления питанием.

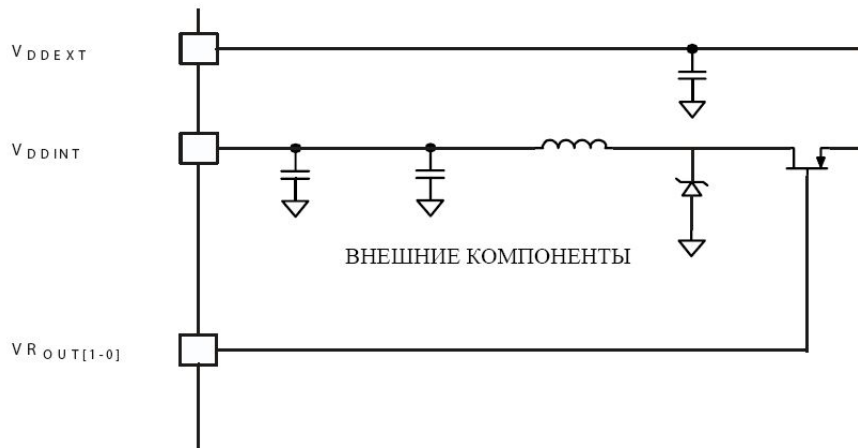


Рис. 1. Схема стабилизатора напряжения

Стабилизатор управляет уровнями напряжения внутренней логики, которые могут программно изменяться в регистре управления стабилизатором напряжения (VR_CTL) с инкрементом 50 мВ. Внутренний стабилизатор напряжения может быть запрограммирован таким образом, чтобы отключать питание ядра процессора, сохраняя подачу питания устройств ввода/вывода, в целях снижения потребляемой в режиме ожидания мощности. При нахождении в таком состоянии можно сохранять подачу V_{DDEXT} , что устраняет необходимость использования внешних буферов. Также, по желанию пользователя, можно отключать стабилизатор и обходить его (запрещать его использование).

Сторожевой таймер

Процессор содержит 32-разрядный таймер, который может использоваться для реализации функции программного сторожевого таймера. Использование программного сторожевого таймера может повысить устойчивость системы при помощи принудительного перевода процессора в известное состояние путём генерации аппаратного сброса, немаскируемого прерывания (NMI) или прерывания общего назначения, если таймер истекает до того, как он будет сброшен программно. Программист инициализирует счётчик таймера, разрешает соответствующее прерывание и затем разрешает работу таймера. После этого, счётчик должен быть перезагружен программно до достижения им нуля. Это защищает систему от попадания в неизвестное состояние в случае, когда выполнение программы, которая при нормальной работе сбросила бы таймер, останавливается из-за воздействия внешнего шума или программной ошибки.

Если сторожевой таймер настроен на генерирование аппаратного сброса, он выполняет сброс и ЦПУ и периферийных устройств. После выполнения сброса программа может определить, являлся ли источником аппаратного сброса сторожевой таймер, путём проверки бита состояния в регистре управления сторожевым таймером. Таймер работает от тактового сигнала системы (SCLK) с максимальной частотой fSCLK.

Часы реального времени

Часы реального времени (RTC, Real-Time Clock) процессора предоставляют набор свойств цифровых часов, включающий функции будильника, секундомера и индикации текущего времени. RTC тактируется внешним кварцевым резонатором с частотой 32,768 кГц. Так как периферия RTC имеет выделенные выводы питания, она может находиться во включённом состоянии и тактироваться, даже когда остальная часть процессора находится в состоянии пониженного потребления мощности. RTC обеспечивает несколько возможных программируемых прерываний, включающих прерывания по секундам, минутам, часам или дням, прерывание программируемого секундомера, или прерывание будильника по наступлению программно заданного времени.

Сигнал частотой 1 Гц получается делением входного тактового сигнала частотой 32,768 кГц в делителе. Функция счёта таймера осуществляется четырьмя счётчиками: 60-секундным счётчиком, 60-минутным счётчиком, 24-часовым счётчиком и 32768-дневным счётчиком.

При разрешении функции будильника по совпадению значения таймера со значением, запрограммированным в регистре управления будильником, генерируется прерывание. Существует два режима будильника. В первом режиме будильник настраивается на время дня. Во втором режиме будильник настраивается на день и время в течение заданного дня. Функция секундомера выполняет обратный отсчёт от запрограммированного значения с минутным разрешением. При разрешении функции будильника по достижению счётчиком нуля генерируется прерывание.

Как и остальные периферийные устройства, RTC может выводить процессор из режима Сна или режима Глубокого Сна при генерации любого события RTC. По событиям RTC может производиться также вывод внутреннего стабилизатора напряжения из выключенного состояния.

Порт UART

Процессор имеет полудуплексный порт универсального асинхронного приёмника-передатчика (UART, Universal Asynchronous Receiver/Transmitter), полностью совместимый со стандартными портами UART персонального компьютера. Порт UART обеспечивает упрощённый UART-интерфейс с другими периферийными устройствами или хост-процессорами, реализующий полудуплексные асинхронные передачи данных в последовательном режиме с поддержкой DMA. Порт UART поддерживает использование от 5 до 8 битов данных, 1 или 2 стоповых битов, а также возможность передачи данных без проверки и с проверкой на чётность или нечётность. Порт UART поддерживает два режима работы:

- Запрограммированный ввод/вывод

Процессор посылает или принимает данные путём записи в регистры UART, отображённые в карте памяти. При приёме и передаче данные подвергаются двойной буферизации.

- Прямой доступ к памяти (DMA)

Передачи принимаемых и передаваемых данных осуществляются контроллером DMA. При этом уменьшается количество и частота прерываний, необходимых для передач данных в память и из памяти. UART имеет два выделенных канала DMA – для приёма и для передачи. Вследствие относительно низких скоростей обслуживания эти каналы DMA имеют приоритет, меньший, чем большинство других каналов DMA.

Скорость передачи в бодах, формат последовательно передаваемых данных, состояние и генерация кода ошибки, и прерывания UART могут настраиваться программно, что позволяет поддерживать:

- широкий диапазон скоростей передачи в битах;
- форматы данных от 7 до 12 битов на кадр;
- генерацию маскируемых прерываний процессора при операциях приёма и передачи.

При использовании порта UART совместно с таймерами общего назначения поддерживается автоматическое определение скорости передачи в канале.

Возможности порта UART, дополнены поддержкой протокола спецификации физического уровня инфракрасного канала последовательной передачи данных (SIR, Serial Infrared Physical Layer Link Specification), разработанного ассоциацией передачи данных по инфракрасному каналу (IrDA®, Infrared Data Association).

Таймеры

Процессор имеет четыре программируемых таймера общего назначения. Три таймера имеют внешний вывод, который может быть сконфигурирован для использования либо в качестве широтно-импульсного модулятора (PWM, Pulse Width Modulator) или выходного сигнала таймера, либо в качестве входа тактового сигнала таймера, либо в качестве механизма измерения ширины импульса внешнего сигнала. Эти таймеры могут синхронизироваться внешним тактовым сигналом, подаваемым на входной вывод PF1 или PPI_CLK, или внутренним сигналом SCLK.

Таймеры могут использоваться совместно с UART для измерения ширины импульсов в потоке данных, обеспечивая функцию автоматического определения скорости передачи в последовательном канале.

Таймеры могут генерировать прерывания ядра процессора для обеспечения периодической синхронизации по тактовому сигналу процессора или по последовательности внешних сигналов. В дополнение к трём таймерам общего назначения в процессоре также имеется дополнительный четвёртый таймер. Он тактируется внутренним тактовым сигналом процессора и обычно используется в качестве системных часов для генерации периодических прерываний операционной системы.

[К блок-схеме таймера](#)

[К блок-схеме](#)

[В начало](#)

Порт последовательного периферийного интерфейса (SPI-порт)

Процессор имеет SPI-совместимый порт, который позволяет процессору соединяться с различными SPI-совместимыми устройствами.

SPI интерфейс имеет три вывода для передачи данных: два вывода данных и один вывод тактового сигнала. Входной сигнал выбора микросхемы SPI позволяет другим SPI-совместимым устройствам выбирать данный процессор, а семь выходных сигналов выбора микросхемы SPI позволяют процессору выбирать другие SPI-совместимые устройства. Выводы выбора SPI являются реконфигурируемыми выводами программируемых флагов. Используя эти выводы, SPI-порт обеспечивает полнодуплексный, синхронный последовательный интерфейс, который поддерживает режимы ведущего и ведомого, а также режим работы в системе с несколькими ведущими.

Скорость передачи данных и фаза/полярность сигналов тактовой синхронизации порта SPI могут программно изменяться. Порт имеет интегрированный контроллер DMA, который конфигурируется таким образом, чтобы поддерживать либо входные, либо выходные потоки данных. В отдельно взятый момент времени контроллер DMA SPI-порта может обслуживать только однонаправленные обращения.

Во время выполнения передач порт SPI одновременно передаёт и принимает данные, осуществляя их последовательный сдвиг по двум линиям данных. Выборка и сдвиг данных на этих линиях управляются по линии последовательного тактового сигнала.

[К блок-схеме SPI-интерфейса](#)

[К блок-схеме](#)

[В начало](#)

Последовательные порты (SPORT)

Процессор имеет два двухканальных синхронных последовательных порта (SPORT0 и SPORT1), обеспечивающих последовательные и межпроцессорные соединения. Последовательные порты имеют следующие особенности:

- **Двунаправленная работа с поддержкой I2S.** Каждый последовательный порт имеет два набора независимых выводов для передачи и приема, позволяющих организовать восемь каналов стереозвука в формате I2S.
- **Порты передачи и приема с буферизацией (глубиной 8 слов).** Каждый порт имеет регистр данных для передачи слов данных в другие процессорные компоненты и приема из них, а также регистры сдвига, соединенные с регистрами данных;
- **Тактирование.** Каждый порт приема и передачи может использовать внешний последовательный тактовый сигнал или генерировать собственный тактовый сигнал в широком диапазоне частот.
- **Длина слова.** Каждый порт поддерживает последовательную передачу слов данных длиной от 3 до 32 разрядов. Данные могут передаваться, начиная со старшего разряда или начиная с младшего разряда.
- **Кадровая синхронизация.** Каждый порт передачи и приема может функционировать с сигналами кадровой синхронизации, сопровождающими каждое слово данных, или без них. Сигналы кадровой синхронизации могут генерироваться внутри порта или внешним источником, иметь активный низкий или активный высокий уровень. Возможна кадровая синхронизация с любой из двух длительностей импульса, а также выбор кадровой синхронизации с опережением или запаздыванием.
- **Аппаратное компандирование.** Каждый последовательный порт может выполнять А- или μ -компандирование в соответствии с рекомендацией ITU G.711. Компандирование может выбираться для порта передачи и/или приёма, оно не вносит дополнительных задержек.
- **Операция с DMA с непроизводительными затратами в один такт.** Каждый последовательный порт может автоматически принимать и передавать несколько буферов памяти данных. Процессор может осуществлять связанные передачи или цепочки последовательностей передач DMA между последовательным портом и памятью.
- **Прерывания.** Каждый порт передачи и приема генерирует прерывание после завершения передачи слова данных или после передачи целого буфера с данными или нескольких буферов при помощи DMA.
- **Многоканальность.** Каждый последовательный порт поддерживает работу 128 каналов из 1024-канального окна и совместим со стандартами H.100, H.110, MVIP-90 и HMVIP.

Параллельный периферийный интерфейс

В процессоре имеется параллельный периферийный интерфейс (PPI, Parallel Peripheral Interface), который позволяет напрямую соединяться с параллельными ЦАП и АЦП, видеокодерами и декодерами стандарта ITU-R 601/656, и периферийными устройствами ввода/вывода общего назначения. PPI включает выделенный вывод входного тактового сигнала, до трёх выводов кадровой синхронизации и до 16 выводов данных. Входной тактовый сигнал допускает параллельную передачу данных со скоростями, достигающими половины тактовой частоты системы.

Поддерживается три различных режима ITU-R 656:

- Приём активной видеоинформации – PPI не выполняет чтение данных, поступающих между символами преамбул Конца Активного Видео (EAV, End of Active Video) и Начала Активного Видео (SAV, Start of Active Video), или данных, присутствующих в течение интервалов обратного хода луча. В данном режиме последовательности управляющих байтов не сохраняются в память; они фильтруются в PPI.
- Приём информации в интервалах обратного хода кадровой развёртки – PPI используется для передач данных в течение интервала обратного хода кадровой развёртки (VBI, Vertical Blanking Interval), а также информации в течение интервала обратного хода строчной развёртки и последовательности управляющих байтов по строкам VBI.
- Приём полного поля – PPI производит чтение всего поступающего битового потока. Он включает активное видео, последовательности управляющих преамбул и служебные данные, которые могут передаваться во интервалах обратного хода кадровой и строчной развёртки.

Устройство интерфейса внешней шины

Устройство интерфейса внешней шины (EBIU) процессора реализует интерфейс с разнообразными устройствами памяти индустриального стандарта. Контроллер состоит из контроллера SDRAM и контроллера асинхронной памяти.

Контроллер SDRAM PC133

Контроллер SDRAM обеспечивает интерфейс с одним банком устройств SDRAM индустриального стандарта или DIMM. Банк, полностью совместимый со стандартом SDRAM PC133, может конфигурироваться на объём от 16 до 128 Мбайт памяти.

Для поддержки банком SDRAM более медленных устройств памяти имеется набор программируемых временных параметров. Банк памяти имеет разрядность 16 разрядов, что позволяет минимизировать количество устройств и снижает стоимость системы.

Асинхронный контроллер

Контроллер асинхронной памяти обеспечивает конфигурируемый интерфейс до четырёх отдельных банков памяти или устройств ввода/вывода. Временные параметры каждого банка могут независимо программно устанавливаться. Это свойство позволяет осуществлять интерфейс с разнообразными устройствами памяти, включая SRAM, ПЗУ и флэш-EPROM, а также устройствами ввода/вывода, имеющими интерфейс со стандартными линиями управления памятью. Каждый банк занимает окно размером 1 Мбайт в адресном пространстве процессора. Если окна заняты не полностью, контроллер памяти не делает их непрерывными. Банки имеют разрядность 16 разрядов, что позволяет организовать интерфейс с рядом устройств памяти и устройств ввода/вывода.

Поддержка DMA

Процессор имеет несколько независимых контроллеров DMA, поддерживающих автоматизированные передачи данных с минимальным вмешательством ядра. Передачи DMA могут осуществляться между блоками внутренней памяти и любым из периферийных устройств процессора, поддерживающих DMA. Кроме того, передачи DMA могут осуществляться между любым периферийным устройством, поддерживающим DMA, и внешними устройствами, подсоединёнными к интерфейсам внешней памяти, включая контроллер SDRAM и контроллер асинхронной памяти. К периферийным устройствам, поддерживающим DMA, относятся последовательные порты, порт SPI, UART и PPI. Каждое отдельное периферийное устройство, поддерживающее DMA, имеет, по меньшей мере, один выделенный канал DMA.

Контроллер DMA поддерживает и одномерные (1D) и двумерные (2D) передачи DMA. Инициализация передач DMA может осуществляться либо при помощи регистров, либо при помощи набора параметров, называемых блоками дескрипторов.

Функция двумерного DMA поддерживает произвольные размеры строк и столбцов до 64К элементов на 64К элементов, и произвольные размеры шагов по строкам и столбцам до +/- 32К элементов. Более того, размер шага по столбцу может быть меньше размера шага по строке, что позволяет реализовать перемежение потоков данных. Это свойство особенно полезно в видеоприложениях, в которых может осуществляться разделение данных на лету.

Примеры поддерживаемых типов DMA:

- операции с одиночным линейным буфером, обращения к которому прекращаются по завершению передачи;
- передачи с использованием циклического, автоматически обновляемого буфера, по заполнению или частичному заполнению которого вызывается прерывание;
- одномерный и двумерный DMA с использованием связанного списка дескрипторов;
- двумерный DMA с использованием массива дескрипторов, определяющих только базовый адрес DMA в пределах общей страницы.

В дополнение к выделенным каналам DMA периферийных устройств, также имеется отдельный канал DMA типа “память-память”, предназначенный для передач между различными устройствами памяти системы. Его использование позволяет передавать блоки данных между любыми устройствами памяти – включая внешние SDRAM, SRAM, ПЗУ и флэш-память – с минимальным вмешательством процессора.

[К блок-схеме](#)

[В начало](#)

Методы загрузки

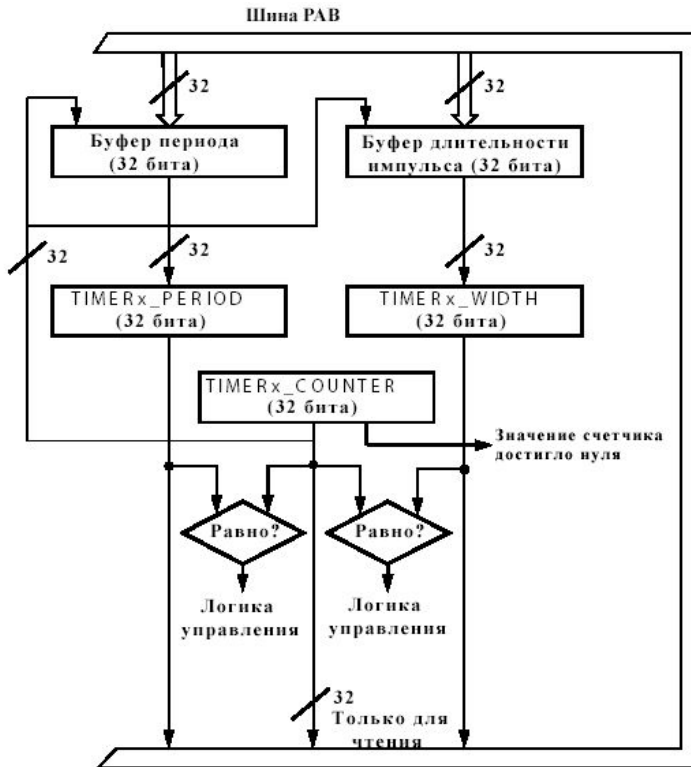
Внутреннее загрузочное ПЗУ содержит небольшое загрузочное ядро, которое можно либо не использовать, либо использовать для загрузки пользовательского кода из внешней памяти. Дополнительную информацию см. в таблице 4-10 “Адреса вектора сброса”. Загрузочное ядро выполняет чтение состояния выводов $VMODE[1:0]$ в момент сброса, идентифицируя источник загружаемого кода (см. таблицу 4-7). При работе в режиме загрузки 0 процессор настраивается на выполнение программы из 16-разрядной внешней памяти по адресу $0x2000\ 000$ (Банк $ASYNC0$).

Существует несколько режимов загрузки пользовательского кода из внешней памяти. В этих режимах загрузочное ядро производит настройку выбранного периферийного устройства в соответствии со значениями выводов $VMODE[1:0]$.

В каждом из режимов загрузки пользовательский код, считываемый из внешней памяти, помещается в начало памяти L1. В соответствии с информацией, содержащейся в заголовке файла загрузчика, может производиться чтение дополнительных разделов во внутреннюю память. Загрузочное ядро прекращает процесс загрузки при выполнении перехода к началу пространства памяти команд L1. Затем процесс начинает выполнение программы с этого адреса.

Таймеры общего назначения

Каждый таймер общего назначения имеет один выделенный двунаправленный вывод, TMRx. Этот вывод является входным в режиме PWM_OUT и выходным в режимах WDTN_CAP и EXT_CLK. Для обеспечения работы в этих режимах, каждый таймер имеет четыре регистра. Регистры счётчика (TIMERx_COUNTER), периода (TIMERx_PERIOD) и длительности импульса (TIMERx_WIDTH) имеют разрядность 32 бита, что повышает диапазон значений и точность таймера. Устройство таймера показано на рис. 2



Каждый таймер общего назначения включает следующие регистры:

- регистр конфигурации (TIMERx_CONFIG);
- регистр счётчика (TIMERx_COUNTER);
- регистр периода (TIMERx_PERIOD);
- регистр длительности импульса (TIMERx_WIDTH).

Рис. 2. Блок-схема

таймера

Назад

К блок-схеме

В начало

На рис. 3 приведена блок-схема SPI-интерфейса. Интерфейс представляет собой сдвиговый регистр, который последовательно передает данные в другое SPI-устройство или принимает данные из другого SPI-устройства (по одному биту за такт SCK). Приём и передача данных при помощи сдвигового регистра осуществляются одновременно. При выполнении передач через SPI передаваемые данные последовательно сдвигаются на выход через сдвиговый регистр, а принимаемые данные последовательно сдвигаются на вход на другом конце того же самого сдвигового регистра. Сигнал SCK служит для синхронизации сдвигаемых и выбираемых данных на двух выводах данных.

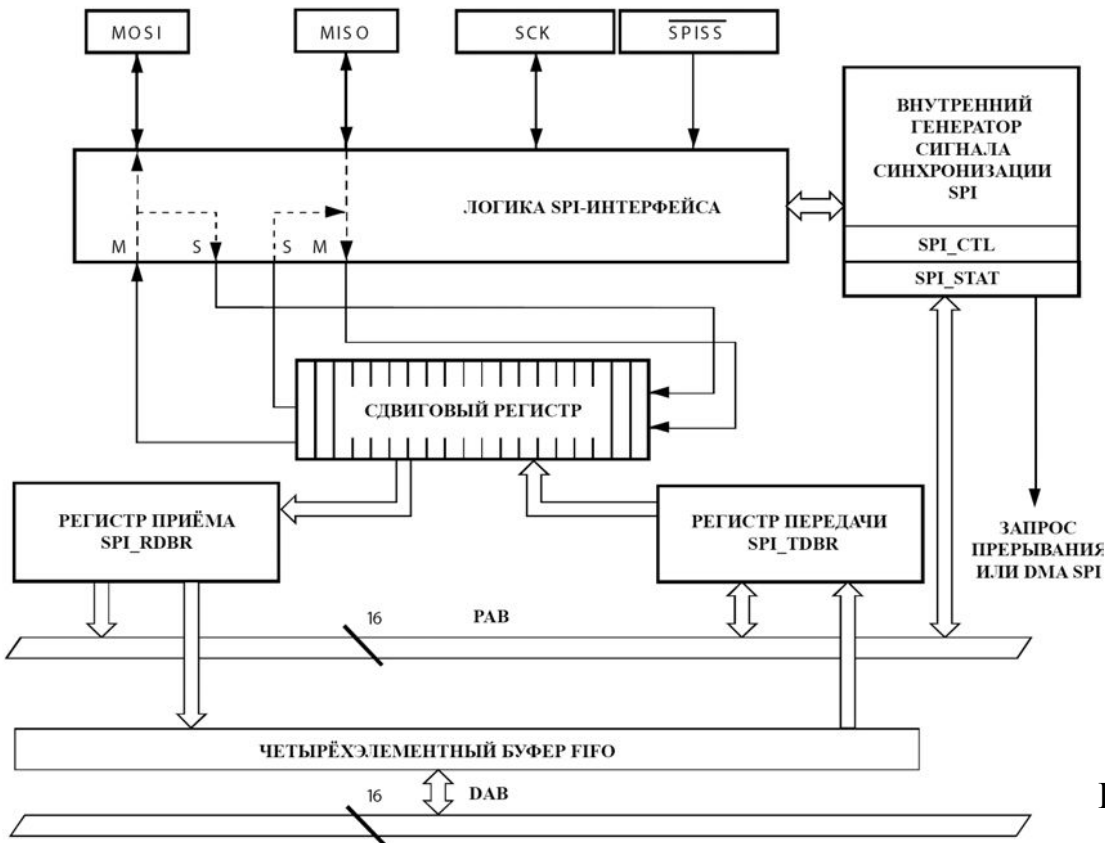


Рис. 3. Блок-схема SPI

Обзор

EBIU обслуживает запросы на обращение к внешней памяти, поступающие от ядра или канала DMA. Приоритет запросов определяется контроллером внешней шины. В зависимости от адреса запрос обслуживается контроллером SDRAM EBIU или контроллером асинхронной памяти EBIU.

EBIU тактируется тактовым сигналом системы (SCLK). Все устройства синхронной памяти, подключаемые к процессору, работают с частотой SCLK. Отношение частоты ядра к частоте SCLK задаётся программно при помощи регистров схемы фазовой автоподстройки частоты (PLL), отображённых в карте памяти.

Пространство внешней памяти показано на рис.4 (перейди по ссылке!!!). Одна область выделена для поддержки SDRAM. Временные параметры интерфейса SDRAM и размер области SDRAM изменяются программно. Размер пространства SDRAM может изменяться в диапазоне от 16 до 128 Мбайт. Начальный адрес пространства памяти SDRAM – 0x0000 0000. Область от конца пространства памяти SDRAM до адреса 0x2000 0000 зарезервирована.

Следующие четыре области выделены для поддержки устройств асинхронной памяти. Параметры каждой области асинхронной памяти настраиваются независимо. Это позволяет процессору поддерживать устройства памяти с различными характеристиками. EBIU имеет отдельные выходные выводы выбора микросхем памяти для каждой области асинхронной памяти.

Следующая область является зарезервированной. Обращения к этой области не вызывают выполнения транзакций внешней шины: операции записи не влияют на значения внешней памяти, при чтении возвращаются неопределённые значения. При этом EBIU генерирует отклик ошибки на внутренней шине, который, в свою очередь, вызывает аппаратное исключение доступа ядра или (необязательно) прерывание канала DMA.

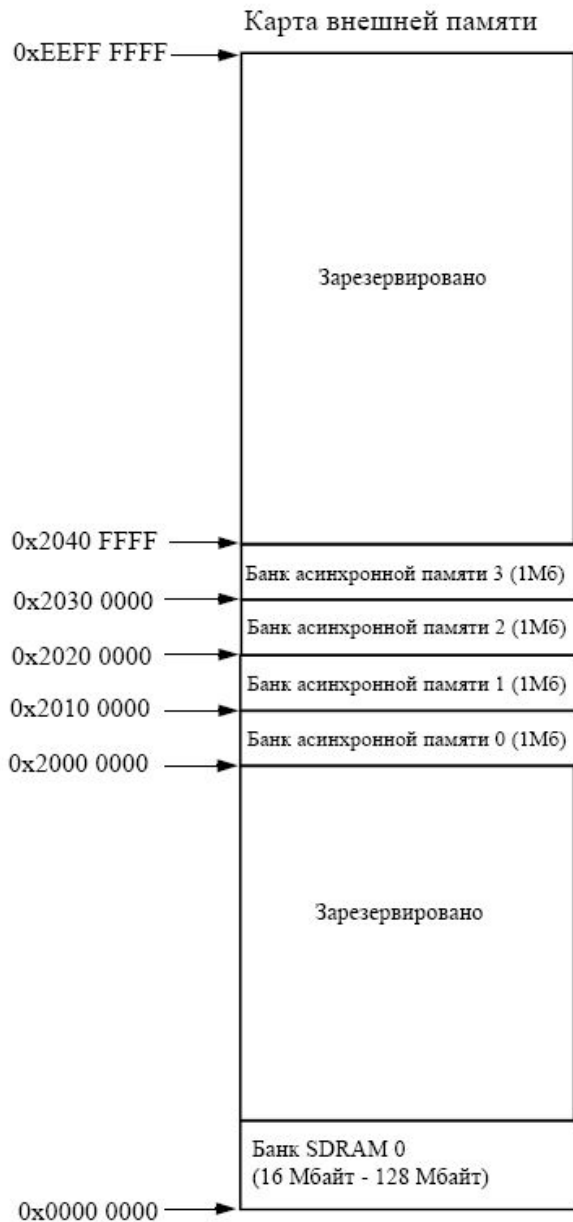


Рис. 4. Карта внешней памяти.

Примечание: Все области памяти, не отмеченные как зарезервированные, могут адресоваться и ядром и системой

Блок-схема

На рис.5 показана концептуальная блок-схема EBIU и его интерфейсов. Названия сигналов, имеющих активный низкий уровень, сопровождаются чертой сверху.

Так как в один момент времени может выполняться обращение только к одному внешнему устройству памяти, управляющие, адресные выходы и выходы данных всех устройств мультиплексируются на выводах процессора. Арбитраж совместно используемых выводов осуществляется контроллером асинхронной памяти (AMC, Asynchronous Memory Controller) и контроллером SDRAM (SDC, SDRAM Controller)

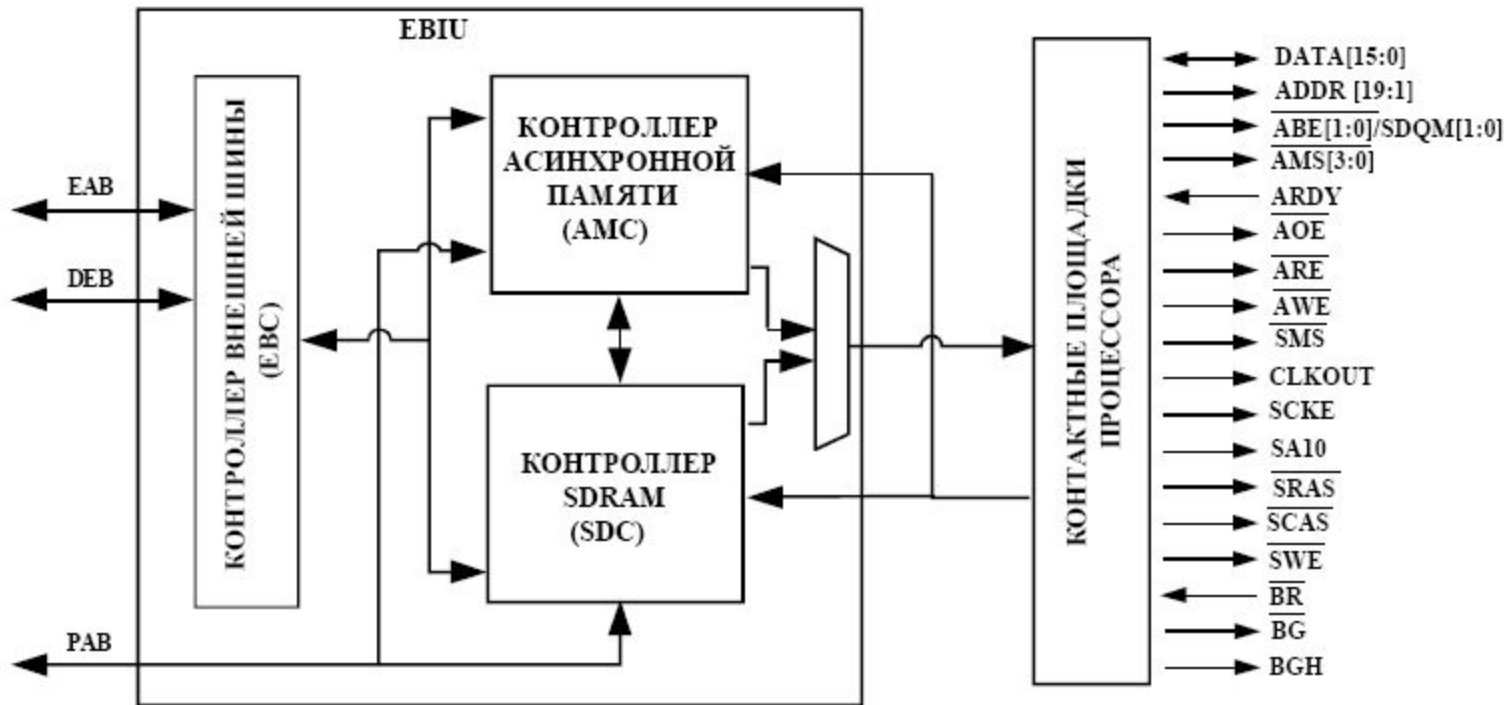


Рис. 5. Устройство интерфейса внешней шины (EBIU)

[Назад](#)

[К блок-схеме](#)

[В начало](#)

ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Вычислительные устройства процессора выполняют обработку данных в алгоритмах ЦОС и общего управления. Процессор содержит шесть вычислительных устройств: включают два 40-разрядных арифметико-логических устройства (АЛУ), два 16-ти разрядных умножителя с 40-ным накопителем, 40-разрядный кольцевой сдвигатель и набор видео АЛУ. Эти устройства получают данные из регистров регистрового файла данных. Вычислительные команды этих устройств обеспечивают выполнение операций с фиксированной точкой, каждая вычислительная команда может быть выполнена за один такт работы процессора. Обработываемые данные могут быть 8-ми, 16-ти или 32-битными.

Вычислительные устройства выполняют различные типы операций. АЛУ выполняют арифметические и логические операции над 16-ти и 32 битными данными. Умножители выполняют умножение 16- битовых операндов и операции умножения/сложения и умножения/вычитания. Устройство сдвига выполняет операции логического и арифметического сдвига, упаковки и извлечения битов. Видео АЛУ выполняют логические SIMD (Single Instruction Multiple Data) операции над определёнными 8-разрядными операндами.

Вычислительные устройства принимают и передают данные через регистровый файл данных, состоящий из восьми 32-разрядных регистров. В операциях, требующих использования 16-разрядных операндов, регистры разбиваются на пары, обеспечивая возможность использования шестнадцати 16-разрядных регистров.

[Архитектура ядра](#)

[К блок-схеме](#)

[В начало](#)

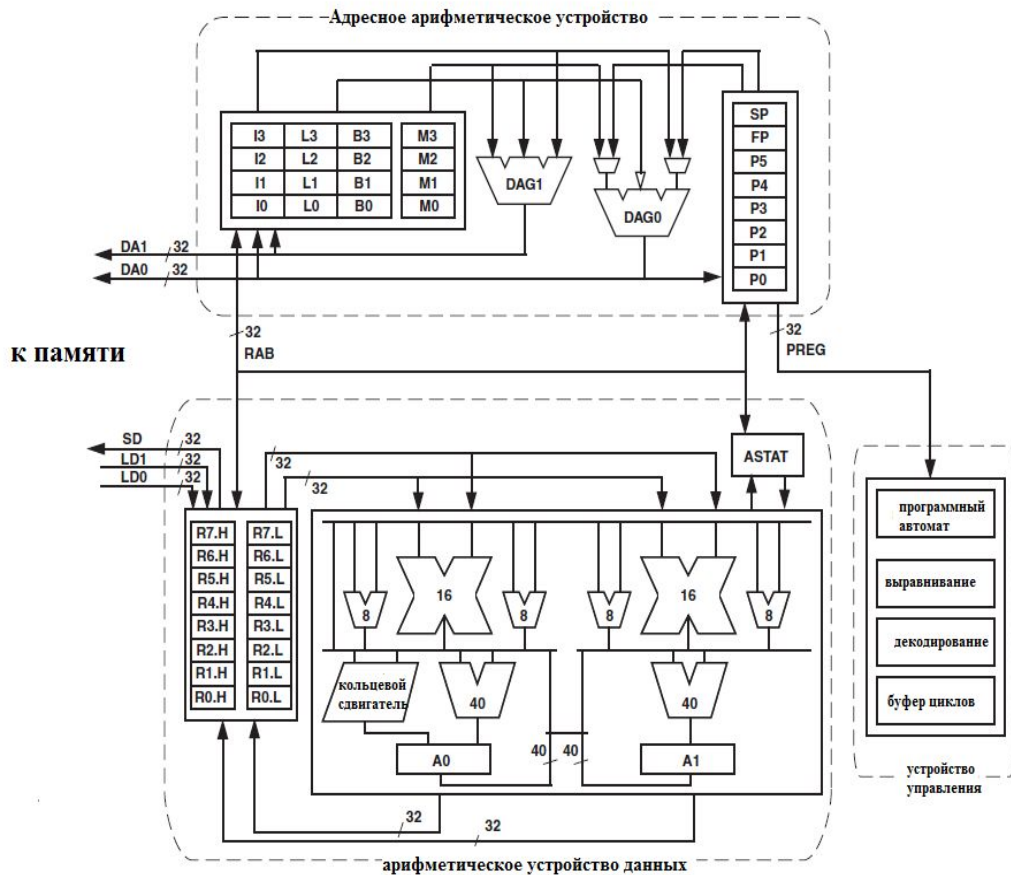


Рис.6. Архитектура ядра процессора

Простые (однофункциональные) команды умножителя, АЛУ и устройства сдвига имеют неограниченный доступ к регистрам регистрового файла данных. Многофункциональные операции могут иметь ограничения, описываемые в разделе, посвящённом данному режиму вычислений. Два дополнительных регистра, A0 и A1, содержат 40-разрядные результаты аккумулятора. Эти регистры являются выделенными регистрами АЛУ и предназначены в основном для реализации умножения-накопления.

Традиционные режимы арифметических операций, такие как целочисленный и дробный, явно указываются в команде. Режимы округления устанавливаются в регистре ASTAT, в который также записываются флаги состояния/условия результатов вычислительных операций.

Назад

Регистровые
файлы

К блок-схеме

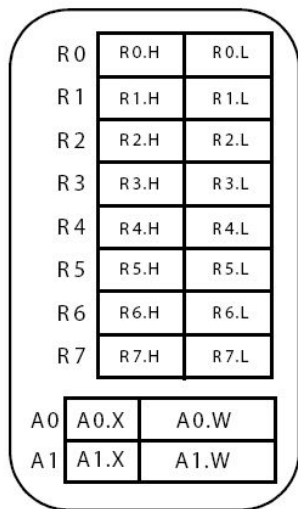
В начало

Регистровые Файлы

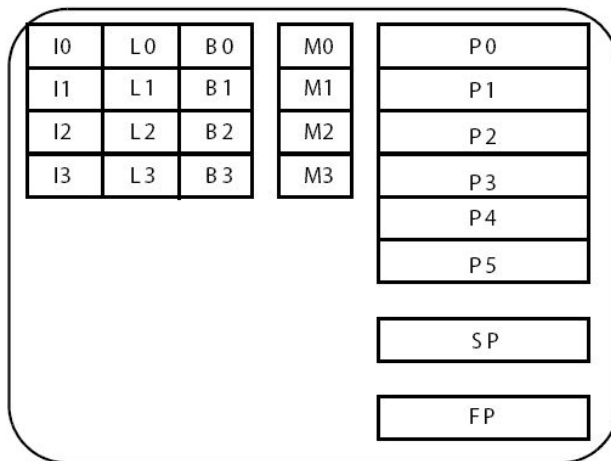
Вычислительные устройства процессора имеют три категории групп регистров – регистровый файл данных, регистровый файл указателей и набор регистров генератора адреса данных (DAG):

- регистровый файл данных принимает операнды для вычислительных устройств по шине данных, и в него помещаются результаты вычислений;
- регистровый файл указателей содержит указатели, используемые в операциях адресации;
- регистры DAG представляют собой набор выделенных регистров, предназначенных для организации циклических буферов с нулевыми непроизводительными затратами в операциях ЦОС. Регистровые файлы процессора показаны на рис. 7.

РЕГИСТРЫ ДАННЫХ



РЕГИСТРЫ ГЕНЕРАТОРА АДРЕСА ДАННЫХ (DAGs)



I- индексные регистры, указатели адреса.
M- регистры модификаторы
B- регистры базового (начального) адреса
L- регистры длины циклического буфера адресов.

Рис. 7. Регистровые файлы.

Память команд L1

Память команд L1 состоит из комбинации выделенной SRAM и банков, которые могут конфигурироваться как SRAM или как кэш. Биты управления в регистре IMEM_CONTROL могут использоваться для организации четырех подуровней банков памяти L1 в 16 Кбайтном банке, конфигурируемом как кэш или SRAM, следующим образом:

- Простая SRAM.
- 4-входовый наборно-ассоциативный кэш команд.
- Кэш с возможностью запираения до 4-х входов.

Память команд L1 может использоваться только для хранения команд.

Регистр управления памятью команд (IMEM_CONTROL) содержит биты управления памятью команд L1. По умолчанию после сброса кэш и проверка адресов в ассоциативном буфере защиты и кэширования (CPLB) запрещены.

Когда установлен (равен 1) бит LRUPRIORST, сбрасываются все биты состояний кэширования CPLB_LRUPRIO (см. раздел «Регистры данных ICPLB (ICPLB_DATAx)). При этом также все кэшируемые строки принудительно переводятся в состояние равной (низкой) значимости. Политика замещения кэша основывается, в первую очередь, на значимости строки, указываемой битами CPLB_LRUPRIO, а затем на алгоритме LRU (поиск строки, которая дольше всех не использовалась, least recently used). Подробности см. в разделе «Запирание кэша команд по строкам». Для разрешения записи в биты CPLB_LRUPRIO при кэшировании новых строк бит LRUPRIORST должен иметь значение 0.

Регистр управления памятью команд L1 (IMEM_CONTROL)

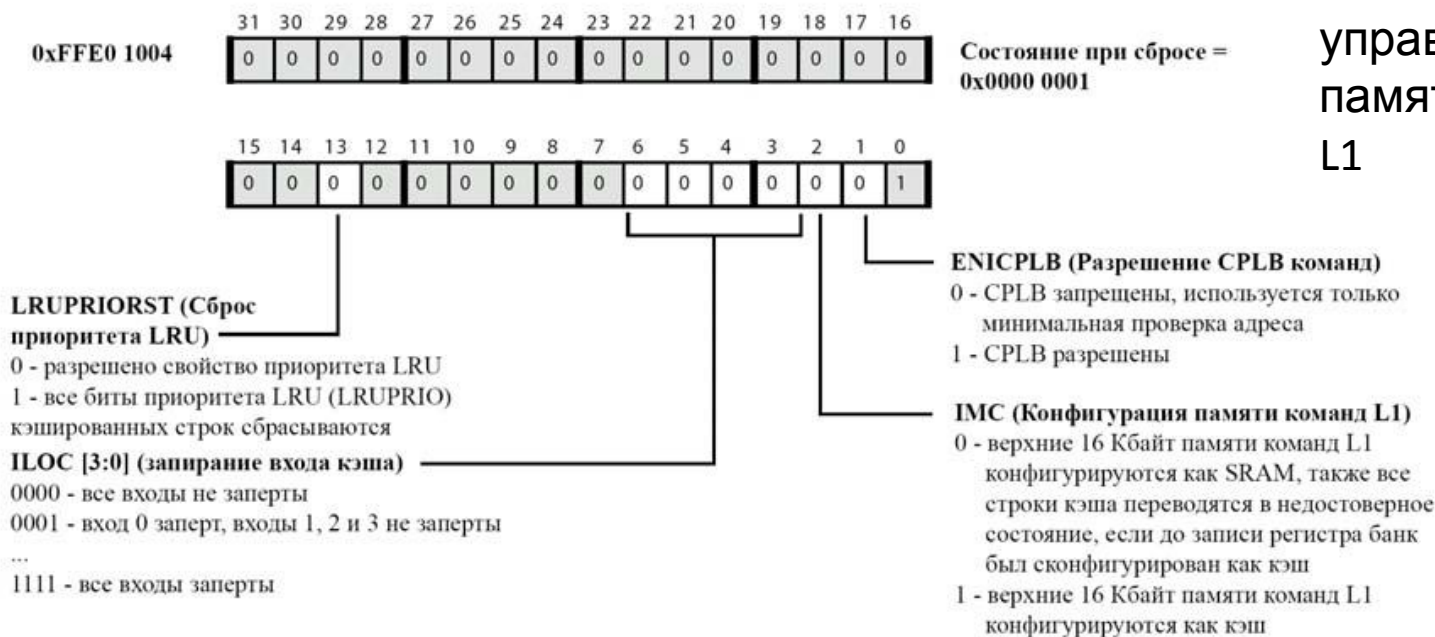


Рис. 7. Регистр управления памятью команд L1

Память данных L1

SRAM/кэш данных L1 состоит из однопортовых подразделов, однако организован таким образом, чтобы снизить вероятность конфликтов доступа. Эта организация приводит к очевидному многопортовому характеру работы. В отсутствие конфликтов за один такт ядра могут выполняться следующие операции с данными в L1:

- две 32-разрядных загрузки с использованием DAG;
- одно конвейерное 32-разрядное сохранение в память с использованием DAG;
- одна 64-разрядная операция ввода/вывода с использованием DAG;
- один 64-разрядный доступ типа жертвования/заполнения строки кэша.

Память данных L1 может использоваться только для хранения данных.

Регистр управления памятью данных (DMEM_CONTROL) содержит биты управления памятью данных L1.

Регистр управления памятью данных (DMEM_CONTROL)

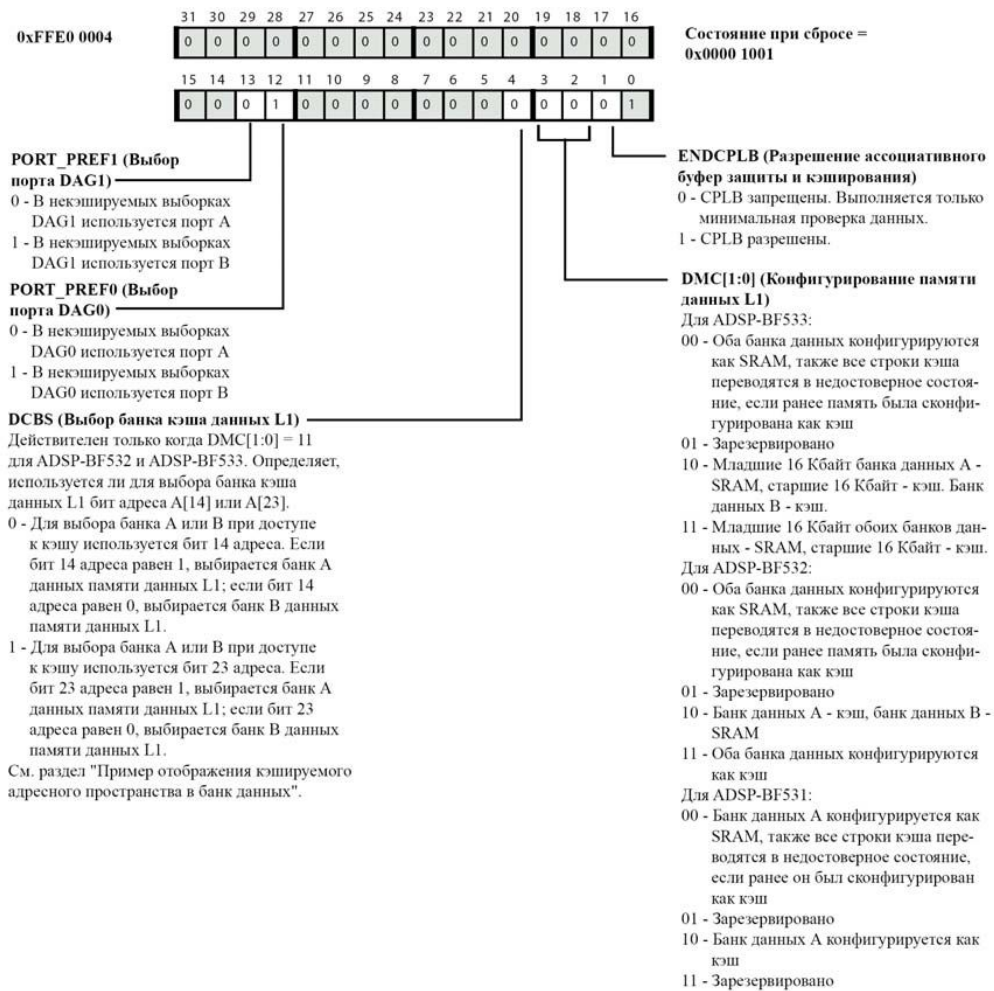


Рис.8. Регистр управления памятью данных.

MMU

Процессор имеет несколько блоков памяти L1, которые могут быть сконфигурированы как смешанный набор SRAM и кэша. Устройство управления памятью (MMU, Memory Management Unit) обеспечивает защиту памяти при выполнении ядром индивидуальных задач и может осуществлять защиту системных регистров от непреднамеренного доступа.