

AN4839: кэш первого уровня на STM32F7



Добавил(а) microsin

В Cortex®-M7 модуль **MPU** (Memory Protection Unit, блок защиты памяти [2]) позволяет модифицировать атрибуты кэша первого уровня (**L1**) по областям. Управление кэшем осуществляется при помощи регистра управления кэша, но MPU может определить режим кэша и будет ли доступ области кэшированным или нет. В некоторых случаях кэшируемые системы должны гарантировать согласованность данных между ядром и основной памятью, когда происходит взаимодействие с совместно используемыми данными.

Здесь приведен перевод апноута **AN4839** [1] (автор перевода Marat Galyamov), который описывает поведение кэша L1 и показывает пример, как гарантировать согласованность данных при использовании кэша в серии **STM32F7**.

Дополнительная информация про MPU и атрибуты памяти

Для получения дополнительной информации о блоке MPU, а также о том, как установить атрибуты памяти согласно типу памяти и политике кэша, обратитесь к документам, доступным на www.st.com:

- STM32F7 Series Cortex®-M7 processor programming manual (PM0253)
- Managing memory protection unit (MPU) in STM32 MCUs (AN4838) [2]

[Управление кэшем]

Контроллеры семейства STM32F7 могут иметь кэш L1 для команд и данных объемом 4 Кбайт, 8 Кбайт или 16 Кбайт. Кэш L1 хранит данные/команды рядом с CPU и таким образом CPU не нужно выбирать те же данные, которые используются неоднократно, например, как в небольшом цикле. Рис. 1 показывает архитектуру системы серии STM32F7.

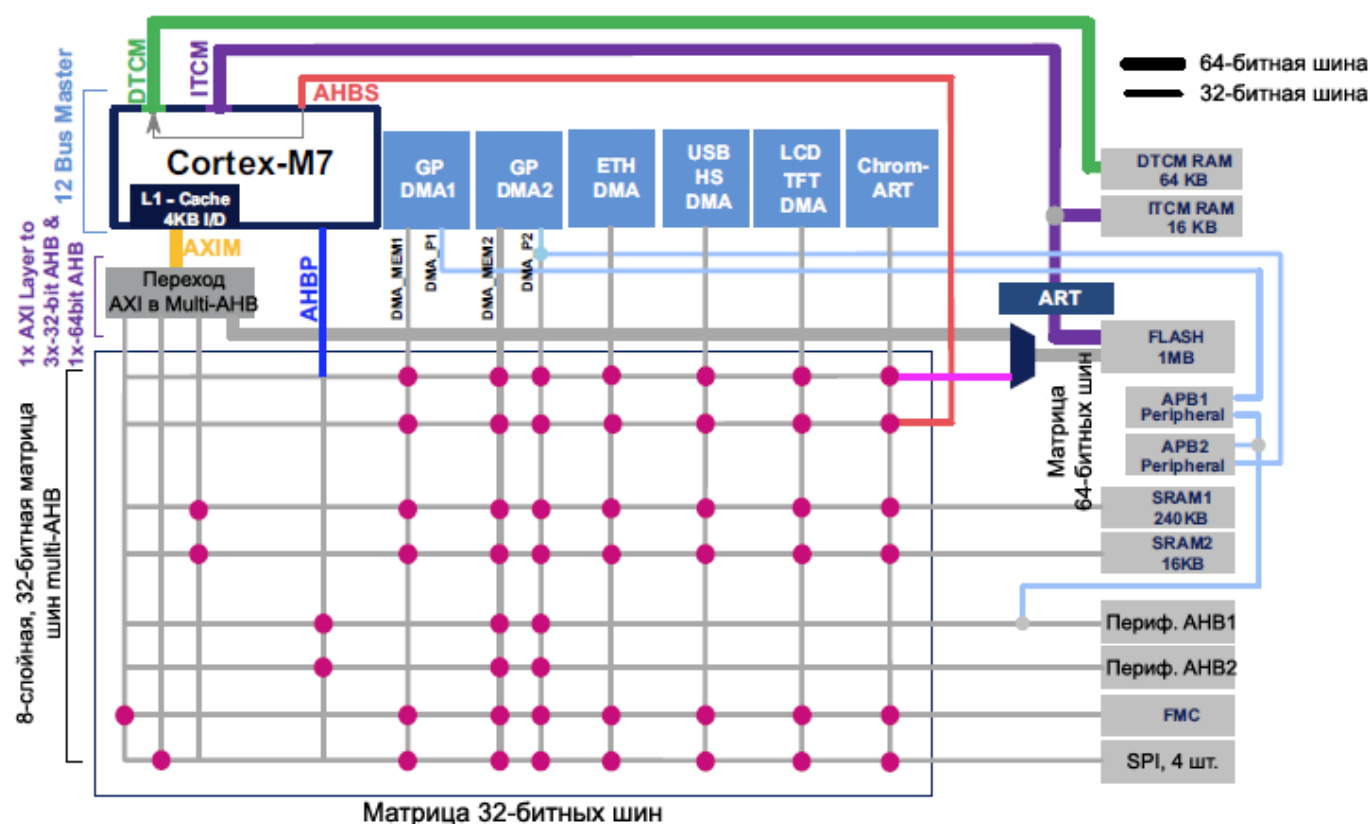


Рис. 1. Архитектура системы серии STM32F7.

Доступы памяти к подсистеме могут занимать множество циклов (особенно в интерфейсах внешней памяти с состояниями ожидания нескольких событий). Кэши нужны, чтобы ускорить операции чтения-записи в память. Идея состоит в том, что обе операции могут быть оптимизированы, если данные доступны локально (в области, которая требует только один цикл доступа).

Доступ шины к подсистеме памяти, который занимает более одного цикла процессора для выполнения, отличается от конвейерного выполнения потока команд процессора. Это позволяет достичь высокой скорости. Кэш обычно реализуется набором строк, где строка - просто короткий сегмент памяти. Число строк в наборе называют X-канальной ассоциативностью. Это свойство установлено на аппаратном уровне.

При чтении получают расположение в памяти только в первый раз, когда получают доступ к расположению. При записи значение будет записано сразу (сквозная запись) или будет размещено в кэше для более поздней записи (режим обратной записи). У каждого режима есть свои плюсы/минусы в отношении производительности, которые должны быть учтены, учитывая специфику приложения.

В случае с памятью с обратной записью (write-back, WB), если строка кэша отмечена как "грязная", то запись выполняется только в интерфейсе AXIM, когда строка выгружается. Когда "грязная" строка кэша выгружена, данные передаются в буфер записи в интерфейс AXIM, который будет записан в систему внешней памяти.

Кэш L1 на всех Cortex®-M7 разделен на строки по 32 байта. Каждая строка имеет тэг с адресом. Кэш данных имеет ассоциативность 4 канала (4 строки в наборе), кэш команд имеет ассоциативность 2 канала. Это аппаратный компромисс, чтобы не присваивать тэг каждой строке с адресом.

Удачное обращение в кэш (cache hit) - когда адрес попадает в произвольное место на данном наборе строк. Таким образом, аппаратные средства должны делать меньше операций сравнения, чтобы узнать, кэширован ли адрес. Если обращение успешное, то значение кэша используется для чтения, или значение сохраняется для записи. Если обращение неудачное (промах, cache miss), то выделяется новая строка и присваивается тэг, и кэш заполняется либо от доступа на чтение, либо от доступа на запись. Если все строки уже выделены (заполнены), контроллер кэш-памяти выполняет процесс замещения строки, в котором выбранная строка (зависит от алгоритма замещения) очищается/аннулируется, и происходит перераспределение. Кэш данных и кэш команд реализуют псевдослучайный алгоритм замещения.

Кэш L1 может увеличить производительность, когда используется в сочетании с интерфейсами памяти на шине AXI. Не нужно путать с памятью с интерфейсом Tightly Couple Memory (TCM), которая не является кэшируемой. Любая обычная область памяти может быть кэшируемой, как описано выше, но лучший результат будет у памяти, к которой осуществляется доступ через шину AXI, такие как: внутренняя Флэш-память, внутренняя SRAM и внешняя память, связанная через модуль FMC или контроллеры Quad-SPI.

Есть четыре основных операции с кэшем: включение, отключение, очистка и аннулирование. Для этих операций доступны специальные функции API Cube STM32F7, что уменьшает время разработки.

[Доступ к настройке кэша Cortex®-M7 с использованием CMSIS]

Функции CMSIS кэша определены в файле core_cm7.h как показано в Табл. 1.

Таблица 1. CMSIS функции кэша.

CMSIS функции	Описание
void SCB_EnableICache (void)	Аннулирование и затем включение кэша команд
void SCB_DisableICache (void)	Отключение кэша команд и аннулирование данных в нем
void SCB_InvalidateICache (void)	Аннулирование кэша команд
void SCB_EnableDCache (void)	Аннулирование и затем включение кэша данных
void SCB_DisableDCache (void)	Отключение кэша данных, очистка и аннулирование его содержимого
void SCB_InvalidateDCache (void)	Аннулирование кэша данных
void SCB_CleanDCache (void)	Очистка кэша данных
void SCB_CleanInvalidateDCache (void)	Очистка и аннулирование кэша данных

Очистка кэша: операция обратной записи "грязной" строки кэша в память (такую операцию иногда называют синхронизацией или сбросом, flush).

Аннулирование кэша: операция помечает содержимое кэша как недостоверное (обычно, это операция удаления).

[Работа с кэшем]

В большинстве случаев использовать кэш несложно. Необходимо установить область в модуле MPU и включить кэш с помощью функций CMSIS, указанных ниже. Например, можно сделать настройки, используя обратную или сквозную запись.

Обратная запись (write-back, WB): кэш не записывает содержимое в память, пока не завершится операция очистки.

Сквозная запись (write-through, WT): запускает запись в память как только содержимое будет записано в строку кэша. Это более безопасно для согласованных данных, но требует большего доступа к шине. На практике запись в память сделана в фоновом режиме и имеет небольшой эффект, если к этому же содержимому кэша не осуществляется очень быстрый и постоянный доступ. Таким образом, это всегда компромисс.

[STM32F7 настройки по умолчанию]

По умолчанию модуль MPU отключен. В этом случае настройки кэша определены как таблица адресов по умолчанию.

Таблица. 2. Область памяти, общий доступ и политики кэша.

Диапазон адресов	Область памяти	Тип памяти	Совместное использование	Политика кэша
0x00000000-0x1FFFFFFF	Code	Обычная (Normal)	Нет	WT
0x20000000-0x3FFFFFFF	SRAM			WBWA
0x40000000-0x5FFFFFFF	Периф. устройства	Device	Нет	-
0x60000000-0x7FFFFFFF	Внешнее RAM	Обычная (Normal)	Нет	WBWA
0x80000000-0x9FFFFFFF				WT
0xA0000000-0xBFFFFFFF	Внешнее устройство	Device	Да	-
0xC0000000-0xDFFFFFFF			Нет	
0xE0000000-0xE00FFFFF	Частная шина периф. устройств	Строго упорядоченная на доступ	Нет	-
0xE0100000-0xFFFFFFFF	Система, зависящая от производителя	Device	Нет	-

[Пример настройки кэша и согласованности данных]

Цель этого примера состоит в том, чтобы ознакомиться с когерентностью кэш-памяти данных ARM® Cortex®-M7.

Сначала CPU копирует 128 константных байт из Флэш-памяти aSRC_Const_Buffer во временный буфер pBuffer SRAM1.

Затем CPU конфигурирует и включает DMA для совершения передачи типа память-память и копирования из SRAM1 pBuffer в буфер назначения aDST_Buffer, определенный в DTCM RAM.

Наконец, CPU сравнивает данные, прочитанные DMA из aDST_Buffer с шаблоном констант из Флэш-памяти aSRC_Const_Buffer.

Рис. 2 показывает схему передачи данных.

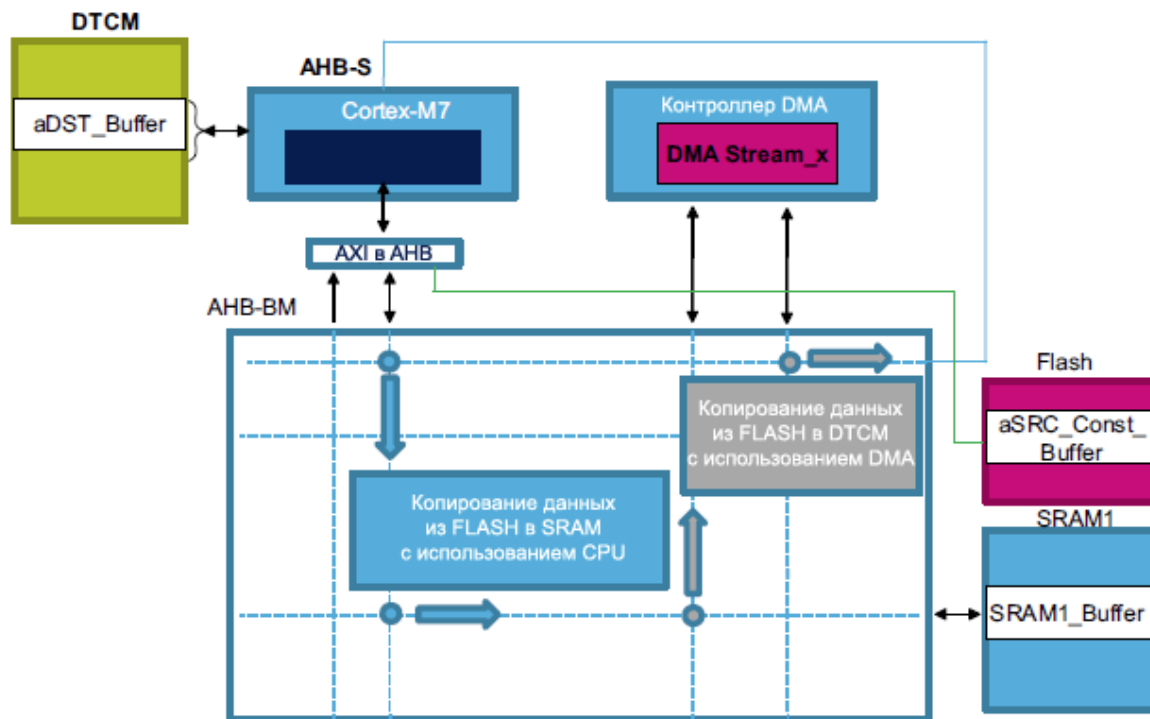


Рис. 2. Пути перемещения данных.

Цель состоит в том, чтобы показать влияние на согласованность данных между CPU и DMA при доступе к кэшируемой области памяти с атрибутом обратной записи (write-back).

По умолчанию, после сброса, кэш данных и команд отключен. Когда кэш данных отключен, передача данных между SRAM1 и RAM DTCM будет завершена успешно в схеме, описанной выше.

При включении кэша данных прежде, чем выполнится описанный сценарий передачи, будет обнаружено несоответствие данных между данными в aDST_Buffer (целевой буфер DMA в DTCM) и aSRC_Const_Buffer (буфер источника данных CPU во Флэш-памяти).

При использовании кэша L1 всегда остается проблема, иногда называемая когерентностью кэша. Этот вопрос возникает, когда мультиплексированные мастер- устройства (CPU, DMA...) используют память совместно. Если CPU пишет что-то в область, у которой есть атрибут кэша с обратной записью (пример SRAM1), результат записи не будет виден в SRAM, поскольку доступ буферизован, и затем, если DMA считывает ту же область памяти, чтобы выполнить передачу данных, то прочитанное значение не будет соответствовать ожидаемым данным.

Решение 1: выполнять операцию по обслуживанию кэша после записи данных в кэшируемую область, программно вызывая операцию очистки кэша данных функцией CMSIS SCB_CleanDCache() (все "грязные" строки будут обратно записаны в SRAM1).

Решение 2: выполнять операцию по обслуживанию кэша, изменяя атрибуты MPU памяти SRAM1 от обратной записи (по умолчанию) к политике сквозной записи.

Решение 3: изменяя атрибуты MPU памяти SRAM1 используя общие атрибуты. Это будет по умолчанию препятствовать кэшированию SRAM1 в кэше данных.

Решение 4: выполнять операцию по обслуживанию кэша, вызывая политику сквозной записи для всех операций записи. Это можно сделать, включив сквозную запись через установку бита кэша данных (D-Cache bit) в регистре CACR.

Согласованность (последовательность) данных между ядром и DMA обеспечивается следующими методами:

1. Буферы SRAM1 делаются не кэшируемыми.
2. Включается кэширование буферов SRAM1 с поддержкой обратной записи, при этом согласованность обеспечивается программно (очистка или аннулирование кэша данных).
3. Изменением атрибута области MPU в SRAM1 на совместную область.
4. Включается кэширование буферов SRAM1 с поддержкой сквозной записи.

Другой случай - когда DMA пишет в SRAM1, и CPU собирается считать данные из SRAM1. Чтобы гарантировать согласованность данных между кэшем и SRAM1, программное обеспечение должно аннулировать кэш перед чтением обновленных данных из SRAM1.

За дополнительной информацией по настройке кэша обратитесь к документу (PM0253) STM32F7 Series Cortex®-M7 processor programming manual.

[Подсказки и советы]

- После сброса надо аннулировать каждый кэш перед его включением, в противном случае поведение может быть непредсказуемым.
- При отключении кэша данных необходимо очищать область кэша, чтобы быть уверенным в том, что любые "грязные" данные сброшены во внешнюю память.
- Перед включением кэша данных необходимо аннулировать область кэша данных, если внешняя память может иметь изменения поскольку кэш был отключен.
- Перед включением кэша команд необходимо аннулировать область кэша команд если внешняя память может иметь изменения поскольку кэш был отключен.
- Если ПО использует кэшируемую область памяти для буферов приема/передачи DMA, оно должно вызвать очистку кэша перед запуском операций DMA, чтобы гарантировать, что все данные подтверждены в подсистеме памяти. После завершения DMA-передачи, при чтении данных из периферии, ПО должно аннулировать кэш перед чтением-DMA обновленной области памяти.
- Всегда лучше использовать некешированную область для буферов DMA. ПО может использовать MPU для установки некешированного блока памяти, чтобы использовать как общую память между CPU и DMA.
- Не включайте кэш для памяти, которая интенсивно используется для операций DMA.
- Когда используется акселератор ART, CPU может прочитать команду всего за 1 такт из внутренней флэш-памяти (например, 0-wait state). Таким образом, кэш команд не может быть использован для внутренней флэш памяти.
- При использовании NOR Flash обратная запись вызывает проблемы, так как команды Erase и Write не отправляются во внешнюю флэш память.
- Если подключенное устройство является обычной памятью, может быть полезным чтение кэша данных (D-cache). Однако, если внешнее устройство является ASIC и/или FIFO, необходимо отключить кэш данных для чтения.

[Ссылки]

1. [AN4839 Level 1 cache on STM32F7 Series site:st.com](http://www.st.com/AN/4839).
2. [AN4838: модуль управления защитой памяти STM32](#).