

Вступление

Устройства серии STM32F7 являются первыми 32-разрядными микроконтроллерами на базе ARM® Cortex®-M7. Используя преимущества ускорителя ART от ST, а также L1-кеша, устройства серии STM32F7 обеспечивают максимальную теоретическую производительность Cortex®-M7.

Результаты тестов стабильно достигают 1082 CoreMark и 462 DMIPS, независимо от того, выполняется ли код из встроенной флэш-памяти, из внутренней оперативной памяти или из внешней памяти (SRAM, SDRAM или флэш-память Quad-SPI).

Устройства серии STM32F7 обеспечивают высокий уровень производительности благодаря:

- Мощному суперскалярному конвейеру и возможности DSP, обеспечивающие быстрый отклик в реальном времени с низкой задержкой прерывания.
- Эффективный доступ к большим объемам внешней памяти
- Высокая производительность операций с плавающей запятой для сложных вычислений.

В этой заметке по применению представлена глобальная архитектура STM32F7, а также интерфейсы и функции памяти, которые обеспечивают высокую гибкость для достижения наилучшей производительности и дополнительных размеров кода и данных. Он также представляет архитектуру с несколькими мастерами, которая способствует повышению производительности системы и разгружает ЦП.

В примечании к применению также представлена программная демонстрация производительности архитектуры устройств серии STM32F7 в различных конфигурациях разделения памяти (разный код и расположение данных), а также производительность архитектуры, в которой разрешены DMA.

Это примечание по применению предоставляется со встроенным программным пакетом X-CUBE-32F7PERF, который включает два проекта:

- Проект Stm32f7_performances направлен на демонстрацию производительности архитектуры STM32F7 в различных конфигурациях, то есть выполнение кода и хранение данных в разных местах памяти с использованием ART accelerator™ и кешей.
- Stm32f7_performances_DMA призван продемонстрировать производительность архитектуры в конфигурации с несколькими ведущими.

Каждый проект выполняется для следующих доступных плат: STM32756G-EVAL, STM32F769I-EVAL и STM32F723E-DISCO.

1 Обзор системной архитектуры серии STM32F7

1.1 Ядро Cortex®-M7

Устройства серии STM32F7 построены на высокопроизводительном 32-битном ядре RISC ARM® Cortex®-M7, работающем на частоте до 216 МГц. Ядро Cortex®-M7 оснащено высокопроизводительным блоком операций с плавающей запятой (FPU). Ядро может содержать модуль с плавающей запятой одинарной точности или модуль с плавающей запятой двойной точности (в зависимости от устройства серии STM32F7), которые поддерживают все инструкции и типы данных ARM® одинарной и двойной точности. Он также реализует полный набор инструкций DSP и блок защиты памяти (MPU), повышающий безопасность приложения. Прямая совместимость Cortex®-M4 с Cortex®-M7 позволяет двоич-

ным файлам, скомпилированным для Cortex®-M4, работать непосредственно на Cortex®-M7.

Cortex®-M7 имеет 6/7 ступенчатый суперскалярный конвейер с предсказанием ветвлений и двойными инструкциями по выпуску. Функция прогнозирования ветвлений позволяет разрешению ветвей предвидеть следующую ветвь и, следовательно, уменьшить количество циклов, потребляемых циклами, с 4 и 3 циклов до 1 цикла на цикл. Функция двойной инструкции позволяет ядру выполнять две инструкции одновременно и, как правило, независимо от их порядка, что увеличивает пропускную способность инструкций.

1.2 Системные кэши Cortex®-M7

В устройствах встроен Cortex®-M7 с кешем уровня 1 (кэш L1), который разделен на два отдельных кеша: кэш данных (D-кэш) и кэш инструкций (I-кэш), что позволяет использовать архитектуру Гарварда, принося лучшую производительность. Эти кэши позволяют достичь состояния нулевого ожидания даже на высоких частотах.

По умолчанию кэши инструкций и данных отключены.

Библиотека ARM CMSIS предоставляет две функции, которые включают кеширование данных и инструкций:

- SCB_EnableICache () для включения кеширования инструкций
- SCB_EnableDCache () для включения и отключения кеша данных

Для получения дополнительной информации о том, как включить и сделать кэш недействительным, обратитесь к «Справочному руководству по архитектуре ARMv7-M».

См. Также кэш уровня 1 в примечании к применению серии STM32F7 (AN4839) для получения дополнительных сведений об использовании кэша L1 в серии STM32F7.

Таблица 1 суммирует размер кэша для каждого устройства в серии STM32F7.

Table 1. STM32F7 Series device cache sizes

Device	Instruction cache size	Data cache size
STM32F74xxx and STM32F75xxx	4 Kbytes	4 Kbytes
STM32F72xxx and STM32F73xxx	8 Kbytes	8 Kbytes
STM32F76xxx and STM32F77xxx	16 Kbytes	16 Kbytes

1.3 Интерфейсы шины Cortex®-M7

Cortex®-M7 имеет пять интерфейсов: AXIM, ITCM, DTCM, AHBS и AHBP. В этом разделе описывается каждый из них.

Все эти интерфейсы являются главными, за исключением интерфейса AHBS, который является подчиненным, позволяющим другому мастеру подключаться к Cortex®-M7.

1.3.1 Интерфейс шины AXI

AXI (advanced extensible interface master - расширенный расширяемый интерфейс). Cortex®-M7 реализует AXIM AMBA4, который представляет собой 64-битный интерфейс для большей пропускной способности выборки инструкций и загрузки данных.

Любой доступ, не предназначенный для интерфейса TCM или АНВР, обрабатывается соответствующим контроллером кэша, если кэш включен. Пользователь должен учитывать, что не все области памяти кэшируются, это зависит от их типов. Области памяти, имеющие типы: Shared Memory, Device или Strongly Ordered, не кэшируются. Кэшируется только обычный тип памяти без общего доступа.

Для получения дополнительной информации об общих правилах, касающихся атрибутов и поведения памяти, обратитесь к «Справочному руководству по архитектуре ARMv7-M».

Чтобы изменить тип и атрибут области памяти, можно использовать MPU, чтобы сделать его кэшируемым. Это делается путем настройки поля TEX и битов S, C и B в регистре MPU_RASR.

Таблица 2 суммирует атрибуты области памяти после сброса cortex®-M7.

Table 2. Cortex®-M7 default memory attributes after reset

Address range	Region name	Type	Attributes	Execute Never?
0x00000000-0x1FFFFFFF	Code	Normal	Cacheable, Write-Through, Allocate on read miss	No
0x20000000-0x3FFFFFFF	SRAM	Normal	Cacheable, Write-Back, Allocate on read and write miss	No
0x40000000-0x5FFFFFFF	Peripheral	Device	Non-shareable	Yes
0x60000000-0x7FFFFFFF	RAM	Normal	Cacheable, Write-Back, Allocate on read and write miss	No
0x80000000-0x9FFFFFFF	RAM	Normal	Cacheable, Write-Through, Allocate on read miss	No
0xA0000000-0xBFFFFFFF	External Device	Device	Shareable	Yes
0xC0000000-0xDFFFFFFF	External Device	Device	Non-shareable	Yes
0xE0000000-0xE00FFFFF	Private peripheral bus	Strongly ordered	-	Yes
0xE0010000-0xFFFFFFFF	Vendor system	Device	Non-shareable	Yes

Дополнительные сведения об использовании MPU см. В руководстве по программированию процессоров Cortex®-M7 серии STM32F7 (PM0253) в разделе «Модуль защиты памяти».

В устройствах серии STM32F7 64-разрядная шина AXI Master соединяет ядро с матрицей шины через высокопроизводительный мост AXI с несколькими АНВ-мостом, который имеет четыре основных интерфейса:

- 1x 64-битный АНВ во внутреннюю флэш-память
- 3x 32-битных АНВ на матрицу шины

1.3.2 Интерфейс шины TCM

TCM (Tightly Coupled Memory локальной жестко связанной памяти) предназначен для подключения ядра к внутренней памяти RAM. Интерфейс TCM имеет гарвардскую архитектуру, поэтому существуют интерфейсы ITCM (Instruction TCM) и DTCM (Data TCM). ITCM имеет один 64-битный интерфейс памяти, а DTCM разделен на два 32-битных порта: D0TCM и D1TCM.

1.3.3 Интерфейс шины АНBS

Cortex®-M7 АНBS (подчиненное устройство АНВ) - это 32-разрядный интерфейс, обеспечивающий доступ системы к ITCM, DITCM и D0TCM. Однако в архитектуре STM32F7 АНBS разрешает только передачу данных из / в DTCM-RAM (см. Рисунок 1). Шина ITCM недоступна в АНBS, поэтому передача данных DMA в / из ОЗУ ITCM не поддерживается. Для передачи DMA в / из флэш-памяти на интерфейсе ITCM все передачи принудительно проходят через шину АНВ. Интерфейс АНBS может использоваться, когда ядро находится в спящем состоянии, поэтому передачи DMA могут выполняться в режимах с низким энергопотреблением.

1.3.4 Интерфейс шины АНBP

Интерфейс АНBP (периферийное устройство АНВ) - это один 32-разрядный интерфейс, предназначенный для подключения ЦП к периферийным устройствам. Используется только для доступа к данным. Выборка инструкций никогда не выполняется на этом интерфейсе. В архитектуре STM32F7 эта шина соединяет периферийную шину АНBP ядра Cortex-M7 с матрицей шины АНВ. Целями этой шины являются периферийные устройства АНВ1, АНВ2, APB1 и APB2.

1.4 Матрица шины STM32F7

Устройства серии STM32F7 имеют матрицу шины 216 МГц, которая соединяет ядро, ведущие и ведомые устройства. Он позволяет использовать несколько параллельных путей доступа между основными шинами, главными шинами и подчиненными шинами, обеспечивая одновременный доступ и эффективную работу даже при одновременной работе нескольких высокоскоростных периферийных устройств. ЦП и его матрица шины могут работать на одной и той же частоте, то есть 216 МГц.

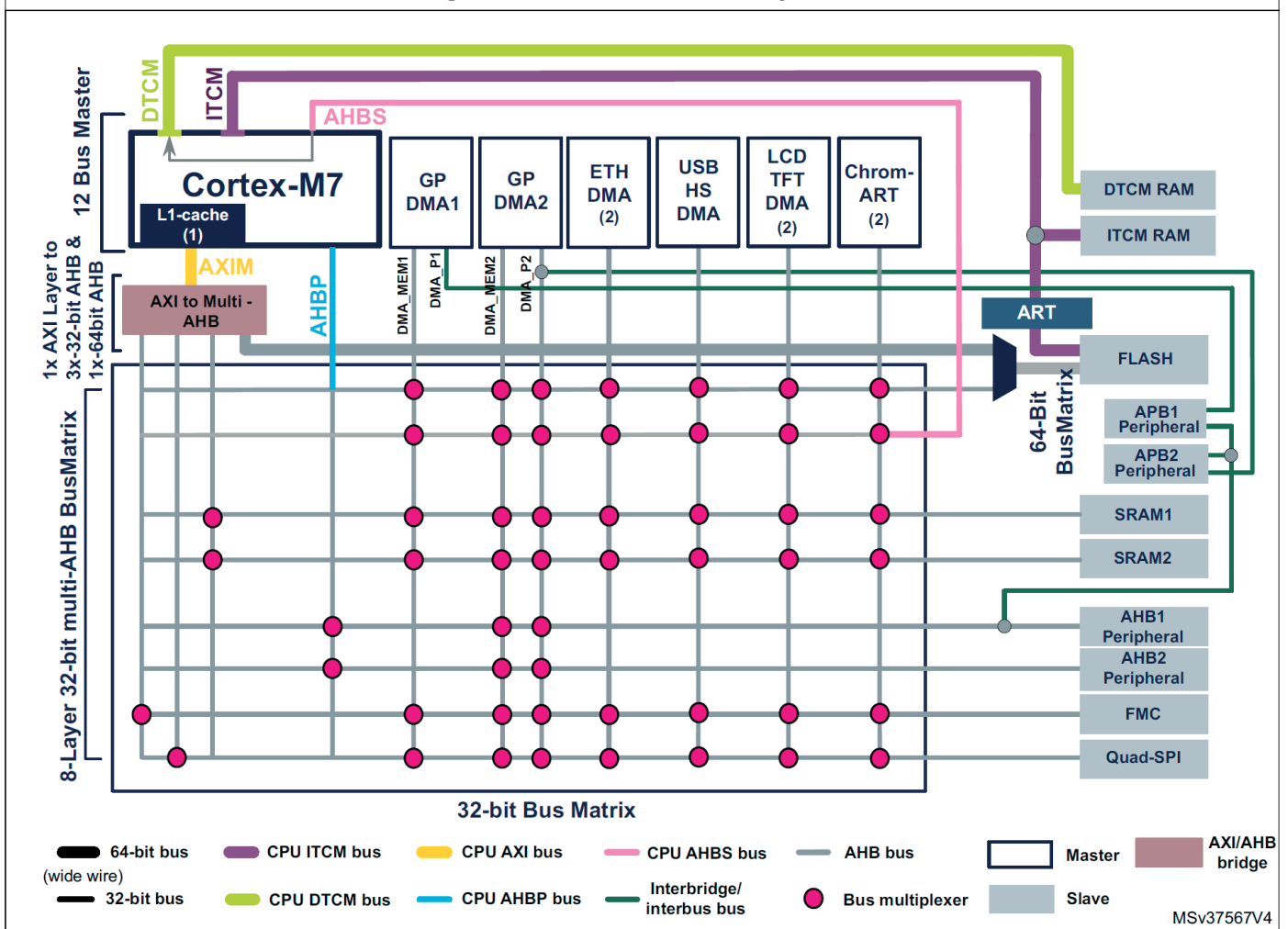
Внутренний арбитр разрешает конфликты и параллелизм мастеров шины в матрице шины. Он использует циклический алгоритм.

На рисунке 1 показана общая системная архитектура устройств серии STM32F7, а также соединения матрицы шины.

Матричные соединения шины STM32F7:

- Двенадцать мастеров шины или инициаторов:
 - Три 32-битных шины АНВ, идущие от моста AXI к мосту АНВ.
 - Одна 64-битная шина АНВ, подключенная к встроенной флэш-памяти, которая исходит от моста AXI к АНВ.
 - Cortex®-M7 Шина периферийных устройств АНВ
 - Шина памяти DMA1 – Шина памяти DMA2
 - Периферийная шина DMA2
 - Ethernet DMA-шина (a)
 - USB OTG HS DMA шина
 - LCD-TFT контроллер DMA-bus (a)
 - Шина памяти Chrom-Art Accelerator® (DMA2D) (a)
- И восемь ведомых шин:
 - Встроенная флэш-память на шине АНВ (для чтения / записи Flash, для выполнения кода и доступа к данным)

Figure 1. STM32F7 Series system architecture



1. Размер кэша I / D:

- Для устройств STM32F74xxx и STM32F75xxx: 4 Кбайта.
- Для устройств STM32F72xxx и STM32F73xxx: 8 Кбайт.
- Для устройств STM32F76xxx и STM32F77xxx: 16 Кбайт.

2. Мастера недоступны в устройствах STM32F72xxx и STM32F73xxx.

- Cortex®-M7 подчиненный интерфейс AHBS для передачи данных прямого доступа к памяти только на DTCM-RAM
- Главная внутренняя SRAM1
- Внутренняя вспомогательная SRAM2
- Периферийные устройства AHB1, включая мосты AHB-APB, периферийные устройства APB1 и APB2
- AHB2 периферийные устройства
- Интерфейс памяти FMC
- Quad-SPI интерфейс памяти

1.5 Виды памяти STM32F7

Устройства STM32F7 встраивают флэш-память разного размера в зависимости от STM32F72xxx / STM32F73xxx, STM32F74xxx / STM32F75xxx или устройств STM32F76xxx / STM32F77xxx (см. Таб 3, Таб 4 и Таб 5), SRAM с разными размерами, разбросанные архитектура и интерфейсы внешней памяти, такие как FMC и Quad-SPI. Эта конфигурация дает пользователю гибкость для разделения ресурсов памяти приложения в соответствии с его потребностями и для получения

оптимального компромисса производительности по сравнению с размером кода приложения.

1.5.1 Встроенная флэш-память

Каждое устройство серии STM32F7 имеет свой собственный размер флэш-памяти (см. Таблицу 3, Таблицу 4 и Таблицу 5). В устройствах STM32F72xxx и STM32F73xxx флэш-память доступна при чтении данных шириной 128 бит, в то время как в устройствах STM32F74xxx и STM32F75xxx флэш-память доступна при чтении данных шириной 256 бит. В устройствах STM32F76xxx и STM32F77xxx флэш-память доступна для чтения данных шириной 128 или 256 бит после включения режима банка (256-битный доступ в режиме одного банка и 128-битный доступ в режиме двойного банка).

Во всех устройствах флэш-память доступна через три основных интерфейса для чтения и / или записи.

- 64-битный интерфейс ITCM: он соединяет встроенную флэш-память с Cortex-M7 через шину ITCM (путь 1 на рисунке 2) и используется для выполнения программы и доступа для чтения данных для констант. Доступ для записи во флэш-память через эту шину не разрешен. Флэш-память доступна ЦП через ITCM, начиная с адреса 0x00200000.

Поскольку встроенная флэш-память работает медленнее по сравнению с ядром, предоставляется адаптивный ускоритель реального времени (ART), чтобы раскрыть производительность ядра Cortex-M7 и обеспечить выполнение из флэш-памяти с нулевым ожиданием на частоте ЦП до 216 МГц. STM32F7 ART доступен только для доступа к флэш-памяти через интерфейс ITCM. Он реализует унифицированный кэш инструкций и кэш ветвлений 128 бит x 64 строки в устройствах STM32F72xxx и STM32F73xxx, 256 бит x 64 строки в устройствах STM32F74xxx и STM32F75xxx и 128/256 бит x 64 строки в устройствах STM32F76xxx и STM32F77xxx, следующих за выбором режим банка. ART доступен как для инструкций, так и для доступа к данным, что увеличивает скорость выполнения последовательного кода и циклов. ART также реализует Prefetcher (ART-Prefetch).

- 64-битный интерфейс АНВ:

Он подключает встроенную флэш-память к Cortex-M7 через мост AXI / АНВ (путь 2 на рисунке 2). Он используется для выполнения кода, доступа для чтения и записи. Флэш-память доступна ЦП через AXI, начиная с адреса 0x08000000.

- 32-битный интерфейс АНВ:

Он используется для передачи DMA из флэш-памяти (путь 3 на рисунке 2). Доступ к флэш-памяти DMA осуществляется, начиная с адреса 0x0800 0000.

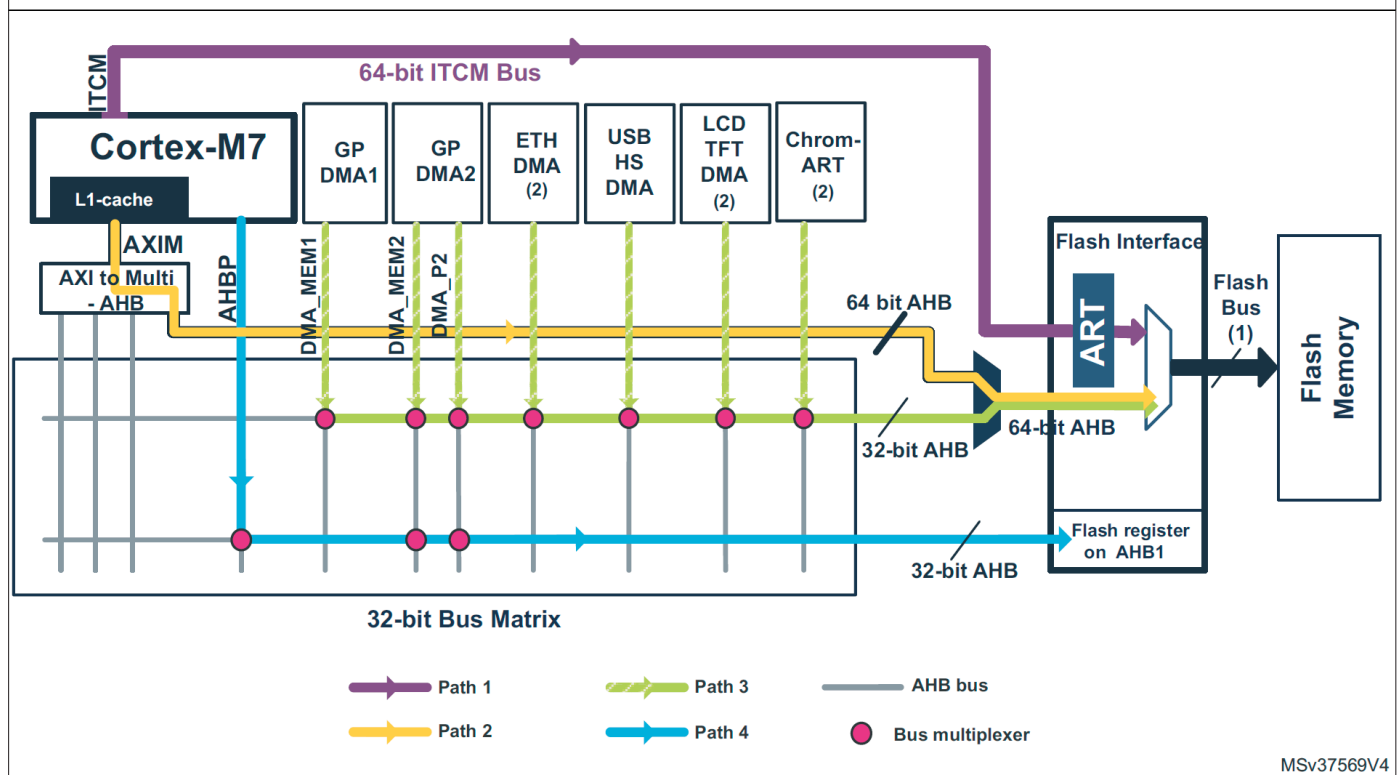
Для доступа к регистрам управления, конфигурации и состояния интерфейс флэш-памяти доступен через периферийный путь АНВР / АНВ1, который представляет собой 32-битную шину АНВ (путь 4 на рисунке 2).

Если доступ к Flash-памяти осуществляется начиная с адреса 0x0800 0000, он выполняется автоматически через AXI / АНВ. Кэши инструкций и / или данных должны быть включены в этой конфигурации, чтобы получить доступ к флэш-памяти, подобный состоянию 0-ожидания.

Если доступ к Flash-памяти осуществляется начиная с адреса 0x0200 0000, он выполняется автоматически через шину ITCM. Ускоритель ART должен быть

включен для получения доступа к флеш-памяти в состоянии, эквивалентном 0-ожиданию, через шину ITCM. ART активируется установкой бита 9 в регистре FLASH_ACR, в то время как ART-Prefetch включается установкой бита 8 в том же регистре.

Figure 2. Flash memory interfaces (paths: 1, 2, 3, 4)



MSv37569V4

1. Ширина строки флэш-памяти:

- Для устройств STM32F74xxx и STM32F75xxx: 256 бит.
- Для устройств STM32F72xxx и STM32F73xxx: 128 бит.
- Для устройств STM32F76xxx и STM32F77xxx: 256 бит в одиночном банке и 128 бит в режиме двойного банка.

2. Мастера недоступны в устройствах STM32F72xxx и STM32F73xxx.

3. Заштрихованный путь означает несколько возможных путей.

1.5.2 Встроенная SRAM

Устройства серии STM32F7 имеют большую SRAM с разнесенной архитектурой. Она разделена на четыре блока:

- ОЗУ команд (ITCM-RAM) отображается по адресу 0x0000 0000 и доступно только ядру, то есть по пути 1 на рисунке 3. Память доступна в байтах, полусловах (16 бит), словах (32 бита) или двойных словах (64 бита). Доступ к ITCM-RAM возможен на максимальной тактовой частоте ЦП без задержки. ITCM-RAM защищена от конфликтов шины, поскольку только ЦП может получить доступ к этой области RAM.

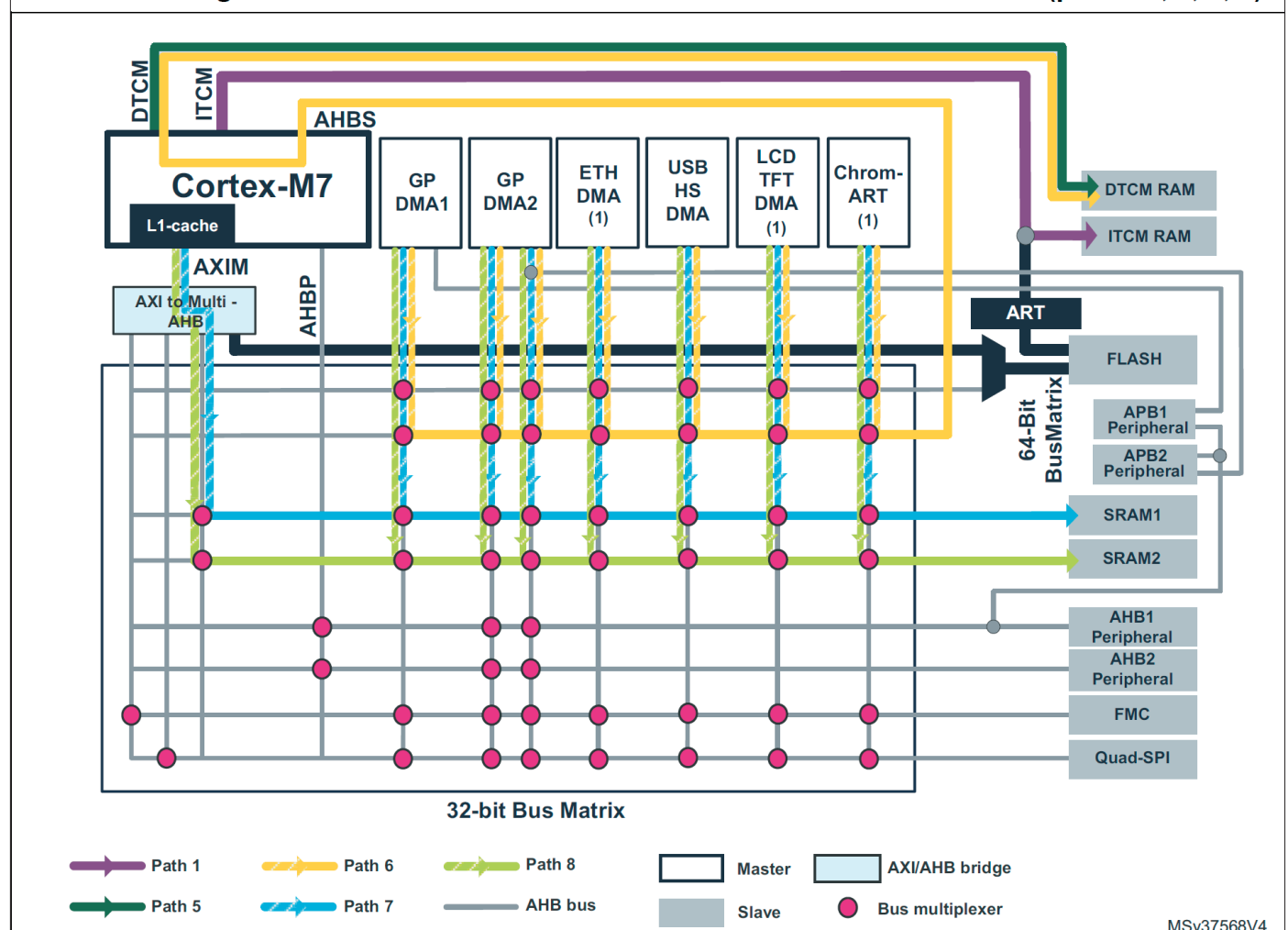
- DTCM-RAM отображается на интерфейсе TCM по адресу 0x2000 0000 и доступна для всех мастеров АНВ из матрицы шины АНВ: для ЦП через шину DTCM (путь 5 на рис. 3) и для DMA через специальную шину АНБС ядро, которое является путем 6 на рисунке 3. Память доступна в байтах, полусловах (16 бит), словах (32 бита) или двойных словах (64 бита). DTCM-RAM доступна при максимальной тактовой частоте ЦП без задержки. Параллельный доступ к DTCM-RAM со стороны мастеров (DMA) и их приоритеты могут обрабатываться ведомым регистром

управления Cortex-M7 (регистр CM7_AHBSCR). ЦП может получить более высокий приоритет для доступа к DTCM-RAM по сравнению с другими мастерами (DMA). Дополнительные сведения об этом регистре см. в техническом справочном руководстве по процессору ARM® Cortex®-M7.

- SRAM1 доступна всем мастерам АHB из матрицы шины АHB, то есть всем DMA общего назначения, а также выделенным DMA. Доступ к SRAM1 можно получить с помощью байтов, полуслов (16 бит) или слов (32 бита). Обратитесь к Рисунку 3 (путь 7) для получения информации о возможном доступе к SRAM1. Ее можно использовать для загрузки / сохранения данных, а также для выполнения кода.

- SRAM2 доступна для всех мастеров АHB из матрицы шины АHB. Все DMA общего назначения, а также выделенные DMA могут получить доступ к этой области памяти. Доступ к SRAM2 можно получить с помощью байтов, полуслов (16 бит) или слов (32 бита). Обратитесь к Рисунку 3 (путь 8) для получения информации о возможном доступе к SRAM2. Ее можно использовать для загрузки / сохранения данных, а также для выполнения кода.

Figure 3. Different accesses of DMAs to some internal memories (paths: 5, 6, 7, 8)



1. Мастера недоступны в устройствах STM32F72xxx и STM32F73xxx.
2. Заштрихованные пути означают несколько возможных путей.

[Table 3](#), [Table 4](#) and [Table 5](#) summarize the internal memory mapping and the memory sizes of the STM32F7 Series devices:

Table 3. internal memory summary of the STM32F74xxx/STM32F75xxx devices

Memory type	Memory region	Address start	Address end	Size	Access interfaces
FLASH	FLASH-ITCM	0x0020 0000	0x002F FFFF	1 Mbyte	ITCM (64-bit)
	FLASH-AXIM	0x0800 0000	0x080F FFFF		AHB (64-bit) AHB (32-bit)
RAM	DTCM-RAM	0x2000 0000	0x2000 FFFF	64 Kbytes	DTCM (64-bit)
	ITCM-RAM	0x0000 0000	0x0000 3FFF	16 Kbytes	ITCM (64-bit)
	SRAM1	0x2001 0000	0x2004 BFFF	240 Kbytes	AHB (32-bit)
	SRAM2	0x2004 C000	0x2004 FFFF	16 KBytes	AHB (32-bit)

Table 4. internal memory summary of the STM32F72xxx/STM32F73xxx devices

Memory type	Memory region	Address start	Address end	Size	Access interfaces
FLASH	FLASH-ITCM	0x0020 0000	0x0027 FFFF	512 Kbytes	ITCM (64-bit)
	FLASH-AXIM	0x0800 0000	0x0807 FFFF		AHB (64-bit) AHB (32-bit)
RAM	DTCM-RAM	0x2000 0000	0x2000 FFFF	64 Kbytes	DTCM (64-bit)
	ITCM-RAM	0x0000 0000	0x0000 3FFF	16 Kbytes	ITCM (64-bit)
	SRAM1	0x2001 0000	0x2003 BFFF	176 Kbytes	AHB (32-bit)
	SRAM2	0x2003 C000	0x2003 FFFF	16 Kbytes	AHB (32-bit)

Table 5. internal memory summary of the STM32F76xxx/STM32F77xxx devices

Memory type	Memory region	Address start	Address end	Size	Access interfaces
FLASH	FLASH-ITCM	0x0020 0000	0x003F FFFF	2 Mbytes ⁽¹⁾	ITCM (64-bit)
	FLASH-AXIM	0x0800 0000	0x081F FFFF		AHB (64-bit) AHB (32-bit)
RAM	DTCM-RAM	0x2000 0000	0x2001 FFFF	128 Kbytes	DTCM (64-bit)
	ITCM-RAM	0x0000 0000	0x0000 3FFF	16 Kbytes	ITCM (64-bit)
	SRAM1	0x2002 0000	0x2007 BFFF	368 Kbytes	AHB (32-bit)
	SRAM2	0x2007 C000	0x2007 FFFF	16 KBytes	AHB (32-bit)

1. 2 Mbytes in single bank and 2 x 1 Mbyte in dual bank.

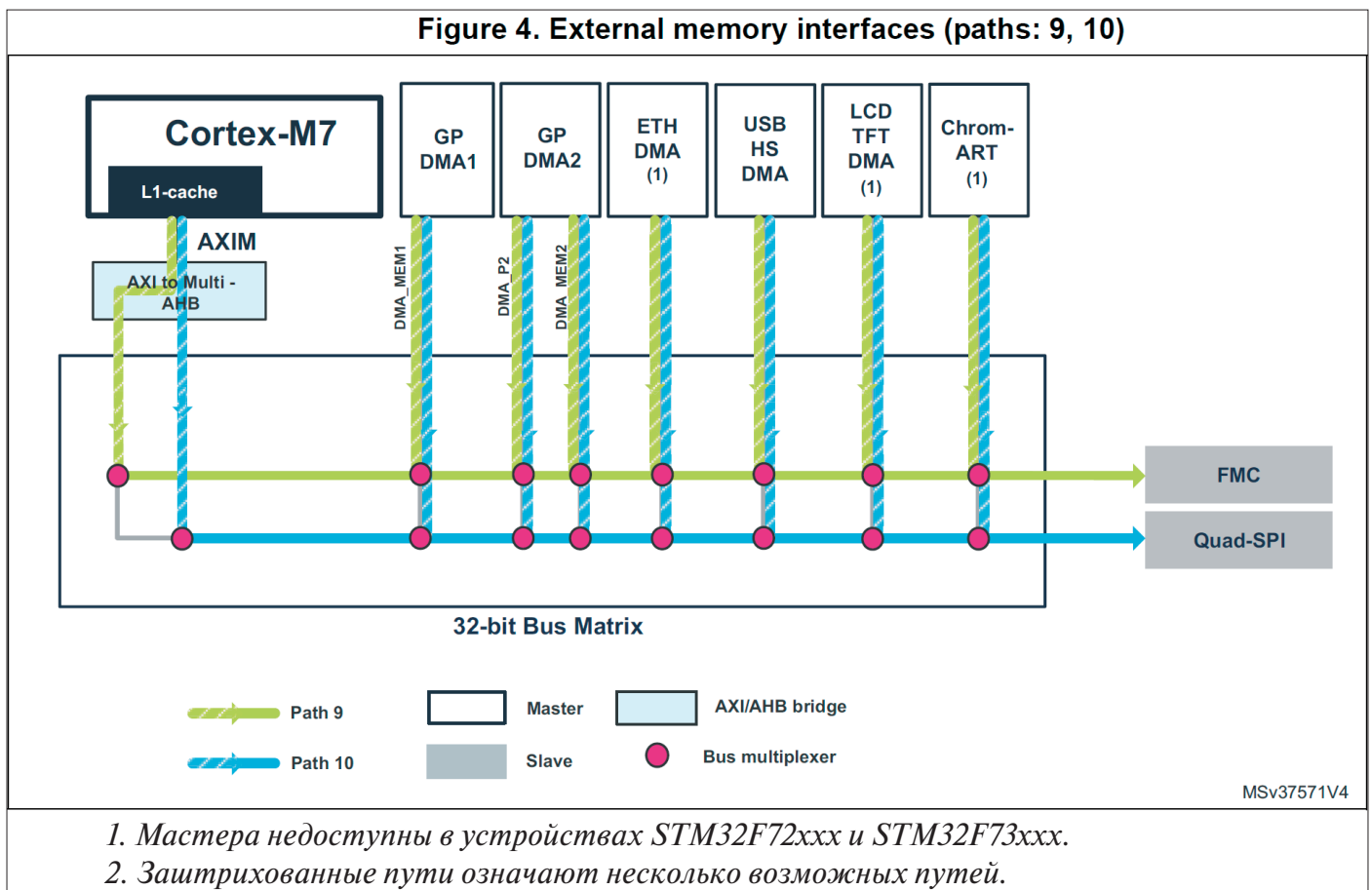
1.5.3 Внешняя память

В дополнение к внутренней памяти и контроллерам хранения, таким как USB и SDMMC, пользователь может расширить память STM32F7 с помощью гибкого контроллера памяти (FMC) и контроллера Quad-SPI.

На рисунке 4 показаны возможные пути, которые соединяют ЦП и различные DMA с этими внешними запоминающими устройствами через шины AXI / AHB. Как показано на рисунке 4, внешние запоминающие устройства могут использовать кэш Cortex®-M7, поэтому они могут получить максимальную производительность, будь то загрузка / сохранение данных или место выполнения кода. Это позволяет сочетать производительность и большой объем памяти.

Путь 9 на рисунке 4 показывает возможные доступы к FMC с использованием ЦП или DMA.

Путь 10 на рисунке 4 показывает возможные доступы Quad-SPI с использованием ЦП или DMA.

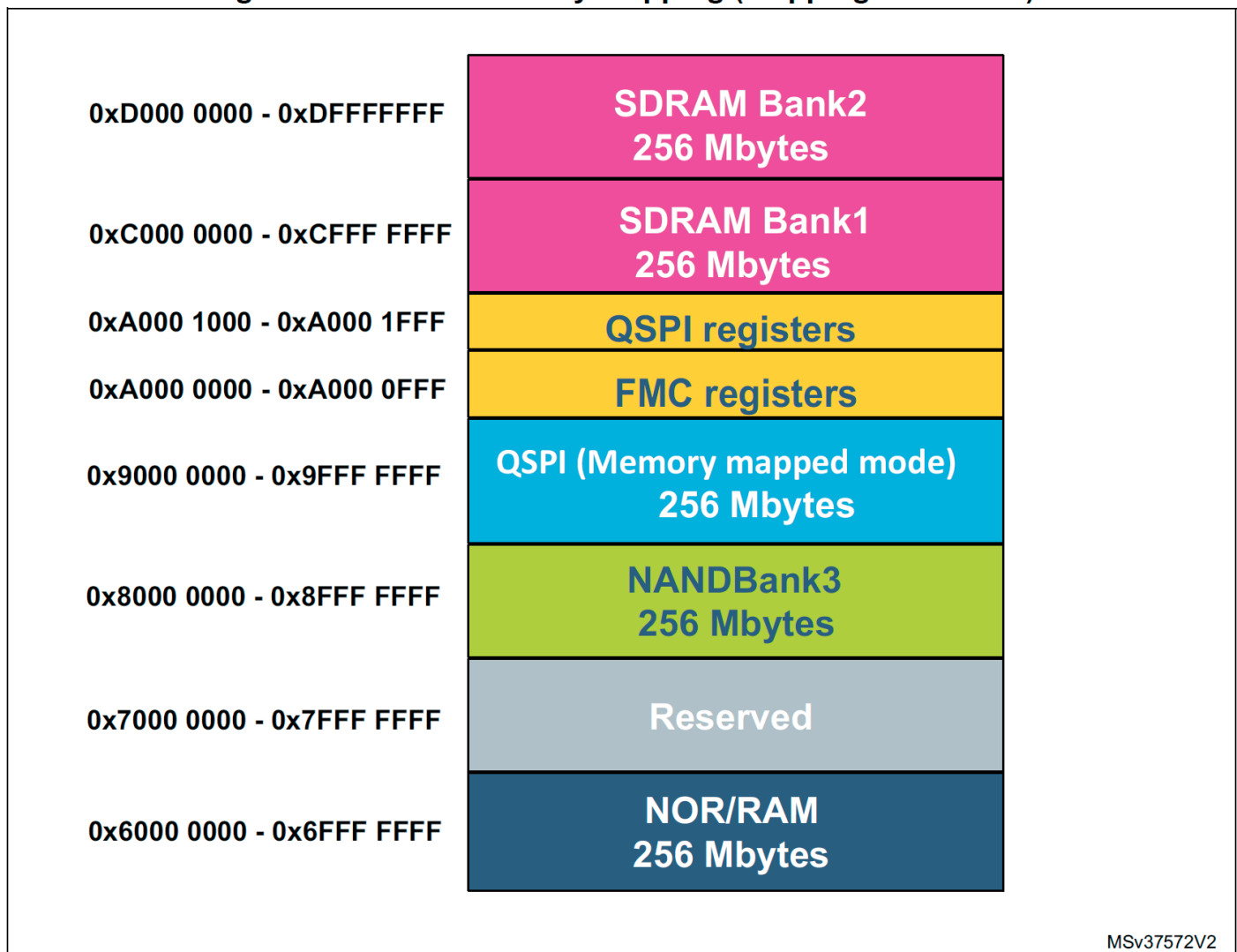


Все внешние устройства памяти доступны для всех мастеров (ЦП и DMA). Таким образом, разрешены DMAх-передачи из памяти в память или DMAх-передачи с периферийных устройств / памяти.

На рисунке 5 показано отображение внешней памяти и диапазоны их адресов после сброса (биты SWP_FMC [1: 0] установлены в 0 в регистре SYSCFG_MEMRMP).

Интерфейс гибкого контроллера памяти (FMC)

Контроллер STM32F7 FMC взаимодействует с устройствами с отображением памяти, включая SRAM, ROM, флэш-память NOR / NAND и устройства SDRAM.

Figure 5. External memory mapping (mapping after reset)

Он используется либо для выполнения программы (кроме NAND Flash), либо для операций загрузки / сохранения данных.

Характеристики STM32F7 FMC:

- 4 банка для одновременной поддержки разной памяти
- Независимое управление выбором микросхемы для каждого банка памяти
- Независимая конфигурация для каждого банка памяти
- Программируемые тайминги для поддержки широкого спектра устройств.
- 8/16/32-битная шина данных
- Внешнее асинхронное управление ожиданием
- Интерфейсы с синхронной DRAM (SDRAM), которая предоставляет два банка SDRAM

Все внешние запоминающие устройства FMC разделяют адреса, данные и управляющие сигналы с контроллером.

Доступ к каждому внешнему устройству осуществляется с помощью уникального выбора микросхемы. FMC выполняет только один доступ к внешнему устройству за раз.

Две стандартные области банков SDRAM не кэшируются. Таким образом, даже если кеш включен, данные или инструкции не будут проходить через кеш. Чтобы извлечь выгоду из ускорения кэширования, банки SDRAM можно переназначить с 0xC000 0000 и 0xD000 0000 на 0x6000 0000 и 0x7000 0000 соответственно, которые

по умолчанию являются кэшируемыми областями. Это делается путем установки поля SWP_FMC [1: 0] = 01 в регистре SYSCFG_MEMRMP. Если переназначение не подходит для приложения, можно использовать MPU Cortex®-M7 для изменения свойств области памяти SDRAM по умолчанию, чтобы она была кэшируемой.

Все внешние запоминающие устройства, которые будут подключены к FMC, получают выгоду от данных и кэша L1 (путь 9 на рисунке 4), что позволит иметь больше данных и / или размеров кода с максимальной производительностью.

Интерфейс Quad-SPI

Устройства серии STM32F7 включают интерфейс памяти Quad-SPI, который представляет собой специализированный интерфейс связи, предназначенный для флэш-памяти с одним, двумя или четырьмя SPI. Этот интерфейс с множественной шириной поддерживает традиционный одноканальный последовательный ввод и вывод SPI, а также двух- и четырехразрядные последовательные команды. Кроме того, интерфейс поддерживает команды чтения с удвоенной скоростью передачи данных (DDR), что означает, что передача адреса и чтение данных выполняются по обоим фронтам тактового сигнала связи. Это позволяет в два раза увеличить пропускную способность данных / инструкций и, следовательно, повысить эффективность доступа к внешней флэш-памяти Quad-SPI.

Он может работать в одном из трех следующих режимов:

- Прямой режим: все операции выполняются через регистры Quad-SPI.
- Режим опроса состояния: регистр состояния внешней флэш-памяти периодически считывается, и в случае установки флага может быть сгенерировано прерывание.
- Режим отображения памяти: внешняя флэш-память отображается в памяти и воспринимается системой как внутренняя память.

Интерфейс STM32F7 Quad-SPI может управлять флэш-памятью объемом до 256 Мбайт, начиная с 0x9000 0000 до 0x9FFF FFFF в режиме отображения памяти. Он отображается в исполняемой области, поэтому повторное сопоставление не требуется.

По сравнению с FMC, Quad-SPI позволяет подключать внешнюю флэш-память с меньшими затратами, небольшими корпусами (уменьшение площади печатной платы) и меньшим использованием GPIO. 6 GPIO используются в режиме с одним SPI (4 бита) для любого размера флэш-памяти или 10 GPIO в режиме с двумя SPI (8 бит).

Как показано на Рисунке 4 (путь 10), Quad-SPI отображается на выделенном уровне на АНВ и может извлекать выгоду из кэша L1. Это позволяет выполнять код и загружать данные из Quad-SPI с хорошей производительностью.

Quad-SPI также доступен всем мастерам на матрице шины АНВ, особенно ускорителю Chrom-ART® и LCD-TFT, что обеспечивает эффективную передачу данных, особенно изображений, для графических приложений, где требуется высокая частота кадров при отображении. нужный.

Обратитесь к интерфейсу Quad-SPI (QSPI) в примечании к применению микроконтроллеров STM32 (AN4760), чтобы узнать, как конфигурировать, программировать и читать внешнюю память Quad-SPI с микроконтроллерами STM32.

1.6 DMAs

Устройства серии STM32F7 включают в себя два общих прямого доступа к памяти (GP DMAx), которые могут обеспечить высокоскоростную передачу данных между памятью и между памятью и периферийным устройством. DMA предназначены для разгрузки ЦП и обеспечения некоторых передач, когда ядро находится в режиме низкого энергопотребления. Каждый DMA имеет 8 потоков и 8 каналов на поток.

Устройства серии STM32F7 встраивают также другие выделенные DMA для Ethernet, USB OTG, LCD-TFT и Chrom-ART accelerator®, за исключением устройств STM32F72xxx и STM32F73xxx, которые встраивают только один выделенный DMA, который является USB OTG. Все DMA имеют доступ к следующей внутренней памяти: встроенная флэш-память, SRAM1, SRAM2 и DTCM через шину AHBS Cortex®-M7. Все DMA также имеют доступ к внешней памяти через контроллеры FMC и Quad-SPI через матрицу шины.

Шина ITCM недоступна на AHBS. Таким образом, передача данных DMA в / из ОЗУ ITCM не поддерживается. Для передачи DMA в / из флэш-памяти на интерфейсе ITCM все передачи принудительно проходят через шину АНВ, то есть все адреса передачи автоматически преобразуются из адресов ITCM в адреса AXI.

Возможные передачи GP DMA перечислены ниже:

- GP DMA1 передает:
 - DMA1 не может обращаться к периферийным устройствам АНВ1 / АНВ2
 - DMA1 может только передавать APB1 из / в память
- GP DMA2 передает:
 - DMA2 может делать все возможные переводы

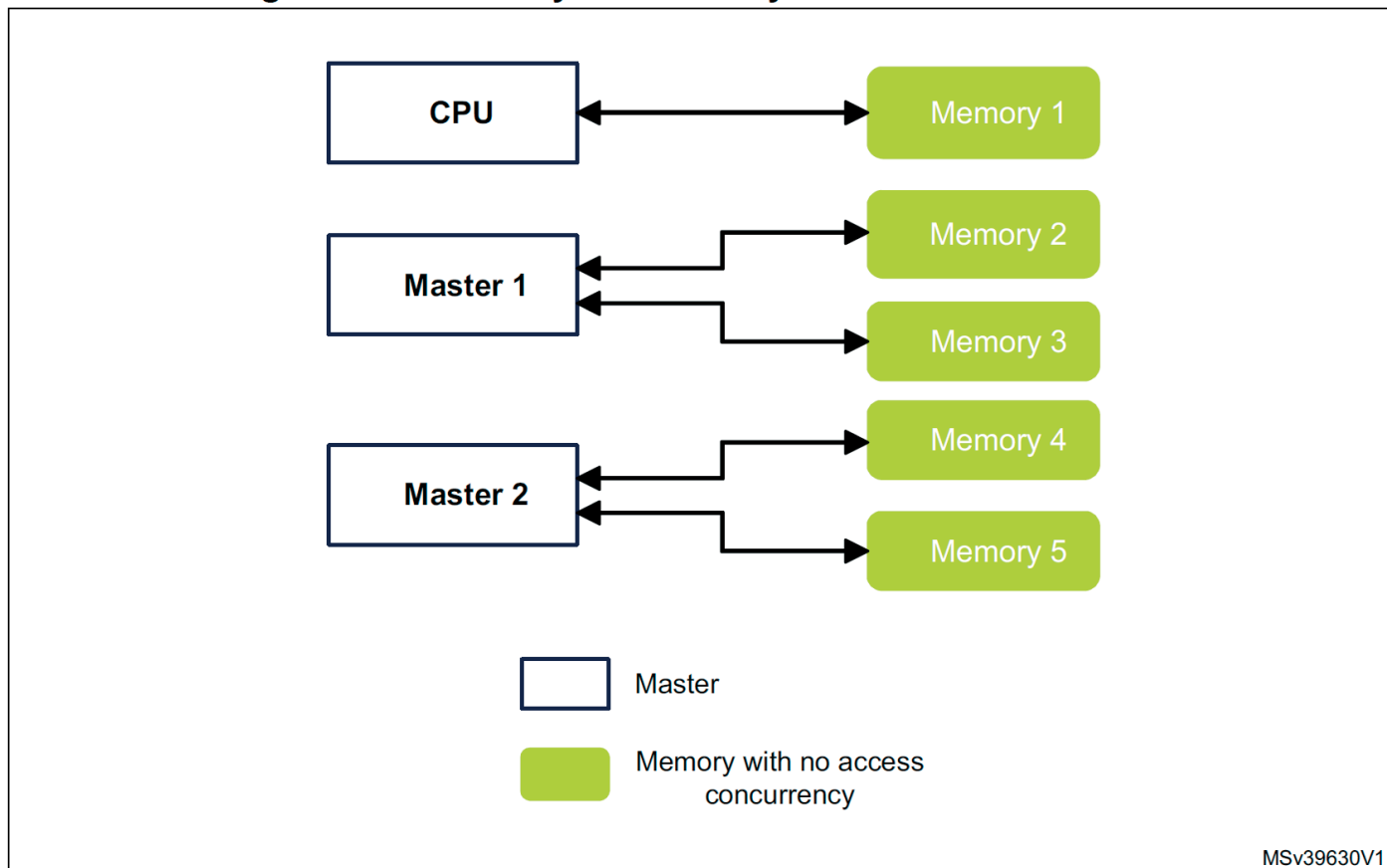
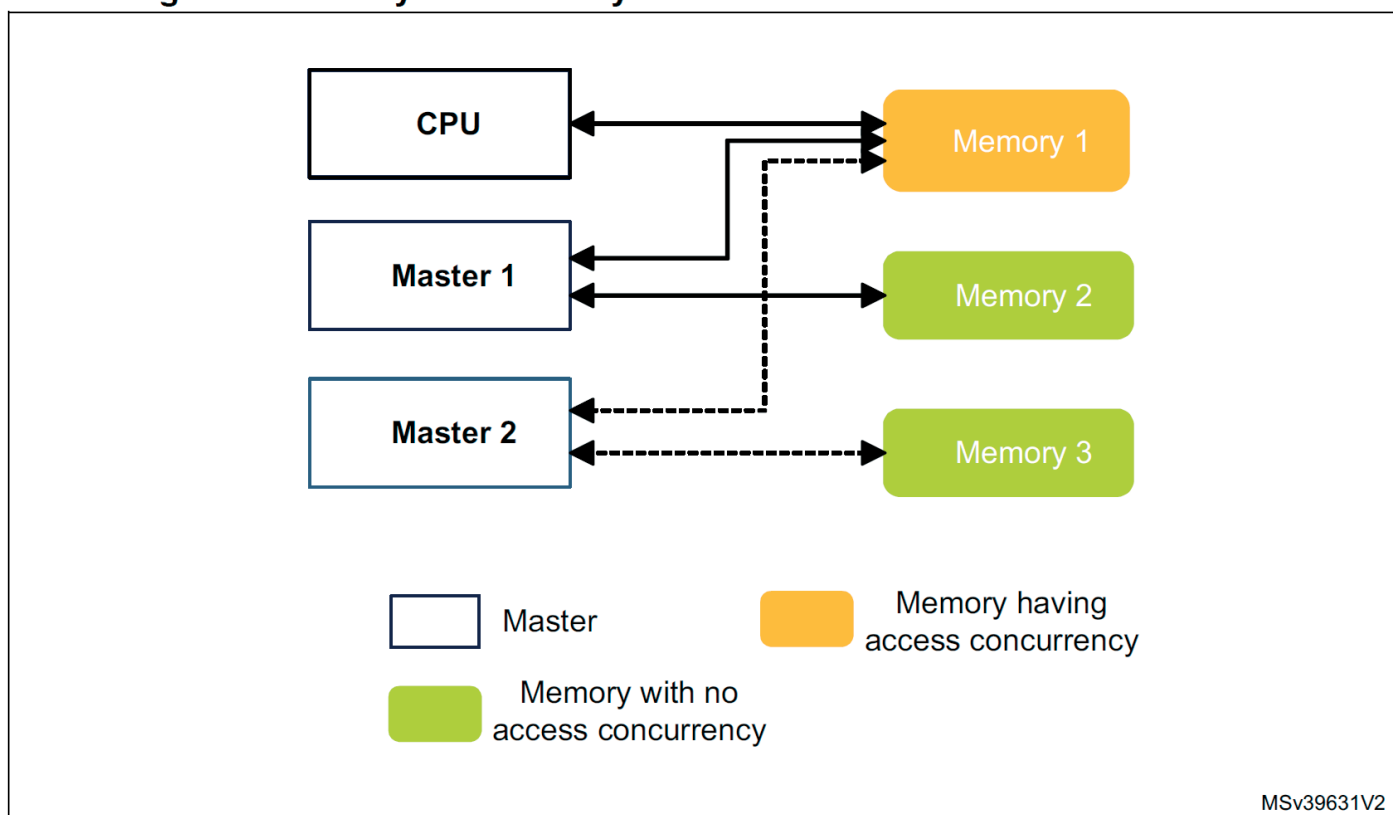
Благодаря мосту AXI-to-multi-АНВ и матричной архитектуре шины одновременный доступ ЦП и DMA к различным ведомым устройствам эффективно управляется, чтобы минимизировать задержку. Конфликт также управляется в Cortex-M7 на уровне TCU между LSU (блоком загрузки / хранения) и доступами к ведомой шине АНВ, когда ЦП и другое системное ведущее устройство пытаются одновременно получить доступ к DTCM-RAM. Только один Мастер может получить доступ к AHBS, который управляется арбитражем матрицы шины.

Доступ ЦП и DMA к одной и той же памяти может повлиять на производительность системы.

Если ЦП и DMA не обращаются к одной и той же памяти одновременно, конкуренция отсутствует. Следовательно, производительность остается такой же, как если бы с ЦП не было другого ведущего устройства (см. Рисунок 6).

Если существует параллелизм доступа к памяти между ЦП и одним или несколькими ведущими устройствами (см. Рисунок 7), производительность снижается в зависимости от скорости доступа к памяти, к которой обращаются DMA (доступ к внутренней памяти относительно быстрый по сравнению с внешней памятью) и если параллелизм выполняется при чтении или записи.

Задержка, генерируемая одновременным доступом к главному серверу, зависит от нескольких факторов, некоторые из которых рассматриваются в Разделе 3.2: Производительность доступа к памяти ЦП при использовании DMA.

Figure 6. No memory concurrency between DMA and CPU**Figure 7. Memory concurrency between one or more masters and CPU**

В разделе 2.3 представлена демонстрация производительности системы при включении нескольких мастеров (DMA) в различных сценариях.

В разделе 3.2 представлены различные результаты и анализ, полученные в типичных сценариях на основе рисунков 6 и 7.

1.7 Основные различия между устройствами серии STM32F7 с точки зрения архитектуры

Таблица 6 суммирует только различия между устройствами серии STM32F7, связанные с архитектурой (периферийные различия не включены).

Table 6. Main differences between the STM32F7 Series devices

	STM32F74xxx/ STM32F75xxx	STM32F72xxx/ STM32F73xxx	STM32F76xxx/ STM32F77xxx
Cortex-M7 revision	r0P1	r1P0	r1P0
Instruction and Data cache sizes	4 Kbytes	8 Kbytes	16 Kbytes
FPU	Single precision floating point	Single precision floating point	Single and double precision floating point
Flash size	1 Mbyte	512 Kbytes	2 Mbytes
Bank mode/Flash width	Single bank with 256-bit access	Single bank with 128-bit access	– In single bank: 256-bit access – in dual bank: 128-bit access
DTCM-RAM	Size: 64 Kbytes @ 0x20000000	Size: 64 Kbytes @ 0x20000000	Size: 128 Kbytes @ 0x20000000
SRAM1	Size: 240 Kbytes @ 0x20010000	Size: 176 Kbytes @ 0x20010000	Size: 368 Kbytes @ 0x20020000
SRAM2	Size: 16 Kbytes @ 0x2004C000	Size: 16 Kbytes @ 0x2003C000	Size: 16 Kbytes @ 0x2007C000
ETH-DMA, LCD-TFT-DMA and Chrom-ART	Available	Not available	Available

2 Типичное применение

В этой заметке по применению представлены два примера программного обеспечения, которые демонстрируют, во-первых, производительность STM32F7 при доступе к памяти ЦП либо для хранения данных, либо для выполнения кода, а также для внутренней или внешней памяти. Во-вторых, примеры показывают влияние использования DMA, когда ЦП загружает / сохраняет данные в памяти в типичных сценариях.

Используемый пример - это пример БПФ, представленный в библиотеке CMSIS. Проект `stm32f7_performances` может использоваться как каркас, на котором пользователь может интегрировать свое приложение. Тот же пример используется для проекта `stm32f7_performances_DMA`, за исключением вычисления величины, которая была удалена, чтобы выделить больше ОЗУ для передачи DMA.

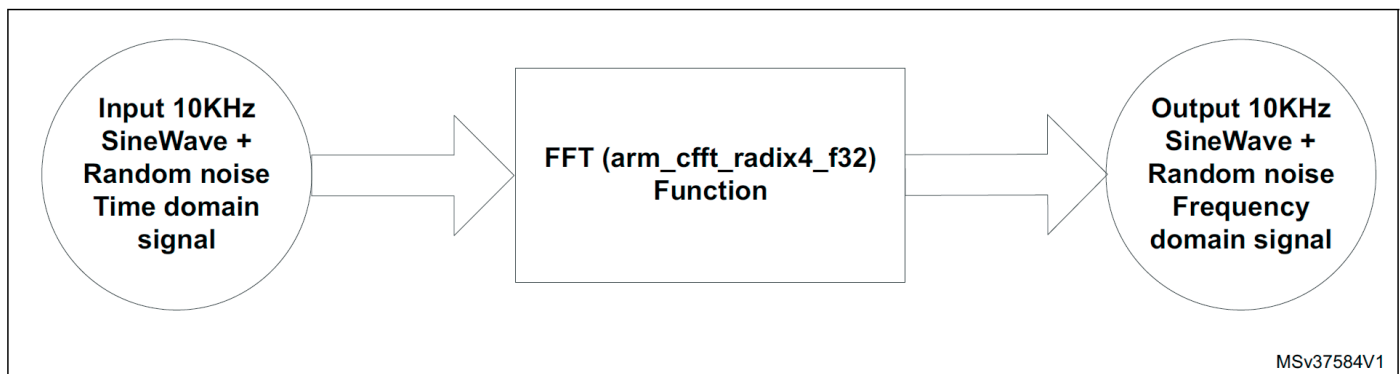
2.1 Демонстрация БПФ

Пример БПФ используется, поскольку он использует модуль с плавающей запятой Cortex-M7, содержит несколько циклов и операций загрузки / сохранения данных и может выполняться по разным путям / памяти. Код может быть выполнен из внутренней или внешней памяти. Пример состоит в вычислении максимальной энергии в частотной области входного сигнала с использованием комплексного БПФ, комплексной величины и функций максимума. Он использует 1024 точки БПФ, и вычисления основаны на числах с плавающей запятой одинарной точности. Входной сигнал представляет собой синусоидальную волну 10 кГц, смешанную с белым шумом. На рисунке 8 показана блок-схема преобразования.

Количество циклов, потребляемых процессом БПФ, также рассчитывается на основе системного таймера. Пример был запущен на платах STM32F732E-DISCO, STM32756G-EVAL и STM32769I-EVAL. Результаты отображаются на Hyperterminal через UART или в программе просмотра IDE `printf` для всех плат, а также на LCD-TFT только для плат STM32F7xxxx-EVAL.

Демонстрация БПФ показывает текущую конфигурацию проекта, частоту системы, различные конфигурации кешей, ART, ART-Prefetch (ON / OFF) и конфигурацию памяти в случае внешней памяти (SDRAM или Quad-SPI).

Figure 8. FFT example block diagram



2.2 Конфигурация проекта демонстрации доступа к памяти ЦП

Демонстрация доступа к памяти ЦП предоставляется с инструментами Keil MDK-ARM, IAR Embedded Workbench и System Workbench для STM32. Восемь / девять конфигураций этого проекта позволяют выбор данных и расположение кода.

Конфигурации названы в соответствии со следующими правилами:

N-ExecutionRegionPath_rwDataRegion где:

N: номер конфигурации.

ExecutionRegionPath: расположение в памяти и путь выполнения кода. Пользователь должен различать область выполнения и область загрузки. Область выполнения - это область памяти, в которой выполняется приложение. Область загрузки - это область памяти, в которую приложение было изначально загружено загрузчиком Flash и скопировано позже (подпрограммой, автоматически сгенерированной некоторыми инструментальными цепочками), в области выполнения, если два адреса различаются.

rwDataRegion: расположение в памяти инициализированных данных RW / Zero, стека и кучи.

Предлагаются следующие конфигурации:

- **1-FlashITCM_rwRAM-DTCM:** программа выполняется из Flash-ITCM в 7 состояниях ожидания с включенными ART и ART-предварительной выборкой, и данные загружаются / сохраняются в DTCM-RAM.

- **2-FlashITCM_rwSRAM1:** программа выполняется из Flash-ITCM в 7 состояниях ожидания с включенными ART и ART-предварительной выборкой, и данные загружаются / сохраняются в SRAM1 с включенным D-кешем.

- **3-FlashAXI_rwRAM-DTCM:** программа выполняется из Flash-AXI в 7 состояниях ожидания с включенным I-cache, и данные загружаются / сохраняются в DTCM-RAM. Кэш данных также включен, потому что некоторые константы загружаются из Flash-AXI для вычисления БПФ.

- **4-FlashAXI_rwSRAM1:** программа выполняется из Flash-AXI в 7 состояниях ожидания с включенным I-кешем, а данные загружаются / сохраняются в SRAM1 с включенным D-кешем.

- **5-RamITCM_rwRAM-DTCM:** программа выполняется из ITCM-RAM, а данные загружаются / сохраняются в DTCM-RAM. В этой конфигурации ничего не должно быть включено.

- **6-Quad SPI_rwRAM-DTCM:** эта конфигурация охватывает два случая, когда флэш-память Quad-SPI работает на частоте 54 МГц с включенным режимом DDR:

- Случай 1 (**6_1-Quad SPI_rwRAM-DTCM**): код выполняется из флэш-памяти Quad-SPI с включенным I-cache, и данные загружаются / сохраняются в DTCM-RAM. Поскольку в процессе БПФ используются огромные константы, данные, доступные только для чтения, в этом случае находятся во флэш-памяти Quad-SPI. Итак, D-cache также включен.

- Случай 2 (**6_2-Quad SPI_rwRAM-DTCM**): только код находится и выполняется во флэш-памяти Quad-SPI. Данные только для чтения находятся во Flash-ITCM. Данные для чтения / записи находятся в DTCM-RAM. В этом случае включены I-cache, ART и ART-prefetch.

- **7-ExtSDRAM-Swapped_rwDTCM:** эта конфигурация недоступна для рабочего пространства STM32F723E-Discovery. Здесь программа выполняется из FMC-SDRAM, а данные загружаются / сохраняются в DTCM-RAM. I-cache и D-cache включены. Обратите внимание, что в этой конфигурации постоянные данные помещаются в SDRAM, поэтому включен D-cache. В этой конфигурации

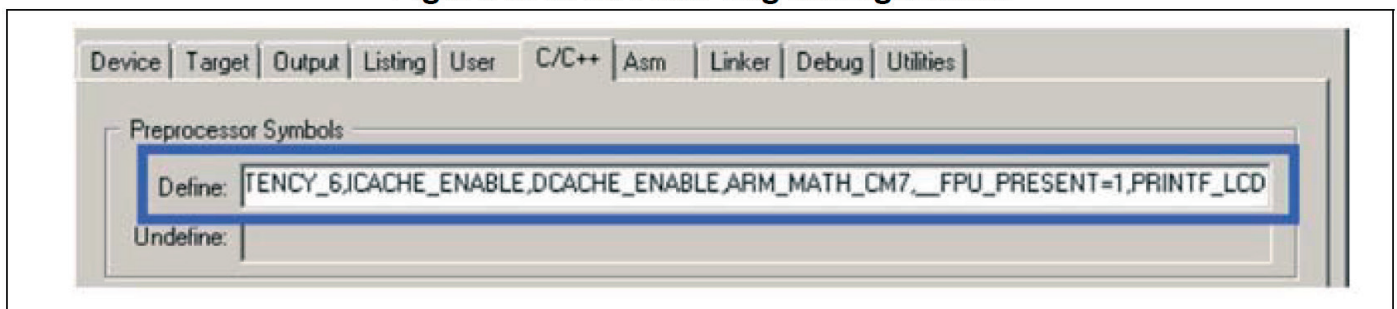
частота процессора составляет 200 МГц, в то время как SDRAM работает на частоте 100 МГц.

- **7-ExtPSRAM_rwDTCM:** эта конфигурация доступна только для рабочего пространства STM32F723E-Discovery. Здесь программа выполняется из FMC-PSRAM, а данные загружаются / сохраняются в DTCM-RAM. I-cache и D-cache включены. В этой конфигурации постоянные данные помещаются в PSRAM, поэтому D-кеш также включен.

- **8-FlashAXI-ROFTCM_rwSRAM1:** эта конфигурация доступна только для рабочего пространства STM32F723E-Discovery. Здесь только код находится и выполняется из Flash-AXI. Данные только для чтения извлекаются из Flash-ITCM. Данные для чтения / записи находятся в SRAM1. I-Cache, D-Cache, ART и предварительная выборка ART включены.

Каждая конфигурация имеет свой собственный флаг. Эти флаги можно установить в проекте конфигурации. На рисунке 9 показано, где эти флаги определены для цепочки инструментов MDK-ARM.

Figure 9. MDK-ARM flag configuration



Код оптимизирован для временного уровня 3 для всех конфигураций.

Описание флагов проекта:

- **DCACHE_ENABLE:** если он определен в проекте конфигурации, кэш данных включен.
- **ICACHE_ENABLE:** если он определен в проекте конфигурации, кэш инструкций включен.
- **ART_ENABLE:** если он определен в проекте конфигурации, ART Accelerator включен.
- **PF_ART_ENABLE:** если определено в проекте конфигурации, предварительная выборка ART Accelerator включена.
- **FLASH_WS:** настраивает количество состояний ожидания внутренней Flash:

FLASH_WS = FLASH_LATENCY_X, где X = от 0 (0 состояний ожидания) до 15 (15 состояний ожидания).

- **DATA_IN_ExtPSRAM:** если он определен в проекте конфигурации, внешний PSRAM настроен для использования либо для хранения данных, либо для выполнения кода.

- **PSRAM_MEM_BUS_WIDTH:** настраивает ширину шины внешнего PSRAM и имеет следующую конфигурацию:

PSRAM_MEM_BUS_WIDTH=FMC_NORSRAM_MEM_BUS_WIDTH_X, где X = 8, 16 или 32. PSRAM, доступный на плате STM32F723E-DISCO, является 16-битной платой PSRAM STM32F723E.

- **DATA_IN_ExtSDRAM:** если определено в проекте конфигурации, SDRAM настроена для использования либо для хранения данных, либо для выполнения кода.

- **SDRAM_MEM_BUS_WIDTH:** настраивает ширину шины внешней SDRAM и имеет следующую конфигурацию:

SDRAM_MEM_BUS_WIDTH = FMC_SDRAM_MEM_BUS_WIDTH_X, где X = 8, 16 или 32.

- **SDRAM_ADDRESS_SWAPPED:** если определено в проекте конфигурации, адрес SDRAM переназначается с 0xC000 0000 на 0x6000 0000. Переназначение SDRAM позволяет переместить его в кэшируемую область.

- **DATA_IN_QSPI:** если определено в проекте конфигурации, флэш-память Quad-SPI настроена для использования для хранения данных только для чтения или для выполнения кода.

- **QSPI_CLK_PRESCALER:** определяет предварительный делитель тактовой частоты Quad-SPI, и его конфигурация следующая:

QSPI_CLK_PRESCALER = X, где X = от 0 до 255.

- **QSPI_DDRMODE:** если он определен в проекте конфигурации, Quad-SPI настраивается в режиме DDR.

- **QSPI_INSTRUCTION_1_LINE, QSPI_INSTRUCTION_4_LINES:** первый флаг предназначен для настройки инструкции Quad-SPI Flash на одну строку, а второй - для настройки инструкции на четыре строки. Если флаг отсутствует, Quad-SPI будет настроен на выполнение одной строчной инструкции. Эти флаги недоступны для платы STM32F723E-DISCO.

- **QSPI_XIP_MODE:** если он определен в проекте конфигурации, Quad-SPI настраивается в режиме XIP, то есть сначала отправляется только инструкция. Обратите внимание, что режим XIP не влияет на производительность, когда кэш включен.

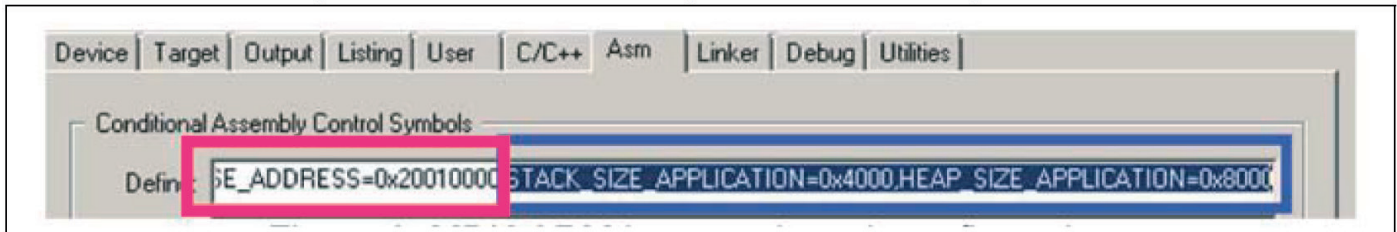
PRINTF_LCD, PRINTF_UART, PRINTF_VIEWER: эти флаги используются для отображения результатов демонстрации, соответственно, на ЖК-дисплее, через гипертерминал (115200, 7 бит, нечетная четность, один стоповый бит, без управления потоком HW) или в программе просмотра IDE printf. плату STM32F723E-DISCO следует использовать только PRINTF_UART и PRINTF_VIEWER. Дополнительные сведения об использовании различных режимов отображения см. В файле readme, который принадлежит соответствующему проекту во встроенном программном пакете X-CUBE-32F7PERF.

Пользователь может создавать новые конфигурации места выполнения кода / хранения данных на основе этих шаблонов: путем объединения соответствующих настроек, путем изменения файлов разброса или компоновщиков и путем установки соответствующего загрузчика Flash.

Также обратите внимание, что для цепочки инструментов MDK-ARM, чтобы изменить области RAM в файлах разброса, то есть области стека и кучи, пользователь должен соответствующим образом изменить размеры стека и кучи в меню ASM. Размер области в scatter-файле не считается реальным размером стека основного приложения. Пользователь должен изменить значения флагов **STACK_SIZE_APPLICATION** и **HEAP_SIZE_APPLICATION**, чтобы привести их значения в соответствие с областями размера кучи / стека, настроенными в файлах разброса.

На рисунке 10 показано, где изменить эти флаги (в синей рамке). Существует также начальный указатель стека, который используется, когда для хранения данных используются внешние запоминающие устройства. Его размер составляет 1 Кбайт, и его можно изменить с помощью переменной `Stack_Size_Init` в файлах запуска `startup_stm32f723xx.s`, `startup_stm32f756xx.s` и `startup_stm32f769xx.s`. Начальный базовый адрес указателя стека можно настроить в меню ASM, как показано на рисунке 10, выделенном розовым цветом.

Figure 10. MDK ARM heap and stack configurations



Scatter-файлы разных конфигураций находятся по пути MDK-ARM \ scatter_files в проекте демонстрации.

Для цепочки инструментов IAR (EWARM) компоновщики расположены в EWARM \ icf_files.

Для цепочки инструментов System Workbench компоновщики расположены в SW4STM32 \ <конфигурация папки проекта>.

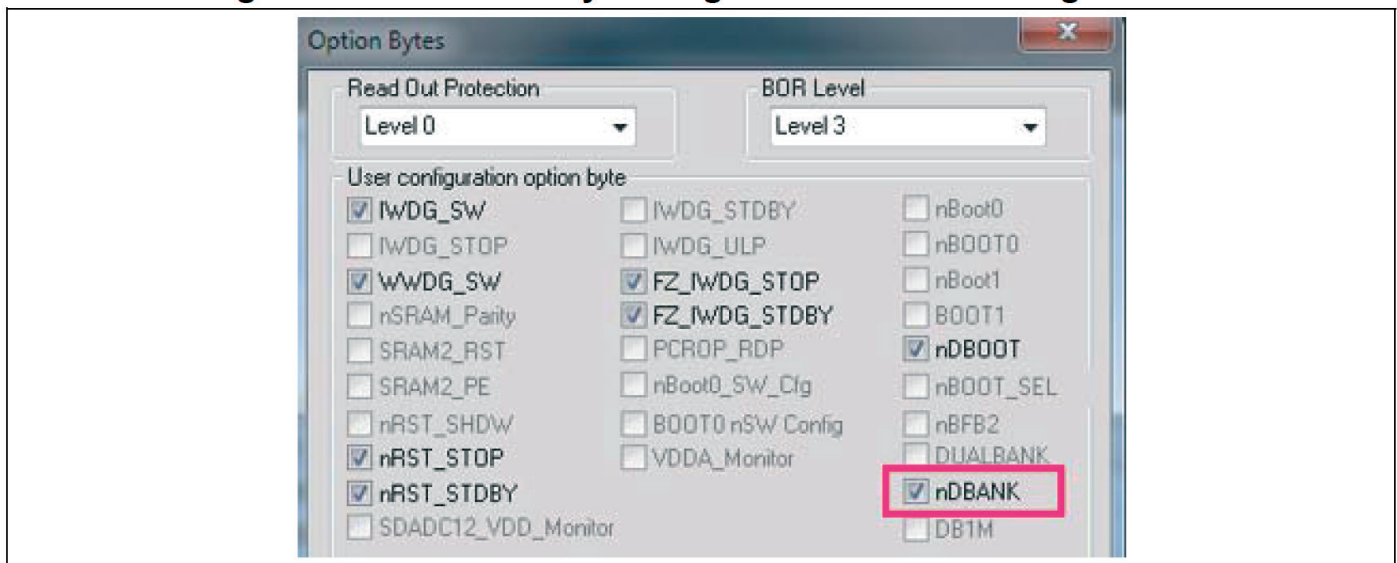
Также обратите внимание, что в шаблоне проекта прерывания выполняются из внутренней флэш-памяти: используется только прерывание системного тикового таймера.

В случае платы STM32F769I EVAL проект не содержит среды для конфигурации с двумя банками. Однако пользователь может использовать тот же проект в одном банке, но он должен выполнить два следующих шага:

1. Сконфигурируйте флэш-память в двойном банке с помощью утилиты ST-Link (v3.8.0 или новее). См. Рисунок 11.

Снимите флажок `nDBANK`, чтобы установить флэш-память в режим двух банков.

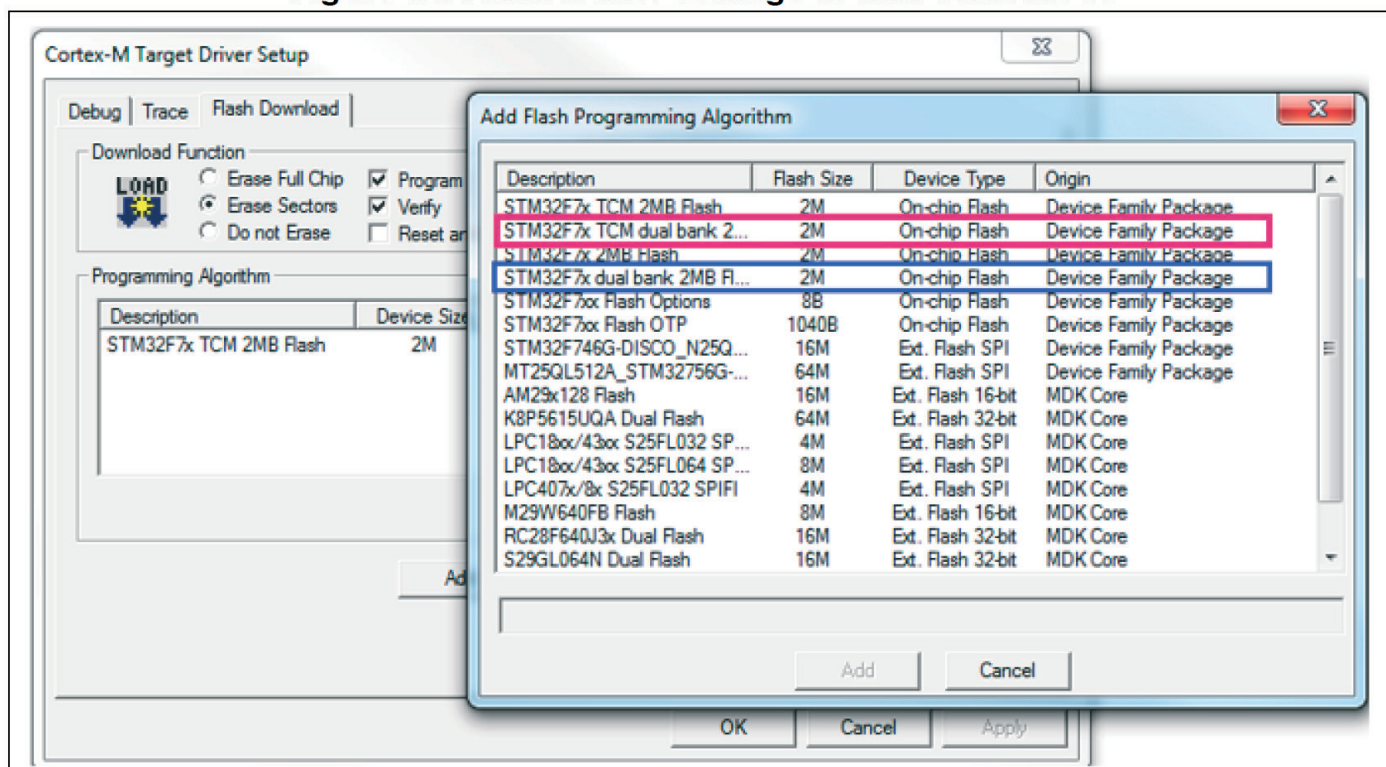
Figure 11. ST-Link utility settings for dual bank configuration



2. Установите двойной банк загрузчика Flash в цепочке инструментов. См. Рисунок 12.

Если пользователь использует Flash-TCM, необходимо выбрать загрузчик Flash (в розовой рамке). В случае использования Flash-AXI необходимо выбрать загрузчик Flash (в синей рамке), как показано на рисунке 12.

Figure 12. Flash loader settings in dual bank mode



Примечание. Пользователь может вернуться к конфигурации с одним банком, восстановив старые конфигурации загрузчика ST-Link / Flash.

2.3 Конфигурация проекта доступа к памяти ЦП с демонстрацией активации DMA

Демонстрация активации DMA предоставляется с инструментами Keil MDK-ARM, IAR Embedded Workbench и System Workbench для STM32. Конфигурации проекта основаны на том же шаблоне, который описан ранее в Разделе 2.2. У проекта есть три конфигурации:

1. FlashITCM_rwRAM-DTCM
2. FlashITCM_rwSRAM1
3. FlashAXI_rwRAM-DTCM

Демонстрация активации DMA использует те же конфигурации, что и демонстрация доступа к памяти ЦП, со следующими отличиями:

- Доступ к SDRAM используется для всех конфигураций передачи DMA.
- Используемая частота процессора составляет 200 МГц вместо 216 МГц для доступа к SDRAM.
- Файлы разброса изменяются для разделения памяти выполнения / данных, используемой процессом БПФ, и другими модулями демонстрации, чтобы избежать индуцированного компоновщиком недетерминизма, влияющего на результаты.
- Кэш данных включен для всех проектов.

Та же самая демонстрация БПФ, которая уже использовалась для производительности доступа к памяти ЦП (stm32f7_performances), использует-

ся для демонстрационной производительности активации DMA (stm32f7_performances_DMAs).

Сохранена только часть вычисления FFT, часть вычисления величины была удалена, чтобы выделить больше RAM для передач DMA.

В демонстрации используются два мастера для выполнения передачи: два DMA общего назначения: DMA1 и DMA2. DMA2 сконфигурирован для передачи данных из памяти в память, в то время как DMA1 используется для передачи данных из памяти в SPI3 и SPI3 для передачи из памяти. Источник и место назначения памяти настраиваются отдельно для каждого прямого доступа к памяти. Конфигурация выполняется в файле main.h, который содержит все необходимое определение для выполнения передачи DMA1 / 2.

Чтобы активировать передачу DMA1 / SPI3, SPI3_MOSI (PC12) должен быть подключен к SPI3_MISO (PC11). В случае использования плат STM32F7xxx-EVAL карту microSD необходимо вынуть из слота.

Чтобы получить результат для данного сценария, нужно выполнить три шага:

- Шаг 1: настройте различные параметры DMA в main.h
- Шаг 2: режим проверки: проверьте конфигурацию и передачу (и) DMA.
- Шаг 3: режим результатов: получить результаты по количеству циклов

2.3.1 Шаг 1: настройте различные параметры DMA

Для каждого мастера необходимо настроить четыре основных параметра:

- Включение / отключение передачи DMA
- Конфигурация адреса источника DMA
- Конфигурация адреса назначения DMA
- Конфигурация размера передачи DMA

Включение / отключение передачи DMA:

Чтобы включить или отключить DMA, определение «#define USE_DMA X » должно быть установлено на 1 или 0, чтобы соответственно включить или отключить DMA X . Где X = 1 или 2.

Пользователь может включить каждый DMA отдельно или включить их вместе, установив для определений USE_DMA1 и USE_DMA2 значение 1.

Конфигурации адреса источника и назначения DMA:

Адреса источника и назначения памяти должны быть настроены следующим образом:

Для DMA1:

#define DMA X _SRC_ADDRESS: начальный адрес источника памяти

#define DMA X _DST_ADDRESS: начальный адрес места назначения памяти

DMA1 передает данные из источника памяти в место назначения памяти через SPI3 (источник памяти -> DMA1_Stream5 -> SPI3_TX -> SPI3_RX -> DMA1_Stream0 -> место назначения памяти).

DMA2 передает данные напрямую из памяти A в память B.

Возможные значения DMA X _SRC_ADDRESS и DMA X _DST_ADDRESS:

- FLASHAXI_DMA_START_ADDRESS
- FLASHTCM_DMA_START_ADDRESS
- DTCMRAM_DMA_START_ADDRESS
- SRAM1_DMA_START_ADDRESS

–SRAM2_DMA_START_ADDRESS
 –SDRAM_DMA_START_ADDRESS ^(b)

Не изменяйте эти определения, предоставленные в файле dma_utilities.h, которые соответствуют отображению, настроенному в файлах разброса.

Конфигурация размера передачи DMA:

Размер передачи DMAX должен быть настроен следующим образом:

#define DMAX_TRANSFER_SIZE: размер передачи в байтах

Размер передачи должен быть наименьшим значением между доступным размером источника памяти DMA и доступным размером места назначения памяти DMA. Для этого предусмотрен макрос MIN (a, b). Это предотвращает доступ любого прямого доступа к памяти к недействительной зоне памяти.

Возможные значения DMAX_TRANSFER_SIZE:

–DTCMRAM_DMA_AVAILABLE_SIZE
 –SRAM1_DMA_AVAILABLE_SIZE
 –SRAM2_DMA_AVAILABLE_SIZE
 –SDRAM_DMA_AVAILABLE_SIZE ^(b)
 –FLASHAXI_DMA_AVAILABLE_SIZE
 –FLASHTCM_DMA_AVAILABLE_SIZE

Пример конфигурации размера передачи DMA2, где SRAM1 является источником, а DTCMRAM - местом назначения: #define DMA2_TRANSFER_SIZE MIN (SRAM1_DMA_AVAILABLE_SIZE, DTCMRAM_DMA_AVAILABLE_SIZE)

Если минимальный размер больше 64 Кбайт, размер передачи принудительно устанавливается равным 64 Кбайт.

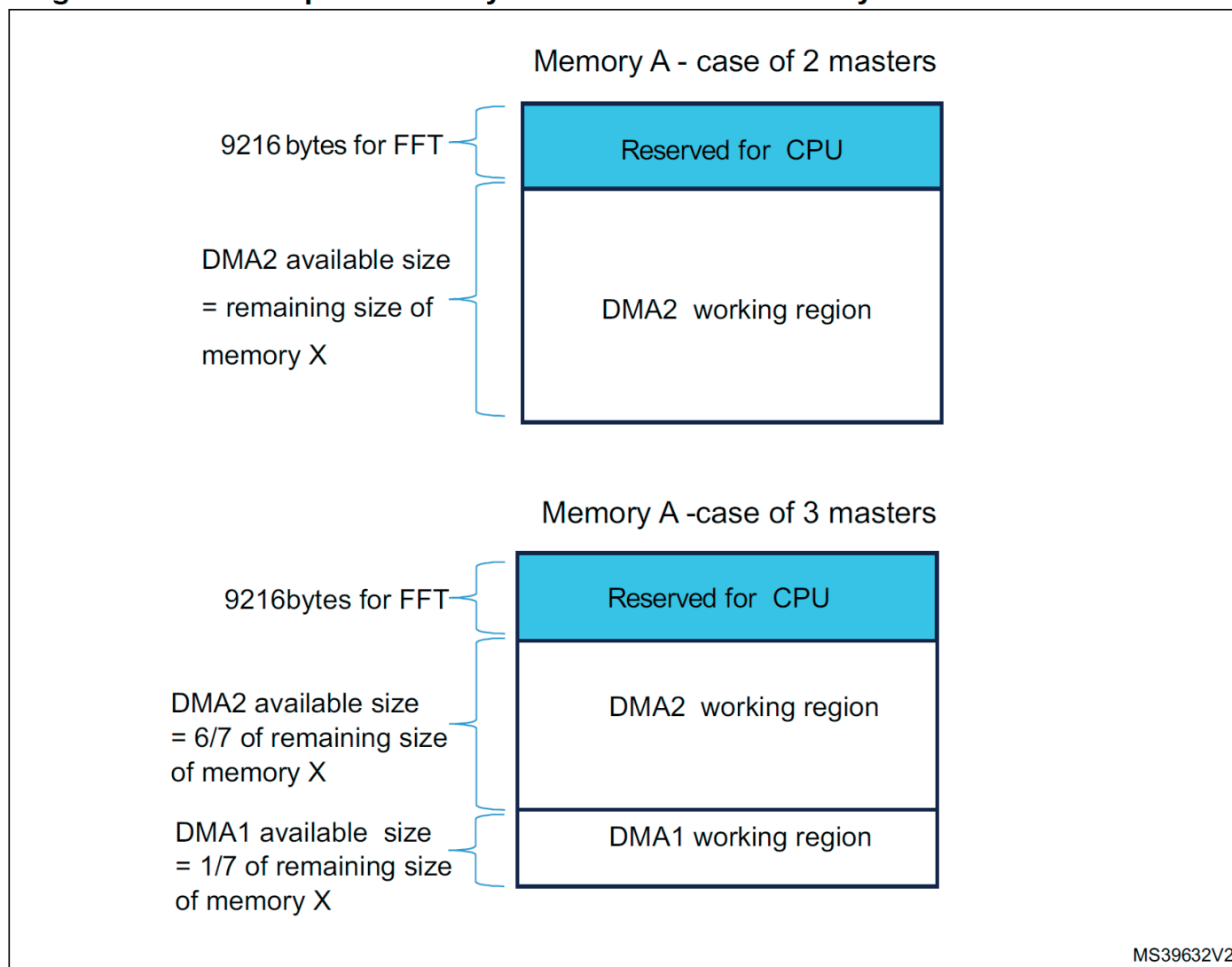
Когда пользователь запускает DMAX с одновременным использованием ЦП в памяти A, он должен разделить память, как показано на рисунке 13. На рисунке показано, как разделить память A (например, SRAM1) в случае параллелизма между двумя и тремя мастерами в программном обеспечении. Разделение в верхней части рисунка показывает память A, совместно используемую ЦП и DMA2. Разделение в нижней части рисунка показывает память A, совместно используемую тремя мастерами: CPU, DMA1 и DMA2. Это разделение позволяет каждому DMA завершить свою передачу после процесса БПФ, чтобы гарантировать, что параллелизм гарантирован во время всего процесса БПФ. Размер ОЗУ, зарезервированный для БПФ, составляет 9216 байт в scatter-файле (для доступа к ЦП).

Два примера ниже, чтобы показать, как настроить DMA1 и DMA2 в двух сценариях (см. Раздел 3.2: Производительность доступа к памяти ЦП с использованием DMA на стр. 38)

Configuration 2 /scenario 3:

```
/* DMA2 */
#define DMA2_SRC_ADDRESS      DTCMRAM_DMA_START_ADDRESS
#define DMA2_DST_ADDRESS      SRAM1_DMA_START_ADDRESS
#define DMA2_TRANSFER_SIZE    MIN (DTCMRAM_REMAINING_SIZE,
SRAM1_REMAINING_SIZE)
```

b. Недоступно для платы STM32F723E-DICO.

Figure 13. How to split a memory A in case of concurrency between all the masters

Configuration 1 /scenario 4:

```

/* DMA2 */
#define DMA2_SRC_ADDRESS      SRAM1_DMA_START_ADDRESS
#define DMA2_DST_ADDRESS      DTCMRAM_DMA_START_ADDRESS
#define DMA2_TRANSFER_SIZE    MIN (SRAM1_REMAINING_SIZE,
(DTCMRAM_REMAINING_SIZE*6/7))
/* DMA1 */
#define DMA1SPI_SRC_ADDRESS    SRAM2_DMA_START_ADDRESS
#define DMA1SPI_DST_ADDRESS    (DTCMRAM_DMA_START_ADDRESS +
(DTCMRAM_REMAINING_SIZE*6/7))
#define DMA1SPI_TRANSFER_SIZE MIN (SRAM2_REMAINING_SIZE,
(DTCMRAM_REMAINING_SIZE/7))

```

2.3.2 Шаг 2: проверьте конфигурацию и передачу (и) DMA

После настройки различных параметров DMA, пользователь должен проверить эту конфигурацию и передачу (и) DMA перед получением результатов, запустив пример в «режиме проверки».

Проверка выполняется для двух целей:

- Целостность передачи данных DMA правильная.

—Передача DMA завершается после процесса БПФ, чтобы гарантировать, что во время всего процесса БПФ DMA по-прежнему нагружает матрицу шины.

- Никакой аппаратной ошибки не произошло (ЖК-дисплей (или гипертерминал) отображает все конфигурации и горит как минимум один светодиод).

Если какое-либо из вышеперечисленных условий не выполняется, результат не согласуется, и пользователь должен изменить конфигурацию (и) DMA.

Чтобы запустить любой сценарий в «режиме проверки», следующее определение должно быть установлено на 1:

```
#define CHECK_TRANSFER      1
```

Затем скомпилируйте пример и запустите его.

Если три условия проверки соблюдены, на ЖК-дисплее (или на гипертерминале) отображается текущая конфигурация и горит зеленый светодиод. В этом случае выбранный сценарий готов к шагу 3 для получения результатов.

Если горит красный светодиод, есть по крайней мере одна передача DMA, которая не была выполнена должным образом. Сообщение об ошибке отображается на ЖК-дисплее (или на гипертерминале), показывая, какой DMA имеет ошибку и смещение неверных данных. На этом этапе пользователь должен проверить диапазоны адресов памяти, используемые DMA (ами), и перекомпилировать пример, чтобы горел только зеленый светодиод, прежде чем перейти к шагу 3. В случае передачи SPI1 проверьте Подключение SPI3_MOSI / SPI3_MISO.

Если красный светодиод мигает, целостность передачи данных правильная, но есть по крайней мере одна передача DMA, которая завершена до завершения передачи FFT. Отображается сообщение об ошибке, в котором указывается, в каком DMA возникла проблема. На этом этапе пользователь должен увеличить размер передачи этого прямого доступа к памяти.

Если произошел аппаратный сбой или приложение сломалось, это означает, что DMA перезаписал данные, используемые приложением, включая стек. Пользователь должен быть осторожен с выбором адресов, к которым обращается DMA, в основном, когда CPU, DMA1 и DMA2 обращаются к одной и той же памяти для воспроизведения параллелизма, представленного на рисунке 7.

2.3.3 Шаг 3: получите результаты

Если горит только зеленый светодиод и конфигурация отображается на ЖК-дисплее, результат будет согласованным и готовым к получению для текущего сценария.

На этом этапе пользователь может перекомпилировать пример в «режиме результатов» (CHECK_TRANSFER = 0), загрузить и запустить. ЖК-дисплей (или гипертерминал в случае платы STM32F723E-DISCO) отображает текущую конфигурацию, а также количество циклов в нижней части дисплея.

3 Результаты и анализ

3.1 Производительность доступа к памяти ЦП

В этом разделе объясняется активация каждой функции после используемой конфигурации и представлены полученные результаты в виде количества циклов, потребляемых процессом БПФ.

Для устройств STM32F74xxx и STM32F75xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.14.0, пакета STM32F7xx версии 1.1.0 и прошивки Cube версии 1.0.0.

Для устройств STM32F76xxx и STM32F77xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.17.0, пакета STM32F7xx версии 2.6.0 и прошивки Cube версии 1.4.0.

Для устройств STM32F72xxx и STM32F73xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.21.1, пакета STM32F7xx версии 2.9.0 и прошивки Cube версии 1.6.0.

Конфигурация оптимизации кода MDK-ARM - уровень 3 (оптимизация по времени).

Для всех конфигураций, которые имеют выполнение кода из Flash-ITCM, ускоритель ART и предварительная выборка ART включены, поскольку инструкции и / или данные только для чтения (постоянные переменные) проходят через ITCM (путь 1 на рисунке 2). Это случай следующих конфигураций:

- «1-FlashITCM_rwRAM-DTCM»
- «2-FlashITCM_rwSRAM1»
- «6_2-QuadSPI_rwRAM-DTCM» ^(b)
- «8-FlashAXI-ROFTCM_rwSRAM1» ^{(c) (d)}

Для всех конфигураций, в которых выполняется код из Flash-AXI (AXI / АНВ), кэш инструкций включен, поскольку инструкции проходят по пути 2 на рисунке 2. В этой конфигурации данные только для чтения (постоянные переменные) также проходят по пути 2 (тот же рисунок), поэтому кэш данных также включен. Это случай следующих конфигураций:

- «3-FlashAXI_rwRAM-DTCM»
- «4-FlashAXI_rwSRAM1»

В случае конфигурации «5-RamITCM_rwRAM-DTCM» ничего не активируется, поскольку код и данные находятся в ITCM-RAM и DTCM-RAM соответственно.

В случае конфигурации «6-Quad SPI_rwRAM-DTCM» присутствуют два случая, чтобы показать разницу в производительности между:

- Размещение данных только для чтения во флэш-памяти Quad-SPI, то есть в том же месте, что и инструкция (случай 1).
- При размещении данных только для чтения в памяти, отличной от флэш-памяти Quad-SPI, они будут находиться во Flash-ITCM (случай 2).

Для конфигурации «6_1-QuadSPI_rwRAM-DTCM», поскольку инструкции и данные находятся во флэш-памяти Quad-SPI для случая 1, включены только I-cache и D-cache.

Для конфигурации «6_2-QuadSPI_rwRAM-DTCM», поскольку данные только для чтения расположены во Flash-ITCM, ART и предварительная выборка ART включены, а D-кэш выключен. Для двух случаев Quad-SPI данные чтения / записи находятся в DTCM-RAM.

Для всех конфигураций, которые имеют доступ к DTCM-RAM в качестве памяти для хранения данных (путь 5 на рисунке 3), у них нет какой-либо конкрет-

c. По шине ITCM проходят только данные только для чтения.

Доступно только для плат:

d. STM32F723E-DISCO.

e. STM32F7xxx-EVAL.

f. STM32F723E-DISCO.

ной функции для активации, поскольку ядро напрямую связано с этой RAM с доступом в состоянии 0-ожидания.

Для всех конфигураций, которые имеют доступ к SRAM1 в качестве памяти для хранения данных (путь 7 на рисунке 3), кэш данных включен. Это справедливо для следующих конфигураций:

- «2-FlashITCM_rwSRAM1»
- «4-FlashAXI_rwSRAM1»
- «8-FlashAXI-ROFTCM_rwSRAM1»^(d)

Если приложение имеет доступ к внешней памяти через FMC для хранения данных, памяти данных и / или выполнения кода (путь 9 на рисунке 4), кэши данных и / или инструкций включаются. Это справедливо для следующей конфигурации:

- «7-ExtSDRAM-Swapped_rwDTCM»^(e)
- «7-ExtPSRAM_rwDTCM»^(f)

Примечание.

Для конфигурации «7-ExtSDRAM-Swapped_rwDTCM» SDRAM была заменена (переназначена с 0xC000 0000 на 0x6000 0000, чтобы разрешить использование кеша), поскольку область атрибутов MPU по умолчанию, начиная с 0xA000 0000 до 0xDFFF FFFF, не кэшируется поскольку это область типа памяти устройства.

Результаты

Результаты получены с платами STM32F723E-DISCO, STM32756G-EVAL и STM32769I-EVAL. ЦП работает на частоте 216 МГц, VDD = 3,3 В и имеет доступ к внутренней флеш-памяти в состоянии ожидания с 7 периодами ожидания.

Таблица 7, таблица 8 и таблица 9 показывают полученные результаты для демонстрации БПФ для MDK-ARM в каждой конфигурации для устройств STM32F74xxx / STM32F75xxx, STM32F72xxx / STM32F73xxx и STM32F76xxx / STM32F77xxx соответственно:

Table 7. MDK-ARM results for the STM32F74xxx and STM32F75xxx devices

Feature configuration	Memory location configuration	CPU cycle number ⁽¹⁾
ART + ART-PF ON	1-FlashITCM_rwRAM-DTCM	118577
ART + ART-PF + D-cache ON	2-FlashITCM_rwSRAM1	135296
I-cache + D-cache ON	3-FlashAXI_rwRAM-DTCM	118653
I-cache + D-cache ON	4-FlashAXI_rwSRAM1	145917
-	5-RAMITCM_rwRAM-DTCM	112428
I-cache + D-cache ON (Const data in Quad-SPI)	6_1-QuadSPI_rwRAM-DTCM	176441
I-cache + ART + ART-PF ON (Const data in Flash TCM)	6_2-QuadSPI_rwRAM-DTCM	126900
I-cache + D-cache ON (Const data in SDRAM)	7-ExtSDRAM-Swapped_rwDTCM ⁽²⁾	128398

1. The cycles number values may change from a version to another of the tool chain.

2. HCLK is running at 200 MHz.

Table 8. MDK-ARM results for the STM32F72xxx and STM32F73xxx devices

Feature configuration	Memory location configuration	CPU cycle number ⁽¹⁾
ART + ART-PF ON	1-FlashITCM_rwRAM-DTCM	123192
ART + ART-PF + D-cache ON	2-FlashITCM_rwSRAM1	120615
I-cache + D-cache ON	3-FlashAXI_rwRAM-DTCM	109667
I-cache + D-cache ON	4-FlashAXI_rwSRAM1	122785
-	5-RAMITCM_rwRAM-DTCM	106030
I-cache + D-cache ON (Const data in Quad-SPI)	6_1-QuadSPI_rwRAM-DTCM	149442
I-cache + ART + ART-PF ON (Const data in Flash TCM)	6_2-QuadSPI_rwRAM-DTCM	121630
I-cache + D-cache ON (Const data in PSRAM)	7-ExtPSRAM_rwDTCM	188340
I-cache + D-cache ON + ART + ART-PF ON (Const data in Flash TCM)	8-FlashAXI-ROFTCM_rwSRAM1	115437

1. The cycles number values may change from a version to another of the tool chain.

Table 9. MDK-ARM results for the STM32F76xxx and STM32F77xxx devices (single bank/single precision FPU)

Feature configuration	Memory location configuration	CPU cycle number ⁽¹⁾
ART + ART-PF ON	1-FlashITCM_rwRAM-DTCM	117059
ART + ART-PF + D-cache ON	2-FlashITCM_rwSRAM1	115004
I-cache + D-cache ON	3-FlashAXI_rwRAM-DTCM	108491
I-cache + D-cache ON	4-FlashAXI_rwSRAM1	112372
-	5-RAMITCM_rwRAM-DTCM	106275
I-cache + D-cache ON (Const data in Quad-SPI)	6_1-QuadSPI_rwRAM-DTCM	142177
I-cache + ART + ART-PF ON (Const data in Flash TCM)	6_2-QuadSPI_rwRAM-DTCM	120315
I-cache + D-cache ON (Const data in SDRAM)	7-ExtSDRAM-Swapped_rwDTCM ⁽²⁾	117744

1. The cycles number values may change from a version to another of the tool chain.

2. HCLK is running at 200 MHz.

Table 10. MDK-ARM results for the STM32F76xxx and STM32F77xxx devices (dual bank/single precision FPU)

Feature configuration	Memory location configuration	CPU cycle number ⁽¹⁾
ART + ART-PF ON	1-FlashITCM_rwRAM-DTCM	123237
ART + ART-PF + D-cache ON	2-FlashITCM_rwSRAM1	119645
I-cache + D-cache ON	3-FlashAXI_rwRAM-DTCM	109209
I-cache + D-cache ON	4-FlashAXI_rwSRAM1	112844

1. The cycles number values may change from a version to another of the tool chain.

В Table10 конфигурации 5, 6_1, 6_2 и 7 удалены, поскольку они не зависят от внутренней флэш-памяти. Они имеют ту же производительность, что и конфигурация с одним банком.

Диаграммы на Рисунке 14, Рисунке 15 и Рисунке 16 показывают относительное соотношение каждой конфигурации по сравнению с конфигурацией 5, которая имеет наилучшие результаты для устройств STM32F74xxx / STM32F75xxx, STM32F72xxx / STM32F73xxx и STM32F76xxx / STM32F77xxx соответственно.

Относительное соотношение = количество_циклов_конфиг_X / число_циклов_конфиг_5

Расчет относительного соотношения позволяет сравнить производительность с лучшей производительностью (5-RamITCM_rwRAM-DTCM) и сравнить конфигурацию с другой.

Анализ

Устройства STM32F74xxx и STM32F75xxx:

Если пользователь фиксирует расположение памяти данных в DTCM-RAM и изменяет расположение памяти для выполнения кода, относительные соотношения демонстрируют, что независимо от того, выполняется ли код из Flash-ITCM или Flash-AXI, производительность будет почти такой же, если ART или кэш инструкций соответственно включены.

Это случай конфигураций:

- «1-FlashITCM_rwRAM-DTCM»
- «3-FlashAXI_rwRAM-DTCM»

Если пользователь фиксирует расположение памяти выполнения во Flash-ITCM или Flash-AXI и изменяет расположение памяти для чтения / записи данных, соответствующие соотношения показывают, что независимо от того, какая внутренняя оперативная память используется для хранения данных, результаты будут почти одинаковыми. то же самое, поскольку кэш данных R / W включен для RAM, подключенных к AXI / АНВ.

Это случай пары конфигураций:

- «1-FlashITCM_rwRAM-DTCM» и «2-FlashITCM_rwSRAM1»
- «3-FlashAXI_rwRAM-DTCM» и «4-FlashAXI_rwSRAM1»

Если соотношение конфигурации «7-ExtSDRAM_Swapped_rwDTCM» сравнивается с конфигурациями «1-FlashITCM_rwRAM-DTCM» или «3-FlashAXI_rwRAM-DTCM», результаты близки, поскольку для этой конфигурации включен кэш. Таким образом, выполнение кода или сохранение данных из внешней памяти не влияет на производительность.

Обратите внимание, что случай конфигурации «5-RamITCM_rwRAM-DTCM» имеет лучшую производительность (наименьший номер цикла). Это связано с тем, что между шинами ITCM и DTCM нет конфликтов, и не выполняется обслуживание кеша (если нет использования кеша по сравнению с использованием кеша). Кроме того, в приведенном примере БПФ отсутствует одновременный доступ ЦП к DTCM-RAM с другими ведущими устройствами.

Если сравнить два отношения случая 1 и случая 2 конфигурации «6-Quad SPI_rwRAM-DTCM» (случай X), обратите внимание, что существует значительная разница в производительности, поскольку демонстрация использует огромные постоянные данные.

Для случая 1 (6_1-Quad SPI_rwRAM-DTCM), поскольку данные только для чтения и инструкции находятся во флэш-памяти Quad-SPI, возникает задержка из-за одновременного доступа выборки инструкций и данных только для чтения. загружается на шину матрицу.

Для случая 2 (6_2-Quad SPI_rwRAM-DTCM) данные только для чтения и код разделены. Данные, доступные только для чтения, находятся во Flash-TCM, поэтому одновременное выполнение данных, доступных только для чтения, и выборки инструкций исключается, и ЦП может одновременно получать инструкцию из AXI и данные, загруженные из TCM. Поэтому производительность второго случая явно лучше, чем первого.

Устройства STM32F76xxx и STM32F77xxx:

Для этих устройств анализ почти такой же, как для устройств STM32F74xxx и STM32F75xxx, за исключением случая, когда используется кэш.

Если сравнить конфигурацию 5 из Table7 и конфигурацию 5 из Table9, можно заметить, что производительность увеличена примерно на 5% для устройств STM32F76xxx и STM32F77xxx по сравнению с устройствами STM32F74xxx / STM32F75xxx. Это возможно благодаря некоторым улучшениям, которые были сделаны в FPU версии Cortex-M7, которая встроена в устройства STM32F76xxx и STM32F77xxx.

Благодаря размеру кэша устройств STM32F76xxx и STM32F77xxx производительность увеличивается во всех случаях с использованием кэша, SRAMx и пути AXI.

Например, конфигурации 3 и 4 занимают первое место в рейтинге по сравнению с устройствами STM32F74xxx и STM32F75xxx благодаря увеличению размера кэш-памяти с 4 Кбайт до 16 Кбайт.

В случае Quad-SPI, а точнее в конфигурации 6_2, производительность увеличивается примерно на 19,5% по сравнению с устройствами STM32F74xxx и STM32F75xxx, включая повышение производительности благодаря усовершенствованию FPU.

В случае внешних запоминающих устройств, таких как SDRAM, производительность также увеличивается примерно на 10% по сравнению с устройствами STM32F74xxx и STM32F75xxx благодаря увеличению размера кэш-памяти, включая повышение производительности благодаря усовершенствованию FPU.

В случае использования двух банков и даже если ширина интерфейса флэш-памяти уменьшена с 256 бит до 128 бит, производительность аналогична или лучше, чем производительность устройств STM32F74xxx и STM32F75xxx в некоторых случаях, когда флэш-память доступ через AXI / АНВ и кэш включен.

Table 11. Single bank versus dual bank performance comparison

Project Configuration	Single bank mode (CPU cycle number)	Dual bank mode (CPU cycle number)	Decrease dual bank vs Single bank (%)
1-FlashITCM_rwRAM-DTCM	117059	123237	5.28
2-FlashITCM_rwSRAM1	115004	119645	4.04
3-FlashAXI_rwRAM-DTCM	108491	109209	0.66
4-FlashAXI_rwSRAM1	112372	112844	0.42

Таблица 11 показывает, что в случае доступа к флэш-памяти через AXI / АНВ производительность одинакова как в режиме одного, так и в режиме двух банков. Если доступ к флэш-памяти осуществляется через TCM, производительность снижается примерно на 5% с текущим алгоритмом.

Устройства STM32F72xxx и STM32F73xxx:

Основываясь на Таблице 8, результаты STM32F72xxx и STM32F73xxx остаются почти такими же, как и для других устройств серии STM32F7, за исключением доступа к Flash-TCM. Устройства STM32F72xxx и STM32F73xxx имеют меньшую

ширину флэш-памяти (128 бит), чем другие устройства серии STM32F7, но имеют тот же ускоритель ART на 64 строки.

По сравнению с другими устройствами серии STM32F7, производительность STM32F72xxx и STM32F73xxx снижается, когда доступ к флэш-памяти осуществляется при чтении через интерфейс TCM. Они имеют ту же производительность, что и устройства STM32F76xxx / STM32F77xxx в режиме двух банков.

При доступе к флэш-памяти через интерфейс AXI производительность увеличивается по сравнению с доступом Flash-ITCM. В этом случае эффект кеширования компенсирует уменьшение ширины флэш-памяти. См. Результаты конфигурации «1-FlashITCM_rwRAM-DTCM» и «3-FlashAXI_rwRAM-DTCM» в таблице 8.

Производительность доступа к TCM-RAM остается такой же, как и для других устройств, поскольку нет разницы при доступе к этим RAM с точки зрения архитектуры.

Что касается результатов PSRAM, по сравнению с результатами SDRAM, полученными с другими устройствами серии STM32F7, производительность снижается, поскольку PSRAM, доступная на плате STM32F723-DISCO, имеет медленные тайминги и имеет доступ к 16-битным данным.

Конфигурация «8-FlashAXI-ROFTCM_rwSRAM1» показывает более высокую производительность, чем конфигурация «4-FlashAXI_rwSRAM1», за счет разделения данных только для чтения и инструкций. В конфигурации 8 доступ к данным только для чтения осуществляется через Flash-ITCM, а инструкции выбираются через Flash-AXI, в то время как в конфигурации 4 данные только для чтения и инструкции выбираются на том же интерфейсе шины, который является AXI. .

3.2 Производительность доступа к памяти ЦП при использовании DMA

В этом разделе рассматривается исполнение, в котором активированы один или несколько мастеров.

Для устройств STM32F74xxx и STM32F75xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.16.0, пакета STM32F7xx версии 2.2.0 и версии микропрограммы куба 1.0.0.

Для устройств STM32F76xxx и STM32F77xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.17.0, пакета STM32F7xx версии 2.6.0 и микропрограммы куба версии 1.4.0.

Для устройств STM32F72xxx и STM32F73xxx результаты получены с помощью цепочки инструментов KEIL MDK-ARM v5.21.1, пакета STM32F7xx версии 2.9.0 и прошивки Cube версии 1.6.0.

Конфигурация оптимизации кода MDK-ARM - уровень 3: оптимизация по времени.

Результаты

(Табл. 12, Табл. 13, Табл. 14), (Табл. 15, Табл. 16, Табл. 17) и (Табл. 18, Табл. 19, Табл. 20) суммируют количество циклов, потребляемых процессом БПФ в различные сценарии для устройств STM32F74xxx / STM32F75xxx, STM32F72xxx / STM32F73xxx и STM32F76xxx / STM32F77xxx соответственно, а также снижение производительности в процентах в случае, когда один или несколько мастеров обращаются к той же памяти, что и ЦП.

Результаты получены с платами STM32F723E-DISCO, STM32756G-EVAL и STM32769I-EVAL, центральным процессором, работающим на частоте 200 МГц (чтобы разрешить доступ к SDRAM @ 100 МГц), VDD = 3,3 В и с 6-периодным доступом к внутренней флеш-памяти. Расчетная ошибка измерения составляет 0,4% из-за некоторого выравнивания данных и инструкций во флэш-памяти и в кеше.

Условия получения результатов:

–Для уровней приоритета доступа DMA1 и DMA2 задан высокий приоритет, а размер данных – байт.

–Скорость передачи SPI3 составляет 25 Мбит / с.

–Регистр CM7_AHBSCR сохраняет значение сброса.

Устройства STM32F74xxx и STM32F75xxx:

Table 12. Configuration 1: execution from the Flash-ITCM / FFT CPU data storage in the DTCM-RAM

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	69247	-
	DMA1_SPI3	-	-		
2	DMA2	SDRAM	SRAM1	69247	0
	DMA1_SPI3	-	-		
3	DMA2	SRAM1	DTCM-RAM⁽¹⁾	70929	2.43
	DMA1_SPI3	-	-		
4	DMA2	SRAM1	DTCM-RAM⁽¹⁾	71275	2.93
	DMA1_SPI3	SRAM2			
5	DMA2	SDRAM	DTCM-RAM⁽¹⁾	70830	2.29
	DMA1_SPI3	SRAM2			
6	DMA2	DTCM-RAM⁽²⁾	SRAM1	70140	1.29
	DMA1_SPI3	-	-		
7	DMA2	Flash-AXI⁽²⁾	SRAM1	69596	0.50
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).

2. Memories with a concurrency on read access between CPU and DMA(s).

Table 13. Configuration 2: execution from the Flash-ITCM / FFT CPU data storage in the SRAM1

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	79925	-
	DMA1_SPI3	-	-		
2	DMA2	SDRAM	SRAM2	79925	0
	DMA1_SPI3	-	-		
3	DMA2	DTCM-RAM	SRAM1⁽¹⁾	81720	2.25
	DMA1_SPI3	-	-		
4	DMA2	DTCM-RAM	SRAM1⁽¹⁾	81828	2.38
	DMA1_SPI3	SRAM2			
5	DMA2	SDRAM	SRAM1⁽¹⁾	81674	2.19
	DMA1_SPI3	SRAM2			
6	DMA2	SRAM1⁽²⁾	DTCM-RAM	81410	1.86
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).

2. Memories with a concurrency on read access between CPU and DMA(s).

Table 14. Configuration 3: execution from the Flash-AXI / FFT CPU data storage in the DTCM-RAM

Scenario	Master	Source	Destination	FFT cycles number	Degradation (%)
1	DMA2	-	-	68206	-
	DMA1_SPI3	-	-		
2	DMA2	Flash-AXI⁽¹⁾	SRAM1	68865	0.97
	DMA1_SPI3	-	-		

1. Memories with a concurrency on read access between CPU and DMA(s).

STM32F72xxx and STM32F73xxx devices

Table 15. Configuration 1: execution from the Flash-ITCM / FFT CPU data storage in the DTCM-RAM

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	73406	-
	DMA1_SPI3	-	-		
2	DMA2	SRAM1	DTCM-RAM⁽¹⁾	74568	1.58
	DMA1_SPI3	-	-		

Table 15. Configuration 1: execution from the Flash-ITCM / FFT CPU data storage in the DTCM-RAM (continued)

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
3	DMA2	SRAM1	DTCM-RAM⁽¹⁾	74788	1.88
	DMA1_SPI3	SRAM2			
4	DMA2	DTCM-RAM⁽²⁾	SRAM1	74321	1.25
	DMA1_SPI3	-	-		
5	DMA2	Flash-AXI⁽²⁾	SRAM1	74214	1.10
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).

2. Memories with a concurrency on read access between CPU and DMA(s).

Table 16. Configuration 2: execution from the Flash-ITCM / FFT CPU data storage in the SRAM1

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	78416	-
	DMA1_SPI3	-	-		
2	DMA2	DTCM-RAM	SRAM1⁽¹⁾	79129	0.91
	DMA1_SPI3	-	-		
3	DMA2	DTCM-RAM	SRAM1⁽¹⁾	79145	0.93
	DMA1_SPI3	SRAM2			
4	DMA2	SRAM1⁽²⁾	DTCM-RAM	79115	0.89
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).

2. Memories with a concurrency on read access between CPU and DMA(s).

Table 17. Configuration 3: execution from the Flash-AXI / FFT CPU data storage in the DTCM-RAM

Scenario	Master	Source	Destination	FFT cycles number	Degradation (%)
1	DMA2	-	-	66374	-
	DMA1_SPI3	-	-		
2	DMA2	Flash-AXI⁽¹⁾	SRAM1	66982	0.92
	DMA1_SPI3	-	-		

1. Memories with a concurrency on read access between CPU and DMA(s).

STM32F76xxx and STM32F77xxx devices:**Table 18. Configuration 1: execution from the Flash-ITCM / FFT CPU data storage in the DTCM-RAM (single bank/single precision FPU)**

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	69250	-
	DMA1_SPI3	-	-		
2	DMA2	SDRAM	SRAM1	69250	0
	DMA1_SPI3	-	-		
3	DMA2	SRAM1	DTCM-RAM⁽¹⁾	70873	2.34
	DMA1_SPI3	-	-		
4	DMA2	SRAM1	DTCM-RAM⁽¹⁾	71191	2.80
	DMA1_SPI3	SRAM2			
5	DMA2	SDRAM	DTCM-RAM⁽¹⁾	70825	2.27
	DMA1_SPI3	SRAM2			
6	DMA2	DTCM-RAM⁽²⁾	SRAM1	70150	1.30
	DMA1_SPI3	-	-		
7	DMA2	Flash-AXI⁽²⁾	SRAM1	69587	0.49
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).

2. Memories with a concurrency on read access between CPU and DMA(s).

Table 19. Configuration 2: execution from the Flash-ITCM / FFT CPU data storage in the SRAM1 (single bank/single precision FPU)

Scenario	Master	Source	Destination	FFT cycles number	Decrease (%)
1	DMA2	-	-	72648	-
	DMA1_SPI3	-	-		
2	DMA2	SDRAM	SRAM2	72648	0
	DMA1_SPI3	-	-		
3	DMA2	DTCM-RAM	SRAM1⁽¹⁾	73071	0.58
	DMA1_SPI3	-	-		
4	DMA2	DTCM-RAM	SRAM1⁽¹⁾	73086	0.60
	DMA1_SPI3	SRAM2			
5	DMA2	SDRAM	SRAM1⁽¹⁾	72964	0.43
	DMA1_SPI3	SRAM2			
6	DMA2	SRAM1⁽²⁾	DTCM-RAM	73082	0.60
	DMA1_SPI3	-	-		

1. Memories with a concurrency on write access between CPU and DMA(s).
2. Memories with a concurrency on read access between CPU and DMA(s).

Table 20. Configuration 3: execution from the Flash-AXI / FFT CPU data storage in the DTCM-RAM (single bank/single precision FPU)

Scenario	Master	Source	Destination	FFT cycles number	Degradation (%)
1	DMA2	-	-	65897	-
	DMA1_SPI3	-	-		
2	DMA2	Flash-AXI ⁽¹⁾	SRAM1	66155	0.39
	DMA1_SPI3	-	-		

1. Memories with a concurrency on read access between CPU and DMA(s).

Table 21. Configuration 3: execution from the Flash-AXI / FFT CPU data storage in the DTCM-RAM (dual bank/single precision FPU)

Scenario	Master	Source	Destination	FFT cycles number	Degradation (%)
1	DMA2	-	-	66370	-
	DMA1_SPI3	-	-		
2	DMA2	Flash-AXI ⁽¹⁾	SRAM1	66960	0.89
	DMA1_SPI3	-	-		

1. Memories with a concurrency on read access between CPU and DMA(s).

Анализ

Устройства STM32F74xxx и STM32F75xxx:

1. Если ЦП и один или несколько мастеров не имеют доступа к одной и той же памяти, конфликт отсутствует. Тот же результат получается из сценариев 1 и 2 для конфигураций 1 и 2. См. Результаты, представленные следующим образом:

- Конфигурация 1 / сценарий 2 по сравнению с конфигурацией 1 / сценарием 1.
- Конфигурация 2 / сценарий 2 по сравнению с конфигурацией 2 / сценарием 1.

2. Когда ЦП и один DMA обращаются к одной и той же памяти, возникает конфликт, поскольку существует арбитраж либо на матрице шины для запоминающих устройств, подключенных к матрице шины, включая внешние запоминающие устройства (SRAM1, SRAM2, SDRAM и т. Д.), Либо к сторона ЦП (интерфейс Tightly-Coupled Unit: блок TCU), где параллелизм выполняется на уровне DTCM-RAM. См. Результаты, предоставленные:

- Конфигурация 1 / сценарий 1 по сравнению с конфигурацией 1 / сценарием 3.
- Конфигурация 2 / сценарий 1 по сравнению с конфигурацией 2 / сценарием 3.

Как показывают результаты, параллелизм ЦП и DMA2 при записи в DTCM-RAM составляет около 2,5% от снижения производительности.

В случае SRAM1 / SRAM2 параллелизм ЦП и DMA2 при доступе на запись составляет около 2,5% от снижения производительности.

3. Задержка увеличивается примерно на 0,5%, когда дополнительный мастер имеет доступ на запись к той же памяти, что и ЦП. См. Результаты, предоставленные:

- Конфигурация 1 / сценарий 3 по сравнению с конфигурацией 1 / сценарием 4.
- Конфигурация 2 / сценарий 3 по сравнению с конфигурацией 1 / сценарием 4.

4. Эффект конкуренции при чтении данной памяти между ЦП и DMA меньше, чем эффект конкуренции при доступе на запись. См. Результаты, предоставленные:

- Конфигурация 1 / сценарий 3 по сравнению с конфигурацией 1 / сценарием 6.
- Конфигурация 2 / сценарий 3 по сравнению с конфигурацией 2 / сценарием 6.

Как показывают результаты, параллелизм ЦП и DMA2 при доступе для чтения на уровне DTCM-RAM составляет от 1% до 2% снижения производительности.

В случае SRAM1 / SRAM2 параллелизм ЦП и DMA2 при доступе для чтения составляет от 1,5% до 2,5% снижения производительности.

5. Если источник памяти относительно медленный, такой как SDRAM, эффект параллелизма уменьшается, поскольку в этой памяти есть некоторая задержка, которая обеспечивает полосу пропускания для доступа ЦП к этой памяти на матрице шины. См. Результаты, предоставленные:

- Конфигурация 1 / сценарий 4 по сравнению с конфигурацией 1 / сценарием 5.
- Конфигурация 2 / сценарий 4 по сравнению с конфигурацией 2 / сценарием 5.

В конфигурации 1 DTCM-RAM в сценарии 4 заменяется SDRAM в сценарии 5. Тот же случай применяется для конфигурации 2.

6. Отсутствует значительная конкуренция при выполнении данных из Flash-AXI или Flash-ITCM и одновременном выполнении передачи из флэш-памяти в другую память. Это связано с тем, что доступ DMA к флэш-памяти медленный по сравнению с доступом ЦП, который использует кэш. Доступ DMA к флэш-памяти каденсируется количеством состояний ожидания, настроенных в интерфейсе флэш-памяти (в нашем случае шесть состояний ожидания), в то время как инструкции / данные загружаются в кэш на стороне ЦП или в ART в случае доступа к Flash-ITCM. Эта конфигурация позволяет отменить эффекты задержки. См. Результаты, предоставленные:

- Конфигурация 1 / сценарий 7 по сравнению с конфигурацией 3 / сценарием 2.

Поскольку DMA1 может выполнять передачи между памятью и периферийным устройством, влияние параллелизма на память между ЦП и DMA1 зависит от скорости периферийного устройства. Чем быстрее периферийное устройство, тем заметнее влияние параллелизма. Например, если SPI3 с максимальной скоростью 25 МГц заменяется SPI4 с максимальной скоростью 50 МГц, задержка будет более важной, поскольку полоса пропускания передачи данных высока. Тем не менее, задержка по-прежнему разумная.

Таким образом, наблюдается очень низкое снижение производительности, примерно на 3%, из-за параллелизма между ЦП и DMA во всех сценариях для всех различных операций доступа к памяти при чтении или записи (DTCM-RAM, SRAMx, внешняя память). На общую производительность не влияет параллелизм с несколькими мастерами благодаря интеллектуальной архитектуре системы и матрице шины.

Устройства STM32F76xxx и STM32F77xxx:

Для устройств STM32F76xxx и STM32F77xxx анализ почти такой же, как для устройств STM32F74xxx и STM32F75xxx, за исключением случаев, когда AXI / AHB используется для доступа к ЦП.

В Таблице 18 производительность такая же, как у устройств STM32F74xxx и STM32F75xxx (см. Таблицу 12), поскольку устройства имеют одинаковую конфигурацию архитектуры.

Но в Table19 снижение производительности не превышает 0,6%, поскольку параллельный доступ к памяти между ЦП и одним или несколькими мастерами не влияет на производительность. Это возможно благодаря эффекту увеличения размера кеш-памяти устройств STM32F76xxx и STM32F77xxx.

В случае конфигурации флэш-памяти с двумя банками (см. Таблицу 21) и если параллельный доступ выполняется к флэш-памяти, снижение производительности такое же, как у устройств STM32F74xxx и STM32F75xxx. Однако в случае конфигурации флэш-памяти с одним банком (см. Таблицу 20) снижение производительности незначительно, как если бы не было одновременного доступа.

Устройства STM32F72xxx и STM32F73xxx:

Для устройств STM32F72xxx и STM32F73xxx в целом анализ остается таким же, как и для других устройств. Размер кеша напрямую влияет на производительность при параллельном доступе при чтении или записи из / в SRAM1 или SRAM2. Кроме того, если есть параллелизм при чтении на Flash-AXI (см. Таблицу 17), и даже если ширина Flash меньше, чем ширина других устройств серии STM32F7, эффекты параллелизма все еще очень низкие благодаря размеру кэша, который был увеличен по сравнению с устройствами STM32F74xxx и STM32F75xxx.

Таким образом, чем больше размер кэша, тем меньше конфликтных эффектов в памяти.

4 Рейтинг и сравнение производительности внутри устройств серии STM32F7

В этом разделе представлен обзор глобального позиционирования устройства STM32F7xxxx внутри устройств серии STM32F7 и того, как выбрать устройство STM32F7xxxx в соответствии с потребностями приложения с точки зрения производительности и стоимости.

4.1 Рейтинг производительности внутри серии STM32F7

Размер кеша очень важен и напрямую влияет на производительность: чем больше размер кеша, тем быстрее выполняется приложение.

Когда код выполняется из Flash-AXI и данные находятся в SRAM1, производительность от одного устройства к другому зависит от размера кэша. Например, в случае конфигураций «3-FlashAXI_rwRAM-DTCM» и «4-FlashAXI_rwSRAM1»:

- Устройства STM32F76xxx и STM32F77xxx имеют лучшую производительность, так как в них встроен самый большой размер кеш-памяти - 16 Кбайт,
- Устройства STM32F74xxx и STM32F75xxx имеют самую низкую производительность, поскольку в них встроен минимальный размер кэша 4 Кбайта.
- Устройства STM32F72xxx и STM32F73xxx имеют промежуточный рейтинг

производительности из-за размера кэша 8 Кбайт. С этими устройствами, обращаясь к Flash через шину AXI, размер кэша компенсирует уменьшение ширины Flash (128 бит) по сравнению с 256-битным доступом к Flash других устройств.

Такой же рейтинг получен для конфигурации «6_1-QuadSPI_rwRAM-DTCM», где код выполняется из Quad-SPI Flash.

Ширина флэш-памяти также влияет на производительность, особенно когда данные / инструкции извлекаются из шины TCM (Flash-ITCM). В случае конфигурации «1-FlashITCM_rwRAM-DTCM» устройства STM32F72xxx и STM32F73xxx имеют самую низкую производительность из-за их 128-битного доступа к флэш-памяти по сравнению с другими устройствами серии STM32F7 с 256-битным доступом.

Таблица 22 суммирует результаты, полученные для всех устройств серии STM32F7.

Table 22. Performance comparison between STM32F7 Series devices

Memory location configuration	STM32F74xxx/ STM32F75xxx (1)	STM32F72xxx/ STM32F73xxx (1)	STM32F76xxx/ STM32F77xxx Single bank (1)	STM32F76xxx/ STM32F77xxx Dual bank (1)
1-FlashITCM_rwRAM-DTCM	118577	123192	117059	123237
2-FlashITCM_rwSRAM1	135296	120615	115004	119645
3-FlashAXI_rwRAM-DTCM	118653	109667	108491	109209
4-FlashAXI_rwSRAM1	145917	122785	112372	112844
5-RAMITCM_rwRAM-DTCM	112428	106030	106275	-
6_1-QuadSPI_rwRAM-DTCM	176441	149442	142177	-

1. Expressed in cycle number.

Рейтинг производительности по устройству STM32F7 и по доступу к памяти с использованием того же алгоритма

Рейтинг представлен от самого высокого до самого низкого показателей:

Доступ к Flash-ITCM (читается) + ART включен:

1. устройства STM32F76xxx и STM32F77xxx в режиме одного банка.
2. Устройства STM32F74xxx и STM32F75xxx.
3. Устройства STM32F72xxx и STM32F73xxx / устройства STM32F76xxx и STM32F77xxx в режиме двух банков.

Доступ к Flash-AXI (читается) + кэш включен:

1. Устройства STM32F76xxx и STM32F77xxx в режиме одного банка.
2. Устройства STM32F76xxx и STM32F77xxx в режиме двух банков.
3. Устройства STM32F72xxx и STM32F73xxx / Устройства STM32F74xxx и STM32F75xxx (g).
4. Устройства STM32F74xxx и STM32F75xxx (ж).

Доступ к SRAM1 и SRAM2 + кэш включен

- 1.устройства STM32F76xxx и STM32F77xxx.
2. Устройства STM32F72xxx и STM32F73xxx.
3. Устройства STM32F74xxx и STM32F75xxx.

Доступ к TCM-RAMs:

Все устройства имеют одинаковую производительность.

Как указывалось ранее, в устройствах STM32F76xxx / STM32F77xxx и устройствах STM32F72xxx / STM32F73xxx есть усовершенствование FPU Cortex-M7 по сравнению с устройствами STM32F74xxx / STM32F75xxx, что объясняет различия в количестве циклов с демонстрацией FFT в случае конфигурация «5-RamITCM_rwRAM-DTCM» (см. таблицу 22).

Доступ к внешней памяти + Кэш включен:

1. устройства STM32F76xxx и STM32F77xxx.
2. Устройства STM32F72xxx и STM32F73xxx.
3. Устройства STM32F74xxx и STM32F75xxx.

4.2 Рекомендации по выбору устройства STM32F7

Если пользователь ищет наилучшую производительность с наибольшим объемом внутренней памяти, устройства STM32F76xxx и STM32F77xxx подходят для его применения.

Если пользователь ищет меньшую производительность, меньший объем внутренней памяти при минимальной стоимости, выбором могут быть устройства STM32F72xxx, STM32F73xxx, STM32F74xxx или STM32F75xxx.

Однако, если пользователь нацелен на графические приложения с меньшей производительностью, чем устройства STM32F76xxx и STM32F77xxx, устройства STM32F74xxx и STM32F75xxx более подходят, чем устройства STM32F72xxx и STM32F73xxx, которые не поддерживают DMA2D и LCD-TFT DMA.

Для высокопроизводительных графических приложений устройства STM32F76xxx и STM32F77xxx более подходят, чем устройства STM32F74xxx и STM32F75xxx, поскольку они имеют интерфейс DSI, обеспечивающий высокую скорость передачи графических данных.

Если приложению требуется подключение к сети Ethernet, устройства STM32F74xxx, STM32F75xxx, STM32F76xxx и STM32F77xxx подходят для пользовательского приложения, поскольку устройства STM32F72xxx и STM32F73xxx не имеют периферийных устройств Ethernet.

Устройства STM32F7xxx включают высокоскоростной интерфейс USB OTG PHY, который недоступен во всех других устройствах серии STM32F7.

5 Программная разметка памяти и советы

В этом разделе представлены несколько советов о том, как разделить код и данные в памяти STM32F7, чтобы получить наилучшую компромиссную производительность в зависимости от кода и размеров данных.

5.1 Разбиение программной памяти на разделы

Поскольку ЦП имеет прямой доступ к памяти TCM с 64-битным доступом и доступом в состоянии 0-ожидания, ячейки DTCM-RAM и ITCM-RAM являются лучшими местами для данных чтения / записи и выборки команд соответственно.

Таким образом, ITCM-RAM (16 Кбайт) зарезервирован для критического кода с детерминированным исполнением, такого как обработчики прерываний, которые не могут ждать промахов в кэше, и некоторые критические контуры управления, предназначенные для приложений управления двигателем.

g. Устройства STM32F74xxx / STM32F75xxx и устройства STM32F72xxx / STM32F73xxx могут иметь одинаковую производительность в зависимости от используемого алгоритма.

DTCM-RAM зарезервирована для операций хранения данных с регулярным доступом и для критически важных данных в реальном времени, таких как стек и куча. В приложениях реального времени, использующих ОСПВ, как правило, массово используется куча. Например, если DTCM-RAM составляет 64 Кбайта и этого размера достаточно для приложения, разброс DTCM-RAM (32 Кбайт для кучи, 8 Кбайт для стека и 24 Кбайт для глобальных переменных) будет относительно разумным. В приложениях переднего плана / фона куча почти не используется, DTCM-RAM будет разбросана между стеком и глобальными переменными.

Когда размер кода пользовательского приложения помещается во внутреннюю флэш-память, последняя также будет лучшей областью выполнения:

- Через TCM (Flash-ITCM) при включении ART-ускорителя или
- Через AXI / АНВ путем включения кеша для достижения состояния 0-ожидания на частоте 216 МГц.

Обратите внимание, что выполнение из Flash-ITCM / data в DTCM-RAM и Flash-AXI / data в DTCM-RAM имеет одинаковую оценку CoreMark, которая составляет 5 CoreMark / МГц.

SRAM1 может быть зарезервирован для буфера графических кадров в графических приложениях, использующих QVGA TFT в 16-битном режиме, который требует относительно большого объема графических данных и производительности отображения, достигаемой с помощью LCD-TFT и DMA2D DMA. Эту память также можно использовать для операций хранения данных, когда в DTCM-RAM больше нет свободного места. В этом случае область из SRAM1 с включенным кешем данных может быть зарезервирована для глобальных переменных, чтобы оставить больше места для критических данных.

SRAM2 может быть зарезервирован для периферийных устройств, таких как Ethernet и USB, для хранения данных, таких как буферы и дескрипторы. Эта память также может использоваться для глобальных переменных.

Когда приложению требуется больше памяти, а его код или данные, или и то, и другое не помещаются во внутреннюю память, внешнюю память можно использовать для увеличения размера памяти без потери производительности.

Например, внешняя флэш-память NOR размером до 250 Мбайт, подключенная через FMC, может содержать инструкции приложения с включенным кешем инструкций, в то время как постоянные данные находятся во внутренней флэш-памяти (путем включения ART или кеша). Это тот случай, когда в приложении используются огромные константы, чтобы избежать параллельного доступа к инструкциям и данным по одному и тому же пути (путь 9 на рисунке 4) в матрице шины.

В случае нехватки внутренней RAM хранение данных может быть достигнуто во внешней SRAM или SDRAM через интерфейс FMC и путем включения кеширования данных. Эта память может содержать либо буферы кадров для графических приложений, либо для некритичных данных. В то же время больший приоритет отдается очень важным данным, которые должны быть помещены в DTCM-RAM.

Флэш-память Quad-SPI может использоваться либо для хранения данных только для чтения (относительно огромные файлы изображений или волн), либо для поддержания почти того же уровня производительности, что и доступ к внутренней флэш-памяти, путем включения кеширования данных.

Флэш-память Quad-SPI также может использоваться для хранения приложения в режиме отображения памяти до 256 Мбайт и в то же время для сохранения нескольких GPIO в меньших пакетах STM32F7 по сравнению с параллельными флэш-памятью, которые должны быть подключены к интерфейсу FMC. . В том случае, когда ЦП регулярно обращается к данным только для чтения, последние должны отображаться во внутренней флэш-памяти. Если приложению требуется больший размер и большая производительность выполнения, пользователь может загрузить свое приложение в Quad-SPI (область загрузки) и использовать внешнюю SDRAM, где приложение копируется и выполняется (область выполнения).

5.2 Советы

В случае, когда DMA обращается к DTCM-RAM одновременно с ЦП, скорость приложения может снизиться, если ЦП не имеет доступа с наивысшим приоритетом. Этим приоритетом можно управлять программно, используя регистр CM7_AHBSCR в критическом разделе кода, который загружает / сохраняет данные в DTCM-RAM.

Чтобы получить максимальную производительность с устройствами STM32F76xxx и STM32F77xxx, а также в случае использования двух банков, рекомендуется обращаться к флэш-памяти через интерфейс AXI / АНВ вместо TCM, чтобы воспользоваться преимуществами производительности, вызванной размером кеша. .

Чтобы получить максимальную производительность с устройствами STM32F72xxx и STM32F73xxx, рекомендуется обращаться к флэш-памяти через интерфейс AXI / АНВ вместо TCM, чтобы воспользоваться преимуществами производительности, вызванной размером кеша.

Когда для области выполнения используется внешняя память, следует проявлять осторожность, если эта память отображается в области, имеющей атрибут по умолчанию Execute-Never (XN). В этом случае пользователь должен преобразовать память в исполняемую область с помощью MPU, иначе произойдет серьезная ошибка. Это случай регионов банка SDRAM после сброса. Обратитесь к Таблице 2, чтобы узнать о исполняемых областях ARM по умолчанию.

Когда для хранения данных используется внешняя память и она не отображается в кэшируемой области, либо используйте переназначение, когда возможно перемещение в кэшируемую область, что имеет место для банков SDRAM (SWP_FMC [1: 0] = 01 в регистре SYSCFG_MEMRMP) или используйте MPU для изменения типа памяти на память обычного типа.

Производительность доступа к внешней памяти зависит от таймингов памяти, а также от ширины ее шины данных. Чем меньше время доступа к внешней памяти, тем выше производительность. Чем больше ширина шины данных, тем выше производительность.

Когда ЦП в приложение загружает огромные константы (что является случаем БПФ), пользователь должен проверить, находятся ли они в правильном месте памяти и предназначены ли они для размещения не во внутренней флэш-памяти. В некоторых случаях, если переменные непреднамеренно находятся во внутренней флэш-памяти (TCM или AXI), а ускоритель ART или D-кеш, или оба они не включены, приложение отключает все состояния ожидания Flash, и приложение значительно замедляется.

Когда это возможно, попробуйте разделить ячейки данных и кода, особенно когда их память подключена к АНВ / AXI, чтобы избежать одновременного доступа к матрице шины, то есть данные выбираются через АНВ / AXI и инструкции загружаются через TCM.

Для доступа к Flash и для повышения производительности попробуйте разделить инструкции и доступ к данным только для чтения. Например, доступ к данным только для чтения осуществляется через Flash-TCM, а к инструкциям - через Flash-AXI или наоборот.

Если в качестве местоположения данных используется DTCM-RAM, а используемые переменные являются байтовыми или / или полусловными, поскольку в этом ОЗУ серии STM32F7 нет управления ECC, рекомендуется отключить чтение-изменение-запись DTCM-RAM в интерфейсе DTCM (в регистре DTCMCR) для увеличения производительности.

Для этого добавьте следующий код C в SystemInit () в файле system_stm32f7xx.c или в начале основной функции:

```
__IO uint32_t * CM7_DTCMCR = (uint32_t *) (0xE000EF94);  
* CM7_DTCMCR &= 0xFFFFFFFF;  
/* Отключить чтение-изменение-запись */
```

Или это можно сделать в ассемблерном коде в стартовом файле:

```
CM7_DTCMCR EQU 0xE000EF94  
LDR R2, = CM7_DTCMCR  
LDR R0, [R2]  
BFC R0, # 1, # 1; Сбросить поле CM7_DTCMCR.RMW  
STR R0, [R2]  
DSB  
ISB
```

Не рекомендуется включать кеш перед вызовом основной функции, то есть перед фазой загрузки с разбросом, потому что может произойти серьезная ошибка.

6. Заключение

Сегодня приложения становятся все более сложными и требуют большей эффективности и производительности микроконтроллера. Благодаря интеллектуальной архитектуре STM32F7 устройство серии STM32F7 является подходящей платформой для приложения Интернета вещей (IoT). Это позволяет разработчикам, которым нужен быстрый микроконтроллер, упростить оптимизацию кода.

Лучшая скорость отклика приложений достигается благодаря двум независимым механизмам STM32F7 для достижения производительности с нулевым ожиданием: ART Accelerator™ от ST для внутренней флэш-памяти, L1-кеш (кеши инструкций и данных) для внутренней флэш-памяти и другие запоминающие устройства (внутренние и внешний). Благодаря кеш-памяти пользователь тратит меньше времени на оптимизацию кода и размера данных за счет добавления ресурсов внешней памяти без потери производительности. Даже DMA-передачи выполняются одновременно с процессором, производительность не снижается благодаря системной архитектуре STM32F7.

Для более высокой производительности пользователь может выбрать устройства STM32F76xxx и STM32F77xxx, чтобы воспользоваться преимуществом увеличения размера кэша.

Для недорогих устройств и для приемлемой производительности пользователь может выбрать устройства STM32F72xxx / STM32F73xxx или STM32F74xxx / STM32F75xxx в зависимости от того, предназначено ли графическое приложение.