

AN4838: модуль управления защитой памяти STM32



Добавил(а) microsin

Это перевод апноута **AN4838** [1] (автор перевода Marat Galyamov), который описывает управление модулем защиты памяти (memory protection unit, **MPU**) в архитектуре **STM32**. Блок MPU является опциональным (программно подключаемым) компонентом для защиты памяти. Включение модуля MPU в STM32 делает работу процессора более устойчивой и надежной. MPU должен быть настроен и включен перед использованием. Если модуль MPU не включен, то нет никаких изменений в поведении системы памяти. Этот апноут касается все продуктов STM32, включая Cortex®-M0+/M3/M4 и M7 варианты, которые поддерживают MPU.

Дополнительная информация про MPU

Таблица 1. Процессоры STM32, которые имеют блок MPU.

Тип	Модели процессоров (серии)
Микроконтроллер	STM32F1, STM32F2, STM32F3, STM32F4, STM32F7, STM32L0, STM32L1, STM32L4

[Обзор системы MPU]

MPU может быть использован для повышения устойчивости и надежности системы, предоставляя следующие возможности:

- MPU запрещает пользовательскому приложению изменять данные, используемые критическими задачами (такой, как ядро операционной системы).
- MPU определяет область памяти SRAM как неисполняемую для предотвращения атаки внедренного кода.
- MPU изменяет атрибуты доступа к памяти.

MPU может быть использован для защиты 8 областей памяти. В свою очередь, они могут иметь 8 подобластей, если размер области не менее 256 байт. Подобласти всегда имеют равный размер, и могут быть включены или отключены с использованием номера подобласти. Поскольку минимальный размер области зависит от длины строки кэша (32 байта), 8 подобластей по 32 байта соответствует 256 байтам.

Области пронумерованы от 0 до 7. В дополнение есть другая область, названная областью по умолчанию, с идентификатором -1. Все области памяти в диапазоне от 0 до 7 имеют приоритет над областью по умолчанию.

Области могут перекрываться, и могут быть вложенными. Область 7 имеет более высокий приоритет, область 0 - низкий, и это определяет поведение перекрывающихся областей. Приоритеты фиксированы и не могут быть изменены.

Рис. 1 показывает пример с 6-ю областями. Этот пример показывает, что область 4 перекрывает области 0 и 1. Область 5 включена полностью в область 3. Так как приоритет идет в порядке возрастания, у областей перекрытия (желтые) есть приоритет над другими областями. Таким образом, если область 0 определена как доступная для записи, а область 4 наоборот, недоступная, то адрес, попадающий в зону перекрытия областей, не будет доступен для записи.



Рис. 1. Пример перекрытия областей.

MPU един для процессора, что означает отсутствие разделения областей для данных и команд.

MPU может быть использован для определения других атрибутов памяти, таких как кэширование, которые могут быть экспортированы (применены) в системном блоке кэш-памяти или в контроллере памяти.

Управление кэшем происходит глобально посредством регистра команд кэша, но MPU может определить политики кэша, и будет ли область кэшируемой или нет. MPU позволяет установить атрибуты кэша для области кэшем уровня L1 (доступно только для серии STM32F7, у которой есть кэш L1).

Модель памяти. В архитектуре STM32 процессор имеет по умолчанию фиксированную карту распределения памяти, которая предоставляет до 4 гигабайт адресуемой памяти. На рис. 2 показана карта распределения памяти.

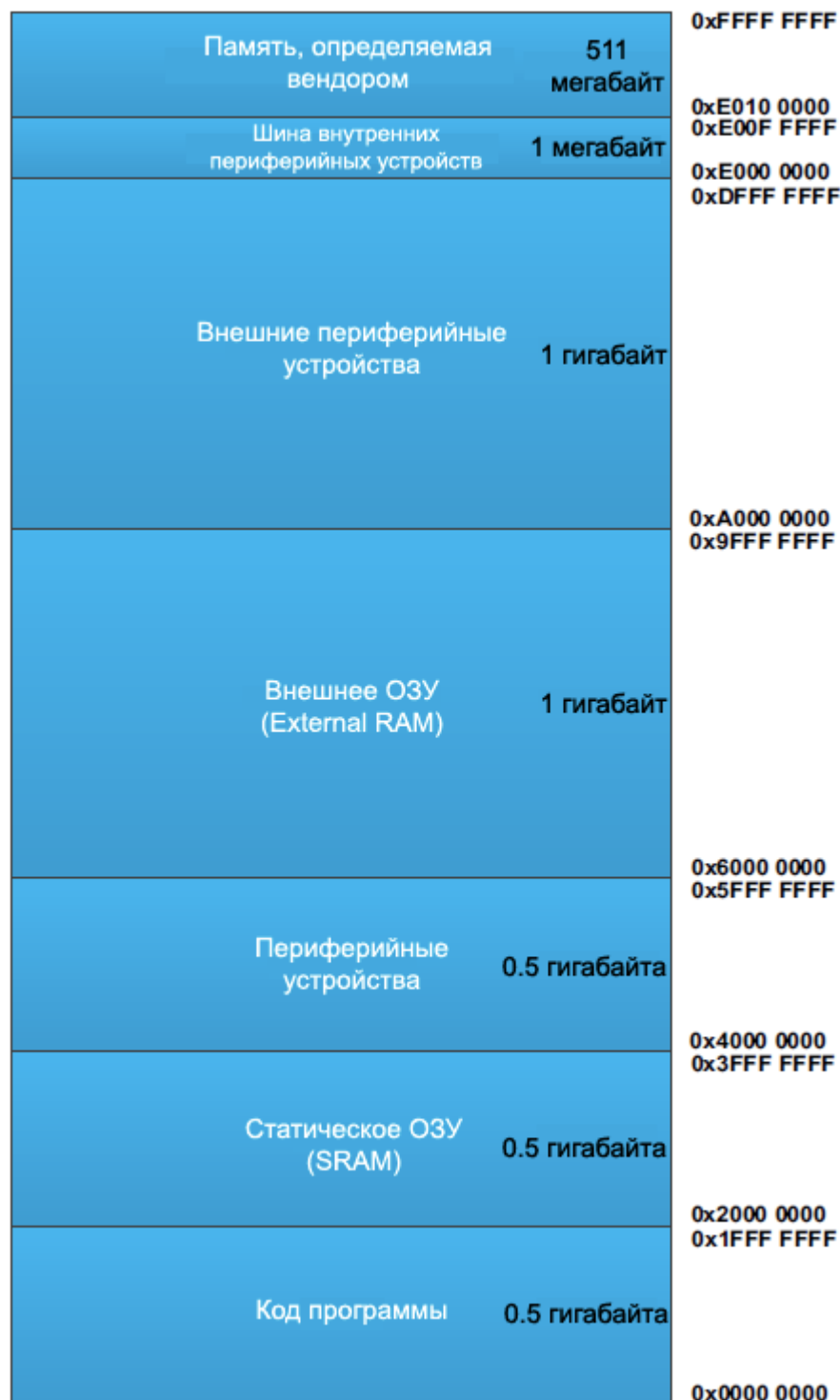


Рис. 2. Распределение адресного пространства процессора.

[Типы памяти, регистры и атрибуты]

Карта распределения памяти и настройка MPU разделяют карту на области. Каждая область имеет определенный тип памяти и её атрибуты.

Типы памяти. Есть три общих типа памяти:

- **Память Normal:** позволяет загружать и хранить байты, полуслова и слова, которые будут эффективно расположены процессором (компилятору неизвестно о типах областей памяти). Для этой области загрузка/хранение не обязательно выполняется для процессора в том порядке, который определен в исходном коде программы.
- **Память Device:** в этом регионе загрузка и хранение совершаются в строго упорядоченном виде. Это должно

гарантировать, что последовательность загрузки регистров выполнена в надлежащем порядке.

- Строго упорядоченная память: всегда выполняется в порядке, написанном в программе, ожидает завершения инструкции записи/хранения (эффективный доступ к шине) прежде, чем выполнить следующую инструкцию в потоке программы. Это может серьезно увеличить производительность.

Регистры MPU. Регистры MPU расположены по адресу 0xE00ED90. Есть 5 основных регистров MPU и несколько регистров-псевдонимов (alias registers) для каждой области. Следующие регистры используются для установки областей в MPU:

MPU_TYPE регистр только для чтения, используется для детекции наличия MPU.

MPU_CTRL регистр управления.

MPU_RNR номер области, используется для определения допустимых операций к ней.

MPU_RBAR базовый адрес области.

MPU_RASR атрибуты области и размер.

MPU_RBAR_An n-й псевдоним MPU_RBAR, где n в диапазоне от 1 до 3^(a).

MPU_RASR_An n-й псевдоним MPU_RASR, где n в диапазоне от 1 до 3^(a).

Примечание (a): В Cortex®-M0+ эти регистры не реализованы.

Для получения подробной информации про регистры MPU обратитесь к руководствам, приведенным во врезке "Дополнительная информация про MPU".

Атрибуты памяти. В регистре атрибутов и размера (MPU_RASR) устанавливаются все атрибуты памяти. Таблица 2 показывает краткое описание регистра атрибутов и размера.

Таблица 2. Регистр атрибутов и размера области - MPU_RASR.

№ бита	Мнемоника	Описание
28	XN	Execute never, никогда не выполнять.
26:24	AP	Data Access Permission, разрешенный тип доступа к региону (RO, RW или нет доступа).
21:19	TEX	Type Extension, расширение типа.
18	S	Shareable, возможность совместного использования.
17	C	Cacheable, возможность кэширования.
16	B	Bufferable, возможность буферизации.
15:8	SRD	Subregion disable, запрет подрегиона. Для каждого подрегиона 1=запрещено, 0=разрешено.
5:1	SIZE	Указывает размер региона, защищаемого MPU.

XN. Этот флаг XN управляет выполнением кода. В случае выполнения команд в области должен быть разрешен доступ на чтение на уровне привилегий и XN должен быть 0. Иначе будет сгенерирована ошибка управления памятью (MemManage).

AP. Это поле разрешения доступа к данным (см. таблицу 3).

Таблица 3. Права доступа к регионам памяти.

AP[2:0]	Привилегированные разрешения	Не привилегированные разрешения	Описание
000	Нет доступа	Нет доступа	Любая попытка доступа вызывает ошибку нарушения защиты доступа (permission fault).
001	RW (полный доступ)	Нет доступа	Разрешен доступ только для привилегированного программного обеспечения.

010	RW (полный доступ)	RO (только чтение)	Попытка записи непривилегированным программным обеспечением приведет к ошибке защиты доступа (permission fault).
011	RW (полный доступ)	RW (полный доступ)	Разрешен полный доступ.
100	Не предсказуемый результат		Зарезервировано.
101	RO (только чтение)	Нет доступа	Разрешено только чтение, и только привилегированным программным обеспечением.
110	RO (только чтение)	RO (только чтение)	Разрешено только чтение.
111			

S. Поле S для области совместно используемой памяти: система обеспечивает синхронизацию данных между устройствами управления шиной в системе с многократными устройствами управления шиной, например, процессор с контроллером DMA.

TEX, C, B. Биты TEX, C и B используются для определения свойств кэша области и возможности его совместного использования. Варианты представлены в таблице 4.

Таблица 4. Свойства кэша и его совместное использование.

TEX	C	B	Тип памяти	Описание	Совместное использование
000	0	0	Strongly Ordered	Строго упорядоченная память.	Да
000	0	1	Device	Совместно используемое устройство.	Да
000	1	0	Normal	Сквозная запись (Write through), нет выделения на запись.	Бит S
000	1	1	Normal	Обратная запись (Write-back), нет выделения на запись.	Бит S
001	0	0	Normal	Нет кэширования.	Бит S
001	0	1	Зарезервировано		
001	1	0	Не определено		
001	1	1	Normal	Обратная запись, обычная запись и выделение на чтение.	Бит S
010	0	0	Device	Устройство, которое нельзя использовать совместно.	Нет
010	0	1	Зарезервировано		

SRD. Это биты отключения подобласти, которые определяют, включена ли определенная подобласть или нет. Отключение подобласти означает, что другая область, перекрывающая отключенный диапазон, соответствует ей вместо этого. Если никакая другая включенная область не перекрывает отключенную подобласть, MPU генерирует ошибку.

Для моделей с реализованным кэшем (только для серии STM32F7, у которой есть кэш L1) существуют дополнительные атрибуты памяти:

- Кэшированный или не кэшированный: означает, что выделенные области могут быть сохранены в кэше или нет.
- Прямая запись без выделения записи: при успехе записывает в кэш и основную память, при пропуске обновляет блок в основной памяти, не переносящей этот блок в кэш.
- Обратная запись без выделения записи: при успехе пишет в кэш "грязный бит" для блока, основная память не обновляется. При пропуске обновляет блок в основной памяти, не переносящей этот блок в кэш.
- Обратная запись с выделением чтения и записи: при успехе пишет в кэш, с установкой "грязного бита" для блока, основная память не обновляется. При пропуске обновляет блок в основной памяти и переносит блок в кэш.

[Сравнение характеристик MPU у различных моделей процессоров]

Есть несколько различий между уровнями MPU Cortex®-M0+, Cortex®-M3/M4 и Cortex®-M7, поэтому программист должен знать о них, если в программном обеспечении будет использоваться конфигурирование MPU. Таблица 5 показывает различия характеристик MPU между Cortex®-M0+, Cortex®-M3/M4 и Cortex®-M7.

Таблица 5. Сравнение характеристик MPU STM32.

Параметр	Модели процессоров (серии)		
	Cortex®-M0+	Cortex®-M3/M4	Cortex®-M7
Количество регионов памяти		8	
Унифицированные регионы I и D		Да	
Адрес региона		Да	
Размер региона		256 байт .. 4 гигабайта	
Атрибуты памяти региона	S, C, B, XN ⁽¹⁾	TEX, S, C, B, XN	
Установка разрешений на доступ к региону памяти (AP)		Да	
Запрет подрегиона		8 бит	
Пропуск MPU для NMI (не маскируемое прерывание) / Hardfault (аппаратная ошибка)		Да	
Псевдонимы для регистров MPU	Нет		Да
Генерация исключения при отказе (fault exception)		Только Hardfault	Hardfault / MemManage

Примечание (1). Cortex®-M0+ поддерживает один уровень политики кэша, поэтому поле TEX недоступно в cortex®-M0+.

Пример настройки MPU с помощью библиотеки CubeMX HAL

[Заключение]

Использование MPU в микроконтроллерах STM32 делает их устойчивыми, надежными, и в некоторых случаях более безопасными, препятствуя задачам приложения получить нежелательный доступ к стеку или области памяти, или повредить стек и память данных, используемые другими задачами.

[Ссылки]

1. [AN4838 Managing memory protection unit \(MPU\) in STM32 MCUs site:www.st.com.](#)
2. [AN4839: кэш первого уровня на STM32F7.](#)