**3.1 Введение.**

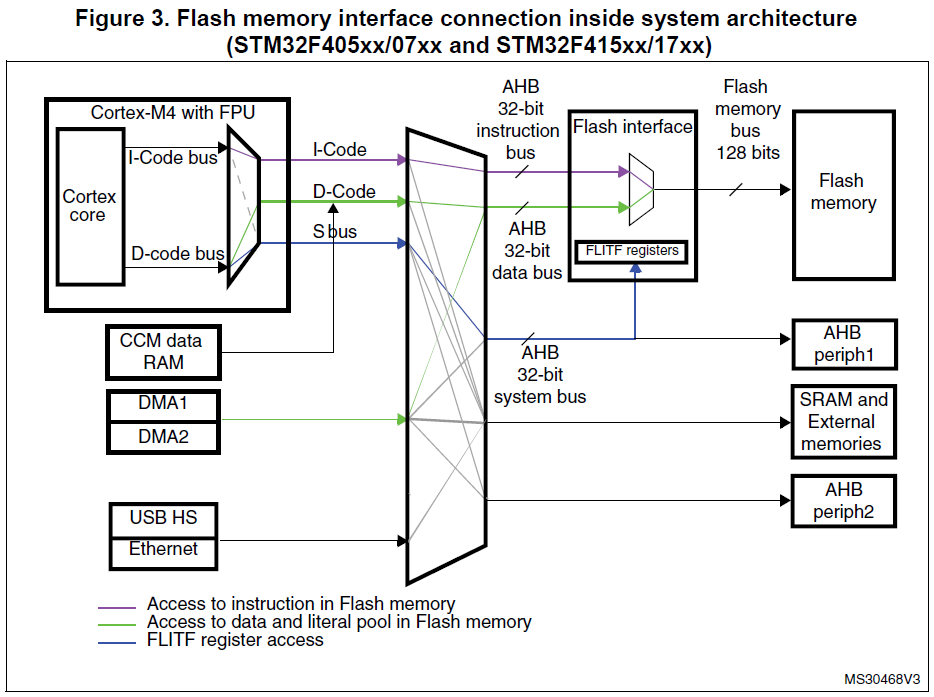
Интерфейс Flash памяти управляет доступами процессора по шинам ICode и DCode к Flash памяти. Он реализует операции стирания и программирования Flash памяти, а также механизмы защиты от чтения и записи.

Интерфейс Flash памяти ускоряет выполнение кода с помощью системы предвыборки инструкции и линий кэша.

**3.2 Основные особенности.**

* Операции чтения Flash памяти.
* Операции программирования стирания Flash памяти.
* Защита от чтения/записи.
* Предвыборка по шине I-Code.
* 64 кэш-линий 128 битов на I-Code.
* 8 кэш-линий 128 битов на D-Code.

Рисунок 3 показывает соединение интерфейса Flash памяти внутри системной архитектуры.



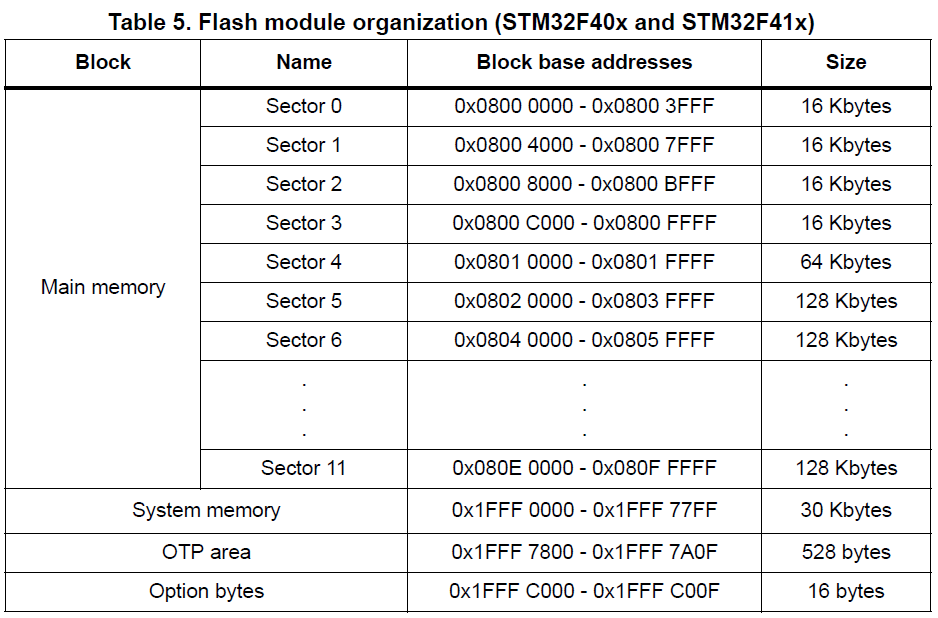
**3.3 Встроенная Flash память в STM32F405xx/07xx и STM32F415xx/17xx.**

Flash память имеет следующие главные особенности:

* Ёмкость до 1 Мбайта.
* Чтение 128 разрядных данных.
* Запись байта, полуслова, слова и двойного слова.
* Стирание сектора/массовое стирание.
* Организация памяти.

Flash память организована следующим образом:

* Главный блок памяти разделён на 4 сектора = 16 кб, 1 сектор = 64 кб, и 7 секторов по 128 кб.
* Системная память, откуда стартует устройство в режиме загрузки с системной памяти.
* 512 OTP (one-time programmable) байтов пользовательских данных. OTP область содержит 16 дополнительных байтов, используемых для блокировки соответствующего блока данных OTP.
* Байты опций для настройки защиты от чтения/записи, уровень BOR, аппаратный/программный сторожевой таймер и сброс когда устройство находится в режиме Standby или Stop.
* Энергосберегающие режимы.



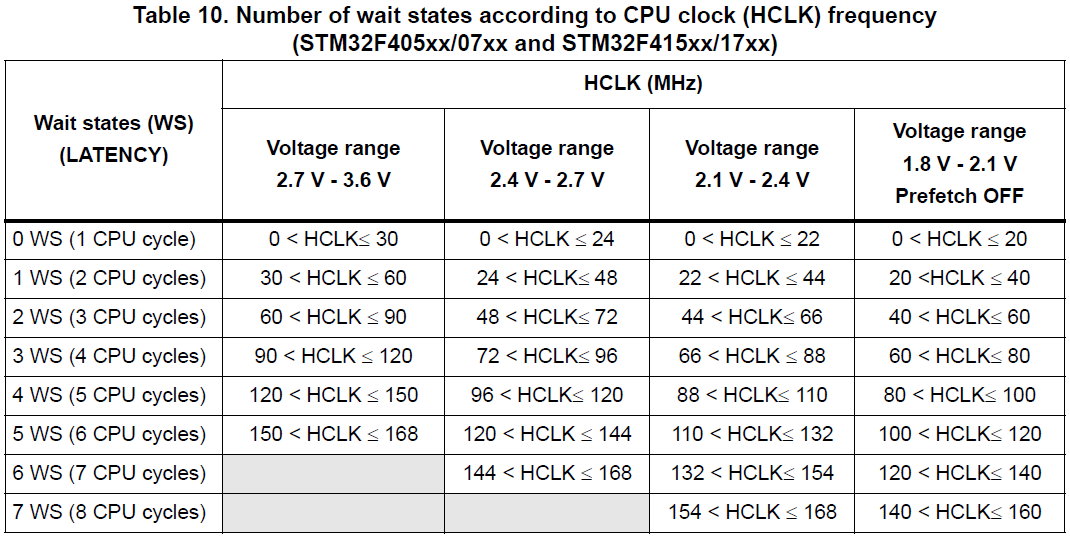
**3.5 Интерфейс чтения.**

**3.5.1 Соотношение между тактовой частотой процессора и временем чтения Flash памяти.**

Для корректного чтения данных из Flash памяти, в регистр контроля доступа к Flash (FLASH\_ACR) должно быть корректно запрограммировано некоторое количество wait states в соответствии с частотой процессора (HCLK) и напряжением источника питания устройства. Буфер предвыборки должен быть отключен, когда напряжение питания падает ниже 2.1В. Соответствие между wait states и тактовой частотой процессора приведено в таблице 10 и 11.

Примечание:

На устройствах STM32F405xx/07xx и STM32F415xx/17xx: там где VOS=’0’, максимальное значение Fhclk = 144 МГц, там где VOS=’1’, максимальное значение Fhclk = 168 МГц.



После сброса, процессор тактируется частотой 16 МГц и регистр FLASH\_ACR настроен на 0 wait states. Настоятельно рекомендуется использовать следующую программную последовательность для настройки количества wait states, необходимую для доступа к Flash памяти на частоте процессора.

Увеличение тактовой частоты процессора.

1. Запрограммировать количество wait states в биты LATENCY в регистре FLASH\_ACR.
2. Проверить, что новое количество wait states принято во внимание для получения доступа к Flash памяти путем чтения регистра FLASH\_ACR.
3. Изменить источник синхронизации процессора записью битов SW в регистр RCC\_CFGR.
4. Если необходимо, изменить прескалер источника синхронизации процессора записью битов HPRE в регистр RCC\_CFGR.
5. Проверить, что новый источник тактирования процессора и новое значение прескалера взято во внимание, для этого необходимо прочитать статусные биты SWS и HPRE, соответственно, в регистре RCC\_CFGR.

Уменьшение тактовой частоты процессора.

1. Изменить источник синхронизации процессора записью битов SW в регистр RCC\_CFGR.
2. Если необходимо, изменить прескалер записью битов HPRE в регистр RCC\_CFGR.
3. Проверить, что новый тактовый источник или/и новое значение прескалера принято во внимание. Для этого прочитать биты SWS или/и HPRE, соответственно, в регистр RCC\_CFGR.
4. Запрограммировать новое количество wait states в биты LATENCY в регистре FLASH\_ACR.
5. Проверить, что новое количество wait states используется для получения доступа к Flash памяти чтением регистра FLASH\_ACR.

Примечание: изменение конфигурации частоты процессора или конфигурации wait states (WS) может не быть эффективным немедленно. Чтобы убедиться, что текущая частота процессора соответствует настроенной, вы должны проверить прескалер AHB и статусные значения тактового источника. Чтобы убедиться, что количество WS соответствует запрограммированному значению, необходимо прочитать регистр FLASH\_ACR.

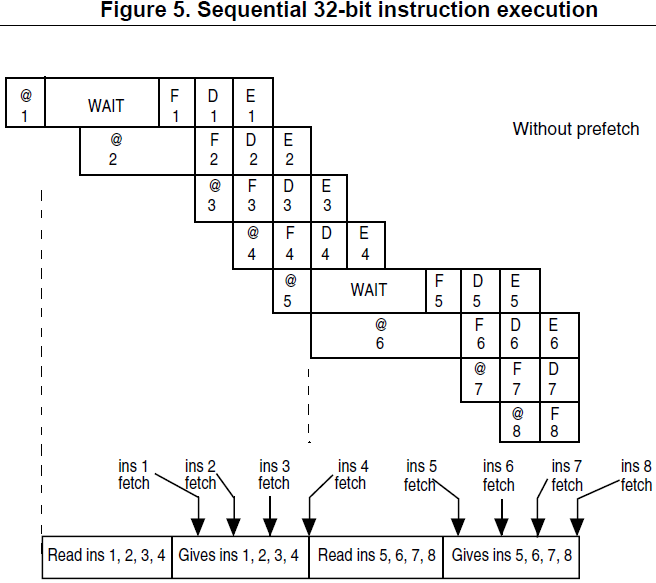
**3.5.2 Адаптивный акселератор памяти реального времени (ART Accelerator).**

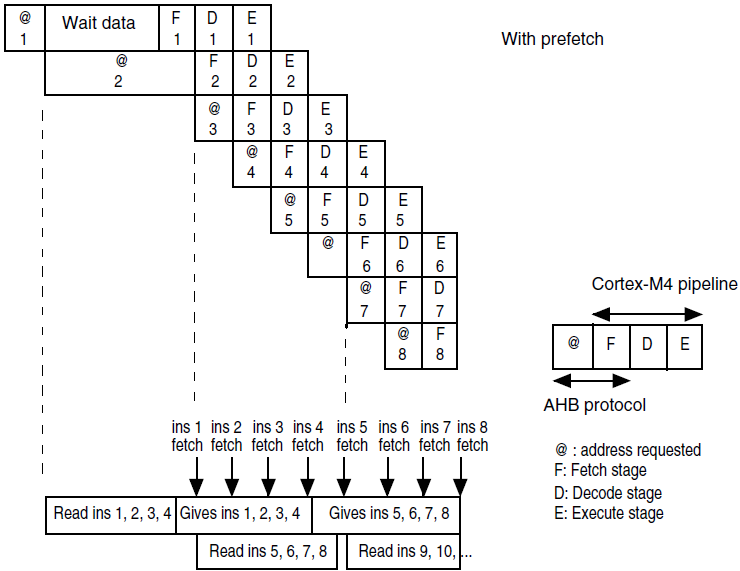
Проприетарный ART акселератор памяти оптимизирован под STM32 промышленный стандарт ARM Cortex-M4 с ядром FPU поверх технологии Flash памяти, которая обычно требует, чтобы процессор ожидал готовность Flash на высоких тактовых частотах.

Для реализации полного потенциала процессора, акселератор реализует очередь предвыборки инструкций и кэш ветвлений, который увеличивает скорость исполнения программы из 128-битной Flash памяти. На основе тестов производительности CoreMark установлено, что производительность, достигнутая благодаря ART акселератору эквивалентна 0 wait state при выполнении программы из Flash памяти на частоте CPU 180 МГц.

Предвыборка инструкций.

Каждая операция чтения из Flash памяти предоставляет 128 битов либо 4 инструкций по 32 бита, или 8 инструкций по 16 бит, в соответствии с запущенной программой. Так, в случае последовательного исполнения кода, как минимум 4 тактовых цикла необходимо для выполнения предыдущей прочитанной строки инструкций. Предвыборка на шине I-Code может быть использована для чтения следующей последовательной строки из Flash памяти, пока текущая строка выполняется процессором. Предвыборка разрешается установкой бита PRFTEN в регистре FLASH\_ACR. Эта возможность полезна, если как минимум 1 WS необходимо для доступа к Flash памяти. Рисунок 5 показывает выполнение последовательных 32 битных инструкций с и без предвыборки, когда 3 WSs необходимо для доступа к памяти.





Пояснение к рисунку (от Автора). Если предвыборка отключена, то после выставления адреса на шину «@» необходимо ждать 3 WS прежде, чем процессор сможет прочитать строку из Flash. После чтения запускается конвейер, и все 4 инструкции выполняются (стадии F, D, E – выборка, декодирование, исполнение). Затем выставляется адрес следующей строки, снова ждём 3 WS, затем запускается конвейер. Не понятно почему адрес второй инструкции выставлен раньше чем нужно. Если предвыборка разрешена, то следующая строка считывается из Flash когда выполняется предыдущая. Таким образом, к концу выполнения инструкций 1-4, инструкции 5-6 начнут выполняться без задержек (WS).

Если код ветвится, инструкции может не оказаться в текущей исполняемой строке. В этом случае (осечка) имеет место пенальти в терминах количества циклов, как минимум равным количеству WS.

**Память кэша инструкций.**

Чтобы ограничить потерянное время из-за прыжков, можно сохранить 64 строки по 128 битов в кэше памяти инструкций. Эта возможность может быть разрешена установкой бита разрешения кэша (ICEN) в регистре FLASH\_ACR. Всякий раз, когда происходит осечка (требуемые данные не присутствуют в текущей строке инструкций, в пред выбранной строке или в кэше памяти инструкций), строка копируется в кэш памяти инструкций. Если некоторые данные содержащиеся в кэше запрашиваются процессором, они предоставляются им без какой либо задержки. Как только будет заполнен весь кэш инструкций, LRU (least recently used) policy используется для определения строки, подлежащей замещению в кэше памяти. Эта особенность особенно полезна в случае кода, содержащего циклы.

**Управление данными.**

Наборы констант извлекаются из Flash памяти через D-Code шину во время стадии исполнения конвейера процессора. Следовательно, конвейер останавливается до тех пор, пока не будет предоставлен запрошенный набор констант. Чтобы ограничить временные потери из за этого факта, доступы шины данных D-Code к памяти имеет приоритет над доступами по шине инструкций I-Code.

**NB**: Именно по этому, в случае частого обращения к flash за данными целесообразно выполнять код из RAM (атрибут \_\_ram\_func). В этом случае – реализуется принцип гарвардской архитектуры, только наоборот – инструкции считываются из ram, а данные считываются из rom.

Если некоторые наборы констант используются часто, кэш памяти данных может быть разрешен установкой бита DCEN в регистре FLASH\_ACR. Это работает подобно кэшу памяти инструкций, но сохраняемый размер данных ограничен 8 строками по 128 битов.

Примечание: данные в секторе пользовательской конфигурации не кэшируются.

**3.6 Операции стирания и программирования.**

Для любой операции программирования Flash памяти (стирание или запись), тактовая частота процессора (HCLK) должна быть по крайней мере 1 МГц. Содержимое flash памяти может не гарантируется, если происходит сброс устройства во время операции с Flash памятью.

Любая попытка чтения Flash памяти на STM32F4xx в процессе записи или стирания приведет к зависанию шины. Операция чтения корректно выполнится, как только завершится операция программирования. Это означает, что выборка кода или данных не может осуществляться пока выполняется операция записи/стирания.

**3.6.1 Разблокировка регистра управления Flash.**

После сброса, запись не разрешена в регистр управления Flash (FLASH\_CR) для защиты Flash памяти от возможных нежелательных операций, возникающих, например, из-за электрических искажений. Для разблокировки этого регистра используется следующая последовательность:

1. Запись KEY1 = 0x45670123 в регистр ключа Flash (FLASH\_KEYR).
2. Запись KEY2 = 0xCDEF89AB в регистр ключа Flash (FLASH\_KEYR).

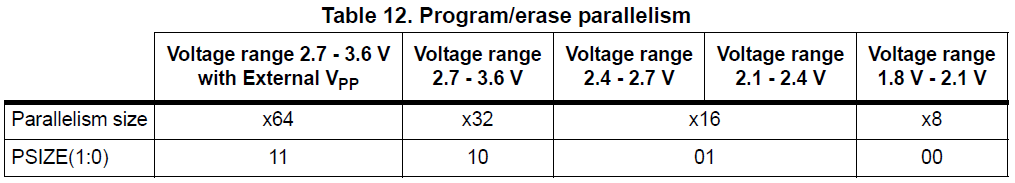
Любая неправильная запись вернёт ошибку шины и заблокирует регистр FLASH\_CR для следующего сброса. FLASH\_CR может быть заблокирован снова программно установкой бита LOCK в регистре FLASH\_CR.

Примечание: регистр FLASH\_CR не доступен на запись, если бит BSY в регистре FLASH\_CR установлен. Любая попытка записи в него с установленным битом BSY приведет к зависанию AHB шины до тех пор, пока он не сбросится.

**3.6.2 Параллелизм программирования/чтения.**

Размер параллелизма устанавливается в поле PSIZE в регистре FLASH\_CR. Он представляет количество байтов, программируемых при каждой операции записи в Flash память. PSIZE ограничивается напряжением источника питания и зависит от того, используется ли внешний источник Vpp или нет. Таким образом, он должен быть корректно настроен в регистре FLASH\_CR перед любой операцией программирования/записи.

Операция стирания Flash памяти может выполнена для сектора, банка или для всей Flash памяти (массовое стирание). Время стирания зависит от запрограммированного значения PSIZE. Таблица 12 предоставляет корректные значения PSIZE.



Примечание. Любая операция программирования или стирания начинающаяся с противоречащих настроек программного параллелизма/диапазона напряжений может привести к непредсказуемому результату. Даже если последующая операция чтения индицирует, что логическое значение было эффективно записано в память, это значение может не сохраниться.

Прим. Автора – Vpp не исп. Для программирования. Питание микроконтроллера 3.3В. Следовательно пишем 32-битные слова.