

8 ТИПОВЫЕ СТРУКТУРНО-АРХИТЕКТУРНЫЕ РЕШЕНИЯ ПОДСИСТЕМЫ ПРЯМОГО ДОСТУПА К ПАМЯТИ МК ОБЩЕГО НАЗНАЧЕНИЯ

8.1 Общие вопросы.....	2
8.2 Принципы реализации обмена данными в режиме ПДП	10
8.3 Типовые структурно-архитектурные решения подсистемы ПДП МК общего назначения.....	14
8.3.1. Базовые структурно-архитектурные решения и принципы работы подсистем ПДП МК семейства <i>ARM Cortex-Mx</i>	14
8.3.2. Особенности структуры и архитектуры подсистемы ПДП МК подсемейств <i>ARM Cortex-M3</i> и <i>ARM Cortex-M4</i>	29
8.4 Примеры программирования подсистемы ПДП.....	54
8.5 Выводы по разделу 8.....	67

8.1 Общие вопросы

8.1.1. Наличие подсистемы прямого доступа к памяти (ПДП, *DMA*) характерно, в основном, для семейств МК класса «*high performance*»; в большинстве семейств классов «*mainstream*» и «*cost-sensitive*» она отсутствует (см., в частности, табл. 1.1). Назначение данной подсистемы – управление обменом **массивами данных** между памятью и ПУ, между 2-мя ПУ или между 2-мя областями памяти в режиме **прямого доступа к памяти** (*Direct Memory Access*). Использование ПДП при обмене одиночными словами **не рационально** [3, 9].

8.1.2. Режим ПДП характеризуется следующими основными **отличительными особенностями** [3]:

- сеанс обмена данными может быть инициирован как ПУ, так и ЦП;
- разрешение на сеанс выдает ЦП;
- собственно процедура обмена данными реализуется без участия ЦП, под управлением **контроллера ПДП**, который до сеанса обмена должен быть сконфигурирован ПО МК (см. подраздел 8.3);
- все необходимые в процессе обмена операции (в том числе формирование и модификация адресов источника и приемника, сигналов управления записью и чтением и т. п.) реализуются контроллером ПДП на **аппаратном** уровне;
- по окончании сеанса обмена данными контроллер ПДП выставляет в активное состояние признак завершения сеанса, располагаемый в регистре статуса контроллера; как правило, при этом также вырабатывается запрос на прерывание по событию «Завершение передачи».

8.1.3. К **преимуществам** режима ПДП по сравнению с режимом обмена по прерыванию и с обменом, инициируемым ПО МК (см. подраздел 7.1) обычно относят [3]:

- более эффективное использование ЦП за счет возможности выполнения им других задач во время сеанса передачи данных;
- более высокую скорость передачи массивов данных, за счет неучастия ЦП в данном процессе.

Первое из вышеперечисленных преимуществ является бесспорным. Что же касается выигрыша в **скорости** обмена данными, его наличие или отсутствие, а также его степень зависят от архитектуры подсистемы ПДП и вычислительной системы в целом. В частности, в МК семейства *ARM Cortex-Mx* при использовании обмена данными в режиме ПДП выигрыш в скорости может быть незначительным или отсутствовать. Продемонстрируем это на несложном **примере**.

Пусть необходимо записать в ПД массив данных, поступающих из некоторого ПУ МК (например, массив отсчетов АЦП). Оценим затраты времени на выполнение данной процедуры при ее реализации на МК популярного у разработчиков модельного ряда *STM32F10xx* [13] подсемейства *ARM Cortex-M3* в режимах обмена данными:

- иницируемого ПО;
- по прерыванию;
- ПДП.

8.1.4. В режиме обмена, **иницируемого ПО**, в каждом цикле чтения очередного слова данных из ПУ и его записи в память ЦП должен, как минимум, выполнить следующие операции:

- 1) ввод содержимого регистра состояния ПУ в некоторый РОН ЦП;
- 2) проверку признака готовности ПУ к передаче очередного слова в ПД (например, признака готовности очередного результата преобразования АЦП) с переходом к пункту 3) при установке признака и к пункту 1) – в противном случае;
- 3) считывание очередного слова из регистра данных ПУ в некоторый РОН ЦП;
- 4) передачу считанного слова из РОН в ПД по определенному адресу с его последующим автоматическим увеличением или уменьшением (в зависимости от направления заполнения памяти), см. пункт 2.6.3;
- 5) проверку, выполнена ли запись в ПД всех элементов массива; если выполнена – выход из процедуры обмена данными; если нет – переход к пункту 1).

В соответствии с *Cortex-M3 Technical Reference Manual*, при однократном выполнении операций 1) и 2) и реализации вышеприведенной последовательности операций на МК модельного ряда *STM32F10xx*, минимальные временные затраты на нее составят **12 периодов** тактовой частоты *AHB*-домена.

8.1.5. В режиме обмена по прерываниям от ПУ пункты 1) и 2) вышеприведенной последовательности операций заменяются ожиданием запроса на прерывание от ПУ по готовности к передаче очередного слова данных, с переходом к подпрограмме обслуживания запроса по его поступлении. Пункты 3) и 4) при этом включаются в данную подпрограмму. Необходимо также учесть затраты времени на переход к подпрограмме обслуживания прерывания и на выход из нее; в соответствии с *Cortex-M3 Technical Reference Manual*, они составляют по 12 периодов тактовой частоты *AHB*-домена. Тогда суммарные затраты времени на один цикл чтения очередного слова данных из ПУ и его записи в ПД при использовании обмена по прерыванию и его реализации на МК модельного ряда *STM32F10xx* составят примерно **32 периода** тактовой частоты *AHB*-домена.

8.1.6. Оценим теперь временные затраты на один цикл обмена данными между ПУ и ПД в **режиме ПДП**. В соответствии с [46], в МК модельного ряда *STM32F10xx*, как и в большинстве МК подсемейства *ARM Cortex-M3* (а также *ARM Cortex-M0*), в каждом цикле ПДП-обмена данными передается только одно слово; передача нескольких слов за цикл (**пакетная передача**) возможна в МК от модельного ряда *STM32F20x* и выше [9]. При этом суммарные затраты времени на чтение очередного слова данных из ПУ и его запись в ПД (при условии, что шина доступа к ПД и шина *AHB*, см. рис. 1.7, используются **только** контроллером ПДП, но не ЦП) складываются из затрат времени на следующие операции [46]:

- обработку контроллером ПДП очередного запроса от ПУ (в частности, АЦП вырабатывает его по готовности очередного отсчета); она включает в себя арбитраж запроса (т. е. определение возможности его обслуживания) и вычисление адреса ПД; ее длительность равна минимум 2-м периодам тактовой частоты *AHB*-домена (см. подраздел 1.3);

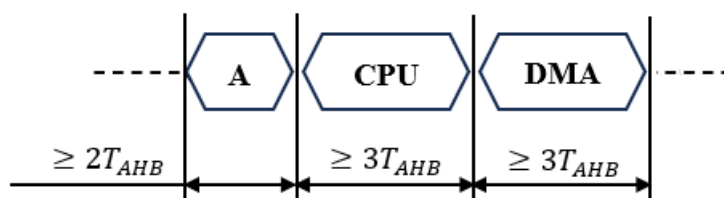
- чтение очередного слова данных из ПУ (минимум 2 периода тактовой частоты домена, к которому принадлежит ПУ);
- запись считанного слова в ПД (1 или 2 периода тактовой частоты *АНВ*-домена);
- передачу контроллером ПДП сообщения о завершении цикла обмена данными (1 период тактовой частоты *АНВ*-домена);
- паузу между циклами обмена данными (1 период тактовой частоты *АНВ*-домена).

Необходимо отметить, что перечисленные операции выполняются **контроллером ПДП**, без участия ЦП, и повторяются по поступлении каждого очередного запроса от ПУ, до тех пор, пока не будет выполнено количество циклов обмена данными, заданное при конфигурировании контроллера ПДП (например, считывание заданного количества отсчетов АЦП).

Таким образом, **минимальные** суммарные затраты времени МК модельного ряда *STM32F10xx* на чтение очередного слова данных из ПУ и его запись в ПД в режиме ПДП, при условиях, что:

- ПД и *АНВ* используются **только** контроллером ПДП, а ЦП к ним не обращается);
 - частота тактирования домена, к которому принадлежит ПУ, равна частоте тактирования *АНВ*-домена;
- составят **7 периодов** тактовой частоты *АНВ*-домена.

В случае, если *АНВ* и / или ПД используются не только контроллером ПДП, но и ЦП (т. е. одновременно с сеансом обмена данными в режиме ПДП выполняется какой-либо фрагмент ПО МК, что обычно имеет место на практике), длительность цикла обмена данными между ПУ и ПД увеличивается. Архитектура подсемейства *ARM Cortex-M3* обеспечивает равномерное распределение времени доступа к *АНВ* и ПД между ЦП и контроллером ПДП, что иллюстрирует рис. 8.1.



A – интервал обработки запроса на ПДП
 CPU и DMA – интервалы времени, в течение которых доступ к ПД и AHB предоставлен ЦП и контроллеру ПДП соответственно
 T_{AHB} – период тактовой частоты AHB -домена

Рис. 8.1. Распределение времени доступа к AHB и ПД между ЦП и контроллером ПДП

Временные диаграммы, приведенные на рис. 8.1, предполагают, что параллельно осуществляются выполнение ЦП некоторого программного фрагмента и обмен данными между ПУ и ПД в режиме ПДП. После обработки очередного запроса на чтение / запись слова данных в режиме ПДП (интервал времени A на рис. 8.1), вначале доступ к ПД и AHB предоставляется ЦП, на интервал времени, необходимый для обмена словом данных между ПУ и ПД, который обозначен на рис. 8.1 как CPU . Минимальная длительность данного интервала – 3 периода тактовой частоты AHB -домена [46]. Затем, на интервал времени с такой же минимальной длительностью, обозначенный на рис. 8.1 как DMA , доступ к ПД и AHB предоставляется контроллеру ПДП. По поступлении следующего запроса на цикл обмена данными в режиме ПДП, представленная рис. 8.1 последовательность интервалов времени повторяется. В промежутках между запросами доступ к ПД и AHB предоставляется ЦП.

Следовательно, при параллельном выполнении некоторого программного фрагмента и обмена данными между ПУ и ПД в режиме ПДП, суммарные затраты времени МК модельного ряда *STM32F10xx* на один цикл обмена данными в режиме ПДП минимум на $3T_{AHB}$ больше, чем при использовании ПД и AHB только контроллером ПДП (см. рис. 8.1), и равны не менее 10-ти периодам тактовой частоты AHB -домена.

8.1.7. Итак, суммарные временные затраты МК модельного ряда *STM32F10xx* на считывание слова данных из ПУ с последующей его записью в ПД составляют:

- в режиме обмена, инициируемого ПО МК – примерно $12T_{AHB}$;
- в режиме обмена по прерыванию - примерно $32T_{AHB}$;
- в режиме ПДП – не менее $7T_{AHB}$ в маловероятной ситуации, когда ПД и *AHB* используются только контроллером ПДП; в более реальной ситуации их совместного использования ЦП и контроллером ПДП – минимум $10T_{AHB}$; на практике данное значение может быть больше, в частности, из-за принадлежности ПУ к *APB*-домену и равенства времени чтения данных из ПУ не $2T_{AHB}$, а $2T_{APB}$.

При использовании МК с уровнем сложности и производительности, сопоставимым с модельным рядом *STM32F10xx* или ниже его, временные затраты будут примерно аналогичными.

Из вышесказанного следует, что использование режима ПДП в относительно несложных МК общего назначения, в целом, не приводит к сколько-нибудь существенному выигрышу в скорости передачи данных по сравнению с обменом, инициируемым ПО (а в ряде частных случаев может даже уступать ему). В то же время режим ПДП обеспечивает несомненный выигрыш в скорости по сравнению с обменом по прерыванию.

Следует отметить, что в МК, контроллеры ПДП которых поддерживают пакетный режим передачи (см. пункт 8.1.6), при использовании режима ПДП может быть достигнут существенный выигрыш в скорости и по сравнению с обменом, инициируемым ПО. То же характерно и для контроллеров ПДП компьютеров, в том числе ПК.

8.1.8. Таким образом, вопреки широко распространенному мнению [3], режим ПДП не всегда обеспечивает выигрыш в скорости передачи данных. В частности, отсутствие существенных преимуществ режима ПДП в скорости по сравнению с обменом данными, инициируемым ПО, характерно для МК подсемейств *ARM Cortex-M0* и большинства МК подсемейства *ARM Cortex-M3*. Тем не менее, во многих практических случаях применение ПДП оправдано

и при отсутствии существенного выигрыша в скорости передачи данных, что обусловлено следующими основными факторами:

- повышение общей производительности МК за счет возможности обмена данными одновременно с выполнением некоторого фрагмента ПО, при несущественном снижении скорости обмена по сравнению с выполнением **только** процедуры передачи данных в режиме ПДП (см. значения затрат времени, приведенные в пункте 8.1.7);

- отсутствие затрат процессорного времени на ожидание готовности ПУ к передаче или к приему данных; готовность определяется контроллером ПДП на аппаратном уровне, по запросам от ПУ, аналогично обмену по прерываниям, но при отсутствии временных затрат на переход к подпрограмме обслуживания запроса и на возврат из нее (см. пункт 8.1.5);

- архитектура подсистем ПДП ряда семейств / подсемейств МК (в частности, *ARM Cortex-M3*) допускает одновременную реализацию 2-х сеансов обмена данными в режиме ПДП между ПД и 2-мя различными ПУ; в данном случае обеспечивается существенный выигрыш в скорости по сравнению и с обменом, инициируемым ПО, и с обменом по прерыванию (подробнее – см. пункт 8.3.1).

Из вышесказанного следует, что, в целом, наличие подсистемы ПДП в структуре МК обеспечивает рациональное решение ряда практических задач обмена данными между его функциональными блоками.

8.1.9. Необходимо, однако, отметить, что сеансу обмена данными в режиме ПДП предшествует относительно сложная подготовительная процедура, включающая в себя: прием центральным процессором запроса на сеанс ПДП, анализ уровня приоритета источника запроса, выделение канала обмена данными, передачу источнику запроса подтверждения о его выделении и т. п. (см. подраздел 8.2). Поэтому преимущества режима ПДП проявляются только, если длительность подготовительной процедуры существенно меньше, чем длительность собственно сеанса обмена данными. Это имеет место при обмене **массивами** данных, причем тем в большей степени, чем больше объем массива.

Таким образом, режим ПДП **рационально применять** для обмена массивами данных. Использование ПДП при обмене одиночными словами данных **не рационально**.

8.1.10. Возможными вариантами обмена данными в режиме ПДП являются [3, 9]:

- ПУ $\rightarrow$ память;
- память $\rightarrow$ ПУ;
- память $\rightarrow$ память;
- ПУ $\rightarrow$ ПУ.

Приведем несколько примеров использования режима ПДП в типовых задачах, решаемых МК в составе систем контроля и управления техническими объектами.

Пример 1 (передача ПУ $\rightarrow$ память). Пусть необходимо определить какие-либо параметры некоторого аналогового сигнала по массиву его отсчетов, полученному посредством встроенного АЦП МК. Формирование массива отсчетов в ПД реализуется их записью в нее (по мере их получения) в режиме ПДП, с последующим расчетом искомых параметров по сформированному массиву отсчетов. См. также пример в пункте 8.4.1.

Пример 2 (передача память $\rightarrow$ ПУ). Пусть необходимо реализовать на МК генератор сигнала некоторой формы (например, гармонического) способом прямого цифрового синтеза (*Direct Digital Synthesis, DDS*). Цифровые отсчеты сигнала хранятся в ПД и в моменты времени, задаваемые одним из таймеров, записываются в регистр данных встроенного ЦАП МК. Передачу отсчетов из ПД в регистр данных ЦАП рационально осуществлять в режиме ПДП. См. также пример в пункте 8.4.2.

Пример 3 (передача память $\rightarrow$ память). Одним из типовых примеров применения такого варианта обмена является скоростная передача массива данных (например, констант) из *Flash*-памяти в ОЗУ МК [9]. Как указано в [9], при использовании режима ПДП может быть достигнут трехкратный выигрыш в скорости по сравнению с передачей под управлением ПО.

Пример 4 (передача ПУ $\rightarrow$ ПУ). Пусть необходимо реализовать обмен потоком данных между 2-мя устройствами, внешними по отношению к МК. Каждое из них подключено к одному из блоков

последовательного периферийного интерфейса (*Serial Peripheral Interface, SPI*) МК (см., например, рис. 1.4). Таким образом, обмен данными сводится к их передаче между блоками *SPI* МК (допустим, *SPI1* и *SPI2*), который рационально осуществлять в режиме ПДП.

Далее рассмотрим общие принципы реализации обмена данными в режиме ПДП.

8.2 Принципы реализации обмена данными в режиме ПДП

Как указано в подразделе 8.1, обмен данными в режиме ПДП осуществляется под управлением контроллера ПДП, являющегося, таким образом, центральным функциональным блоком подсистемы ПДП. Поэтому принципы реализации обмена данными в режиме ПДП будем рассматривать с использованием типовой структуры и архитектуры контроллера ПДП в качестве иллюстративного материала.

Приводимые в технической документации на современные МК [13, 14, 15] структурные схемы контроллеров ПДП недостаточно подробны для описания принципов реализации ПДП на их базе. Учитывая, что данные принципы практически не изменились за ближайшие 30 – 40 лет, в качестве типового примера контроллера ПДП будем использовать БИС *Intel 8237*, применявшуюся в составе первых поколений персональных компьютеров (ПК) *IBM PC*. Структура и архитектура данного контроллера достаточно подробно описаны в доступных источниках информации (см., например, [47]). Естественно, по рассмотрении общих принципов реализации ПДП на примере контроллера *Intel 8237*, далее будут рассмотрены и типовые структурно-архитектурные решения подсистемы ПДП современных МК (см. подраздел 8.3).

Упрощенная структурная схема контроллера ПДП *Intel 8237* и его одного канала представлены на рис. 8.2. Данный контроллер относительно прост по сравнению с аналогичными функциональными блоками современных МК. Однако, его базовые структурно-архитектурные решения и принципы функционирования, в целом, не отличаются от применяемых в

контроллерах ПДП современных МК, за исключением второстепенных деталей (см. подраздел 8.3).

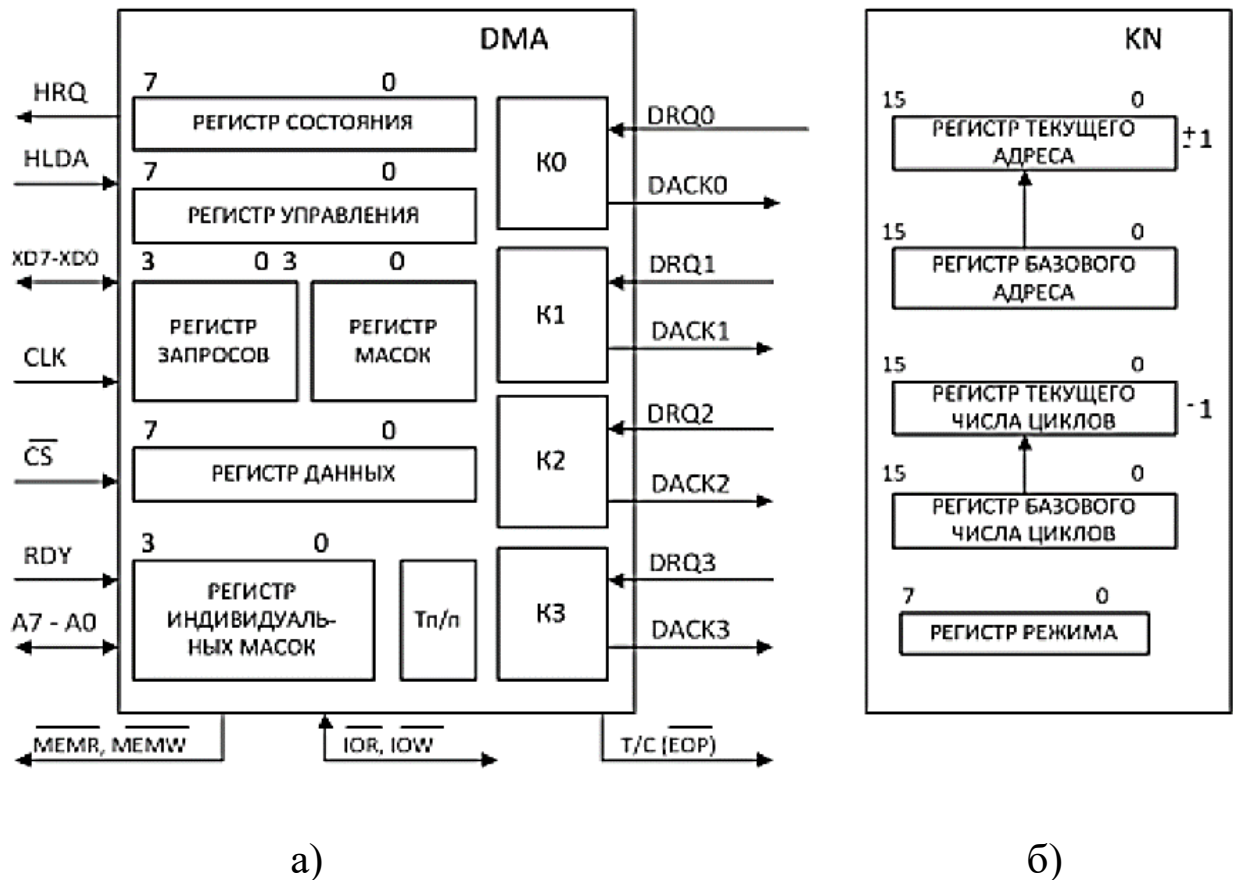


Рис. 8.2. Упрощенная типовая структурная схема контроллера ПДП (а) и его канала KN (б), $N = 0...3$ (на примере БИС *Intel 8237*).

Пояснения приведены в тексте

Типовой контроллер ПДП обеспечивает обмен данными с памятью по одному из нескольких независимых каналов; *Intel 8237* – по одному из 4-х. Его архитектура «по умолчанию» предполагает, что к каждому из каналов подключается одно ПУ, но, в принципе, возможно и обслуживание нескольких ПУ по каналу, что характерно для контроллеров ПДП современных МК (см. подраздел 8.3). При этом под каналом контроллера ПДП подразумевается не только тракт обмена данными, но и аппаратные средства, обеспечивающие обмен по соответствующему тракту (см. рис. 8.2б).

При реализации обмена в режиме ПДП каждый канал перед его использованием должен быть настроен на выполнение процедуры

обмена данными, планируемой для реализации по данному каналу. К **основным параметрам настройки** относятся:

- направление обмена (контроллер *Intel 8237* предоставляет возможность передачи ПУ → память, память → ПУ и память → память);
- базовый адрес, начиная с которого данные записываются в память или считываются из памяти в сеансе связи по соответствующему каналу;
- направление заполнения / считывания памяти (с увеличением или с уменьшением адреса при записи / чтении очередного слова);
- число слов данных, передаваемых во время сеанса связи по каналу.

Например, если по некоторому каналу предполагается ввод в память массива отсчетов АЦП (см. Пример 1):

- в качестве базового адреса по данному каналу должен быть указан адрес, начиная с которого массив должен размещаться в памяти;
- в качестве числа слов данных – количество отсчетов в массиве;
- вариант передачи – ПУ → память;
- заполнение памяти, в принципе, может осуществляться как с увеличением, так и с уменьшением адреса при записи очередного отсчета.

В контроллере *Intel 8237* перечисленные параметры настройки указываются в регистрах базового адреса (РБА), базового числа циклов (РБЧЦ) и в регистре управления каждого из каналов (см. рис. 8.2б). Все данные регистры программно-доступны.

Инициатором сеанса обмена данными по какому-либо из каналов может выступать как ПУ, так и ЦП. На практике чаще встречается первый из названных вариантов. ПУ, инициирующее сеанс обмена, выставляет в активное состояние сигнал запроса *DRQ_x* (где *x* – номер канала). Обслуживание запросов практически во всех существующих подсистемах ПДП осуществляется в соответствии с **приоритетами** источников запросов, аналогично обслуживанию прерываний (см. пункт 7.2.2). В контроллере *Intel 8237* «по умолчанию» используется **одноуровневая статическая** приоритезация; приоритет 0-го канала наивысший, 3-го –

наименьший. Установкой в единицу 4-го бита регистра управления (см. рис. 8.2а) может быть установлен режим **циклической** приоритезации. Во встроенных контроллерах ПДП современных МК используется **многоуровневая квазистатическая** приоритезация (см. подраздел 8.3). Как и запросы на прерывания, запросы на сеансы обмена в режиме ПДП могут быть выборочно **замаскированы** (см. регистры масок на рис. 8.2а).

Если запрос DRQ_x не замаскирован, а уровень его приоритета допускает его обслуживание, контроллер ПДП направляет на ЦП запрос HRQ («запрос шины»). По получении от ЦП разрешения на захват шины ПК ($HLDA$) контроллер передает источнику запроса сигнал разрешения обслуживания ($DACK_x$), и начинается сеанс обмена данными по каналу x , в соответствии с его настройками (например, сеанс записи в память массива отсчетов АЦП, длина которого указана в РБЧЦ, начиная с адреса, указанного в РБА). В течение сеанса управление обменом, в том числе формирование адресов памяти, стробов ее записи и чтения ($\overline{MEMW}$ и $\overline{MEMR}$), а также стробов записи и чтения ПУ ($\overline{IOW}$ и $\overline{IOR}$) осуществляется контроллером ПДП, на аппаратном уровне. В процессе обмена данными младший байт адреса памяти передается по выводам $A7 - A0$ контроллера, а старший байт адреса и данные – по выводам $XD7 - XD0$ (см. рис. 8.2а). Регистр текущего адреса (РТА) отображает адрес памяти, по которому записывается или с которого считывается очередной элемент передаваемого массива. В каждом цикле передачи содержимое РТА инкрементируется или декрементируется (см. рис. 8.2б), в зависимости от направления заполнения памяти, заданного при конфигурировании канала. Содержимое регистра текущего числа циклов (РТЧЦ) декрементируется после каждого цикла передачи; по достижении им значения $0xFFFF$ выставляется признак окончания сеанса обмена данными.

При одновременном поступлении запросов от 2-х и более источников они ставятся в очередь и обслуживаются в порядке убывания уровня их приоритета. По обслуживании всех поступивших запросов контроллер переходит в режим ожидания очередного запроса.

Детали структуры и архитектуры контроллера *Intel 8237*, не существенные для описания принципов реализации ПДП, здесь не приведены; с ними можно ознакомиться по его *datasheet*.

Описанные в настоящем подразделе принципы реализации обмена в режиме ПДП и архитектурные решения контроллера ПДП являются типовыми (за исключением второстепенных деталей) практически для всех подсистем ПДП современных средств вычислительной техники, в том числе МК. Поэтому на основании представленных выше материалов можно перейти к рассмотрению структурно-архитектурных решений современных МК общего назначения.

8.3 Типовые структурно-архитектурные решения подсистемы ПДП МК общего назначения

Как указано в подразделе 8.1, наличие подсистемы ПДП, характерно, в основном, для семейств МК класса «*high performance*», в том числе *ARM Cortex-Mx*. Поэтому типовые структурно-архитектурные решения подсистемы ПДП будем рассматривать на примере МК данного семейства.

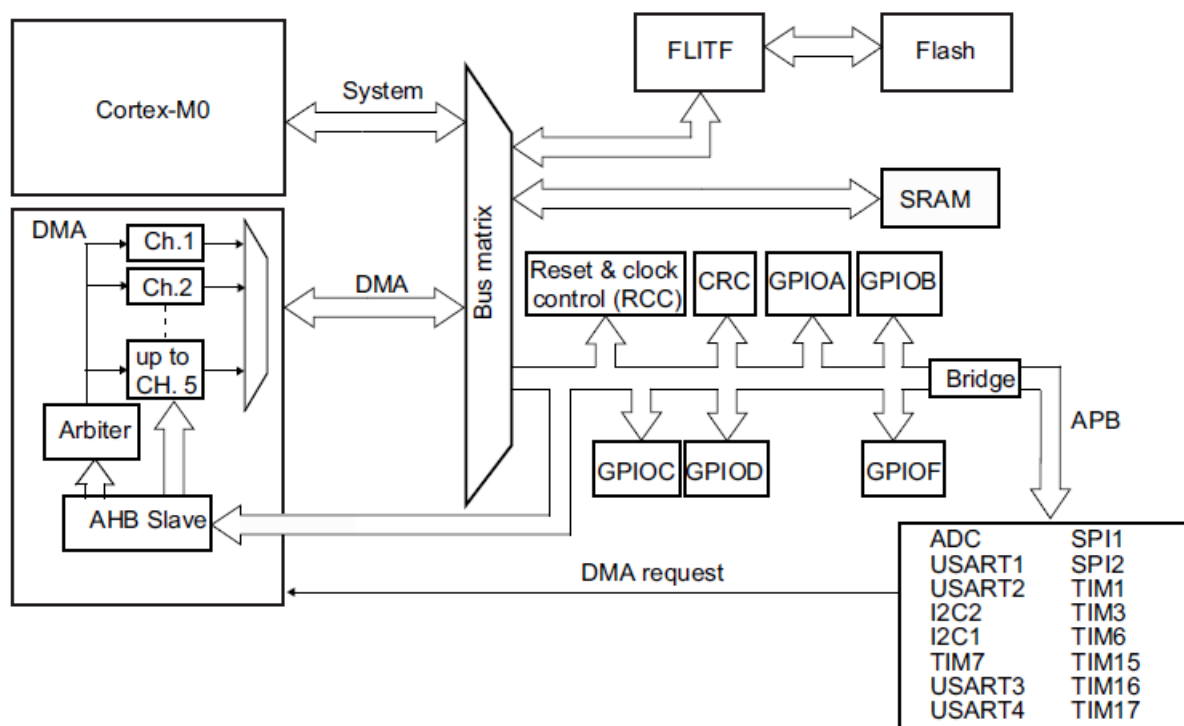
8.3.1. Базовые структурно-архитектурные решения и принципы работы подсистем ПДП МК семейства *ARM Cortex-Mx*

8.3.1.1. Структурные схемы и архитектура подсистем ПДП всех подсемейств / модельных МК семейства *ARM Cortex-Mx* существенно не отличаются по принципу построения (который, в целом, аналогичен описанному в подразделе 8.2). Различия состоят, в основном, в количестве контроллеров ПДП в составе блока, количестве каналов ПДП, а также в наличии или отсутствии буферной памяти каналов.

Наиболее проста подсистема ПДП МК подсемейства *ARM Cortex-M0*, с которой начнем рассмотрение базовых структурно-архитектурных решений и принципов работы подсистем ПДП МК семейства *ARM Cortex-Mx* в целом. Затем будут рассмотрены

особенности структуры и архитектуры подсистем ПДП подсемейств более высокого уровня сложности.

8.3.1.2. Структурная схема подсистемы ПДП МК модельного ряда *STM32F030xx* [15], принадлежащего к подсемейству *ARM Cortex-M0*, представлена на рис. 8.3.



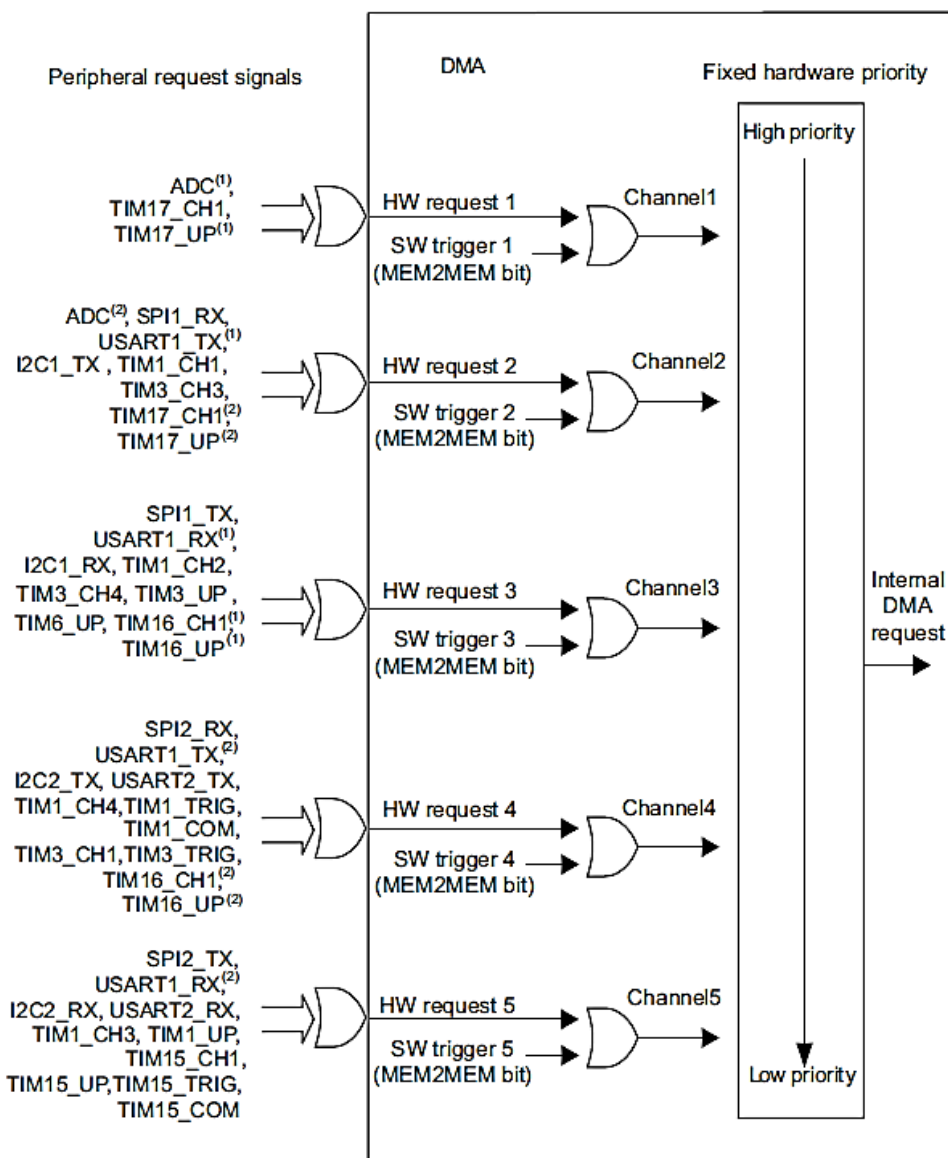
FLITF – Flash Memory Interface

Рис. 8.3. Структурная схема подсистемы ПДП МК модельного ряда *STM32F030xx* [15]

Центральным функциональным блоком подсистемы является 5-канальный контроллер ПДП, обозначенный на рис. 8.3 как *DMA*. Схема формирования запросов на сеанс ПДП от функциональных блоков МК по каналам приведена на рис. 8.4. Как видно из данного рисунка, по каждому из каналов может быть инициирован:

- аппаратный запрос (*HW request*), формируемый сигналами от нескольких различных ПУ, объединяемыми по схеме ИЛИ;
- программный запрос (*SW trigger*), формируемый по установке в единицу бита разрешения работы канала в его регистре конфигурации, при условии, что установлен бит *MEM2MEM* в том же регистре (подробнее – см. подпункт 8.3.1.11).

Аппаратный и программный запросы по каждому из каналов объединяются по схеме ИЛИ (см. рис. 8.4).



- (1) Данные запросы ПДП направляются на соответствующий канал только при сброшенном в ноль бите *DMA_RMP* соответствующего ПУ (например, *ADC_DMA_RMP*) в регистре *SYSCFG_CFGRI* (см. рис. 8.5)
- (2) Данные запросы ПДП направляются на соответствующий канал только при установленном в единицу бите *DMA_RMP* соответствующего ПУ (например, *ADC_DMA_RMP*) в регистре *SYSCFG_CFGRI* (см. рис. 8.5)

Рис. 8.4. Схема формирования запросов по каналам контроллера ПДП МК модельного ряда *STM32F030xx* [15]

Поскольку источником аппаратных запросов на ПДП по каждому из каналов может быть одно из нескольких ПУ (см. рис.

8.4), для обеспечения корректной идентификации источников запросов необходимо:

- по каждому из каналов разрешить формирование аппаратных запросов только одним из «приписанных» к нему ПУ (биты разрешения / запрета генерации запросов на ПДП располагаются в одном из РСФ ПУ, см. пункт 8.3.1.6);

- для выполнения вышеприведенного требования, при необходимости, произвести перераспределение (*remapping*) источников запросов по каналам с помощью битов с 8-го по 12-й и 26-го регистра *SYSCFG_CFGR1* (см. рис. 8.5 и пояснения к рис. 8.4).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	USART3_DMA_RMP	Res.	Res.	I2C_PA10_FMP	I2C_PA9_FMP	Res.	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP
					rw			rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	Res.	Res.	PA11_PA12_RMP	Res.	Res.	MEM_MODE [1:0]		
			rw	rw	rw	rw	rw			rw			rw	rw	

Рис. 8.5. Формат регистра *SYSCFG_CFGR1* МК модельного ряда *STM32F030xx* [15]

Программные запросы на ПДП генерируются ПО МК, поэтому проблема идентификации их источника, естественно, не существует.

Работа каждого из каналов, в целом, может быть программно разрешена или запрещена. В МК подсемейств *ARM Cortex-M0* разрешение / запрет осуществляются установкой или, соответственно, сбросом бита *EN* в регистре конфигурации соответствующего канала (*DMA_CCRx*) [15], см. рис. 8.10.

8.3.1.3. Архитектура подсистемы ПДП МК подсемейств *ARM Cortex-M0* и большинства модельных рядов *ARM Cortex-M3* [13, 15] допускает два **основных варианта** обмена данными – ПУ → память и память → ПУ. Как частный случай, возможна и реализация передачи ПУ → ПУ; при этом регистр данных одного из ПУ выполняет роль ячейки ПД. Направление передачи задается битом *DIR* регистра конфигурации канала (см. рис. 8.10).

8.3.1.4. Обработка запросов на ПДП осуществляется в соответствии с присвоенным им **приоритетом**. В подсистемах ПДП МК семейства *ARM Cortex-Mx* используется способ **многоуровневой квазистатической приоритезации** (см. подпункт 7.2.2.2). В соответствии со спецификой конкретной задачи, решаемой МК, каждому из каналов контроллера ПДП программно может быть назначен один из 4-х уровней приоритетности: очень высокий (*Very high*), высокий (*High*), средний (*Medium*) или низкий (*Low*). Назначение уровня приоритетности осуществляется посредством битового поля *PL[1:0]* в регистре конфигурации соответствующего канала (см. рис. 8.10). В пределах группы каналов с приоритетностью одинакового уровня приоритет тем выше, чем меньше номер канала (аналогично многоуровневой квазистатической приоритезации прерываний, см. подпункты 7.2.2.2 и 7.3.2.7).

«По умолчанию» (после сброса) всем каналам назначается одинаковый (низкий) уровень приоритетности, а их приоритет определяется только номером канала (см. рис. 8.4).

По поступлении запроса на передачу по каналу с более высоким приоритетом во время сеанса обмена данными по каналу с более низким приоритетом, текущий сеанс приостанавливается, и выполняется процедура передачи данных по более приоритетному каналу. По ее окончании, т. е. после считывания и записи последнего слова передаваемого массива, возобновляется сеанс обмена данными по каналу с более низким приоритетом.

Если же во время сеанса обмена данными по каналу с более высоким приоритетом поступает запрос по менее приоритетному каналу, этот запрос ставится в очередь, и обслуживается по окончании текущего сеанса.

8.3.1.5. В МК подсемейства *ARM Cortex-M0*, в каждом цикле ПДП-обмена данными передается только **одно слово** (см. описание структуры цикла в пункте 8.1.6). Циклы автоматически повторяются по поступлении каждого очередного аппаратного (от ПУ) или программного (от ЦП) запроса на ПДП (см. рис. 8.4), до тех пор, пока не будет передано количество слов, заданное при конфигурировании соответствующего канала (см. подпункт 8.3.1.11).

8.3.1.6. Аппаратные запросы на ПДП генерируются ПУ по определенным событиям, в зависимости от типа ПУ и разрешения / запрета формирования запросов по тому или иному событию. Например, в МК модельного ряда *STM32F030xx* [15]:

- запрос от АЦП генерируется по завершении очередного цикла преобразования, при единичном состоянии бита *DMAEN* в регистре *ADC_CFGR1*;

- запрос от блока *USART* может независимо формироваться по 2-м событиям – освобождению регистра данных передатчика и поступлению очередного кадра в регистр данных приемника; разрешение или запрет генерации запроса на ПДП по этим событиям осуществляется установкой или сбросом битов *DMAT* и *DMAR* соответственно в регистре *USART_CR3*.

Детальное описание условий генерации запросов на ПДП от ПУ МК конкретной модели содержится в *Reference Manual* модельного ряда, к которому принадлежит данный МК.

Как указано в подпункте 8.3.1.2, аппаратные запросы на ПДП от каждого из ПУ, являющихся их потенциальными источниками, могут быть программно разрешены или запрещены (аналогично запросам на прерывания), установкой или сбросом некоторого бита (битов) в одном из РСФ соответствующего ПУ. Например, в МК модельного ряда *STM32F030xx* [15] разрешение или запрет запросов на ПДП от АЦП осуществляется установкой или сбросом бита *DMAEN* в регистре *ADC_CFGR1*; запросов на ПДП от блока *USART* – установкой или сбросом битов *DMAT* и *DMAR* в регистре *USART_CR3* (см. выше).

Программные запросы на ПДП по некоторому каналу с номером *x* в МК подсемейства *ARM Cortex-M0* генерируются установкой в единицу бита разрешения работы соответствующего канала (*EN*) в регистре его конфигурации (*DMA_CCRx*), при предварительно установленном бите *MEM2MEM* в том же регистре [15] (см. также рис. 8.4).

8.3.1.7. Для МК семейства *ARM Cortex-Mx* характерна возможность программного задания разрядности и формата данных, передаваемых в процессе ПДП-сеанса. При этом разрядности слова данных, считываемого из памяти источника, и слова данных,

записываемого в память приемника за один цикл ПДП-обмена, задаются отдельно и могут различаться, например:

- при записи в ПД массива 16-битовых отсчетов АЦП (см. раздел 10), которые должны запоминаться в ПД и обрабатываться как 32-битовые числа, разрядность слова данных, считываемого из памяти источника (т. е. из регистра данных АЦП) равна 16-ти битам, а слова данных, записываемого в ПД – 32-м битам;

- при реализации на МК генератора сигнала способом *DDS* (см. Пример 2 в пункте 8.1.10), если разрядность ЦАП равна 8-и битам, а отсчеты синтезируемого сигнала хранятся в ПД как 16-битовые числа, разрядность слова данных, считываемого из ПД, равна 16-ти битам, а слова данных, записываемого в память приемника (т. е. в регистр данных ЦАП) – 8-и битам;

и т. п.

На рис. 8.6 представлена таблица возможных сочетаний разрядностей слов, считываемых из памяти источника и записываемых в память приемника в процессе ПДП-обмена данными в МК подсемейства *ARM Cortex-M0* [15]. Там же приведены форматы передачи данных и их размещения в памяти источника и приемника. Из данной таблицы, например, следует, что:

- при разрядности слова, считываемого из памяти источника, равной 16-ти битам, а слова, записываемого в память приемника – 32-м битам, данные передаются в формате $B_{i+1}B_i$ и записываются в память приемника как 32-битовые числа формата $0000B_{i+1}B_i$ (здесь 0 – **4-битовое** число, равное нулю, в шестнадцатеричной системе);

- при разрядности слова, считываемого из памяти источника, равной 16-ти битам, а слова, записываемого в память приемника – 8-и битам, данные передаются в формате $B_{i+1}B_i$, а в память приемника записывается только младший байт (B_i).

В МК семейства *ARM Cortex-Mx* при ПДП-передаче массива данных из ПД или в ПД, их считывание из ПД или их запись в ПД производятся в направлении **возрастания** адреса. При этом, после очередного цикла ПДП, адрес ПД автоматически увеличивается на **число байт** в слове, считываемом из ПД или, соответственно, записываемом в ПД.

Source port width	Destination port width	Number of data items to transfer (NDT)	Source content: address / data	Transfer operations	Destination content: address / data
8	8	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B1[7:0] @0x1 then WRITE B1[7:0] @0x1 3: READ B2[7:0] @0x2 then WRITE B2[7:0] @0x2 4: READ B3[7:0] @0x3 then WRITE B3[7:0] @0x3	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3
8	16	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 00B0[15:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 00B1[15:0] @0x2 3: READ B2[7:0] @0x2 then WRITE 00B2[15:0] @0x4 4: READ B3[7:0] @0x3 then WRITE 00B3[15:0] @0x6	@0x0 / 00B0 @0x2 / 00B1 @0x4 / 00B2 @0x6 / 00B3
8	32	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: READ B0[7:0] @0x0 then WRITE 000000B0[31:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 000000B1[31:0] @0x4 3: READ B2[7:0] @0x2 then WRITE 000000B2[31:0] @0x8 4: READ B3[7:0] @0x3 then WRITE 000000B3[31:0] @0xC	@0x0 / 000000B0 @0x4 / 000000B1 @0x8 / 000000B2 @0xC / 000000B3
16	8	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B2[7:0] @0x1 3: READ B5B4[15:0] @0x4 then WRITE B4[7:0] @0x2 4: READ B7B6[15:0] @0x6 then WRITE B6[7:0] @0x3	@0x0 / B0 @0x1 / B2 @0x2 / B4 @0x3 / B6
16	16	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B3B2[15:0] @0x2 3: READ B5B4[15:0] @0x4 then WRITE B5B4[15:0] @0x4 4: READ B7B6[15:0] @0x6 then WRITE B7B6[15:0] @0x6	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6
16	32	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: READ B1B0[15:0] @0x0 then WRITE 0000B1B0[31:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE 0000B3B2[31:0] @0x4 3: READ B5B4[15:0] @0x4 then WRITE 0000B5B4[31:0] @0x8 4: READ B7B6[15:0] @0x6 then WRITE 0000B7B6[31:0] @0xC	@0x0 / 0000B1B0 @0x4 / 0000B3B2 @0x8 / 0000B5B4 @0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BC[7:0] @0x3	@0x0 / B0 @0x1 / B4 @0x2 / B8 @0x3 / BC
32	16	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B5B4[15:0] @0x2 3: READ BBBAB9B8[31:0] @0x8 then WRITE B9B8[15:0] @0x4 4: READ BFBEBDBC[31:0] @0xC then WRITE BDBC[15:0] @0x6	@0x0 / B1B0 @0x2 / B5B4 @0x4 / B9B8 @0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B3B2B1B0[31:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B7B6B5B4[31:0] @0x4 3: READ BBBAB9B8[31:0] @0x8 then WRITE BBBAB9B8[31:0] @0x8 4: READ BFBEBDBC[31:0] @0xC then WRITE BFBEBDBC[31:0] @0xC	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC

Source port width, Destination port width – разрядность слова, считываемого из памяти источника и записываемого в память в приемника соответственно (задаются битовыми полями $MSIZE[1:0]$ и $PSIZE[1:0]$ регистра конфигурации канала, см. рис. 8.10)

Number of data items to transfer – число слов данных, подлежащих передаче

Source content – формат размещения данных в памяти источника

Destination content – формат размещения данных в памяти приемника

Bx – байт данных с номером x

@ - начальный адрес размещения передаваемых данных

@0x0, @0x1, @0x2,... - адреса, равные начальному плюс 0x0, 0x1, 0x2 и т. д.

Рис. 8.6. Возможные сочетания разрядностей и форматов слов, считываемых из памяти источника и записываемых в память приемника в процессе ПДП-обмена данными в МК подсемейств *ARM Cortex-M0* [15]

8.3.1.8. Архитектура подсистемы ПДП МК семейства *ARM Cortex-Mx* предусматривает возможность **циклического режима** обмена данными (*CIRC*, см. рис. 8.10), реализуемого следующим образом. После передачи количества слов, заданного при конфигурировании контроллера ПДП, счетчик слов и регистры адресов памяти источника и приемника данных автоматически устанавливаются в их начальные состояния, и запускается новый сеанс ПДП-обмена данными, с теми же параметрами конфигурации (количеством передаваемых слов, их разрядностью и форматами, адресами памяти источника и приемника и т. п.), что и у предыдущего сеанса. Данная процедура автоматически повторяется до тех пор, пока разрешена работа соответствующего канала ПДП.

Типовыми примерами использования циклического ПДП-обмена данными является формирование аналогового сигнала способом *DDS* (см. Пример 2 в пункте 8.1.10). В ПД хранятся отсчеты одного периода сигнала, передаваемые на ЦАП за один сеанс ПДП, по окончании которого автоматически запускается следующий сеанс, и формируется очередной период сигнала и т. д. (см. пример в пункте 8.4.2).

В циклическом режиме возможна реализация **кольцевой буферной памяти** в процессе ПДП-передачи. Данные из некоторого ПУ непрерывно поступают в область ПД, выделенную при конфигурировании контроллера ПДП. После заполнения половины данной области генерируется прерывание по событию «*Half-transfer*» (см. подпункт 8.3.1.10), по которому ЦП начинает считывание и обработку первой половины поступившего массива данных, во время которых заполняется вторая половина массива. После ее заполнения генерируется прерывание по событию «*Transfer complete*» (см. подпункт 8.3.1.10), по которому ЦП осуществляет считывание и обработку второй половины массива, а также запускается очередной сеанс ПДП, и происходит заполнение первой половины массива новыми данными и т. д.

8.3.1.9. Во время ПДП-сеанса контроллер ПДП осуществляет мониторинг отсутствия / наличия ошибок обмена данными (например, попыток чтения или записи физически не существующих адресов памяти). При обнаружении ошибки:

- работа вызвавшего ее канала запрещается (сбрасывается бит *EN* в регистре конфигурации канала);
- устанавливается в единицу бит признака ошибки по соответствующему каналу в регистре статуса контроллера ПДП;
- вырабатывается запрос на прерывание по ошибке передачи (*Transfer Error Interrupt, TEI*), если оно разрешено (см. подпункт 8.3.1.10).

8.3.1.10. Каждый из каналов контроллера ПДП может быть источником **запросов на прерывания** по нескольким событиям (естественно, при условии, что соответствующие прерывания разрешены). В большинстве моделей МК подсемейства *ARM Cortex-M0* источниками прерываний могут быть следующие события по каждому из каналов ПДП [15]:

- завершение сеанса передачи массива данных по каналу (*Transfer Complete Interrupt, TCI*);
- завершение передачи половины массива данных по каналу (*Half Transfer Interrupt, HTI*); используется, в частности, при реализации кольцевой буферной памяти (см. подпункт 8.3.1.8);
- ошибка передачи по каналу (*Transfer Error Interrupt, TEI*), см. подпункт 8.3.1.9.

В таблице векторов прерываний не выделяется специальных позиций под прерывания по каждому из данных событий; отдельные позиции назначаются только каналам или группам каналов ПДП, под прерывания по какому-либо из вышеперечисленных событий в соответствующем канале или в одном из каналов группы (см. рис. 8.7). Например, в МК модельного ряда *STM32F030xx* прерывание по любому из 3-х данных событий как во 2-м, так и в 3-м канале ПДП вызовет одну и ту же подпрограмму обслуживания.

Position	Priority	Type of priority	Acronym	Description	Address
9	16	settable	DMA_CH1	DMA channel 1 interrupt	0x0000 0064
10	17	settable	DMA_CH2_3	DMA channel 2 and 3 interrupts	0x0000 0068
11	18	settable	DMA_CH4_5	DMA channel 4 and 5 interrupts	0x0000 006C

Рис. 8.7. Фрагмент таблицы векторов прерываний МК модельного ряда *STM32F030xx* [15]

Таким образом, при обслуживании прерываний от контроллера ПДП требуется идентификация конкретного события, вызвавшего прерывание, что характерно в целом для подсистемы прерываний МК семейства *ARM Cortex-Mx* (см. подпункты 7.3.2.15 и 7.3.2.16). Для возможности идентификации номера канала и конкретного события, явившихся источниками прерывания от контроллера ПДП, в его составе имеется специальный регистр признаков событий, которые могут быть вызывать прерывания (*DMA interrupt status register, DMA_ISR*). Пример формата данного регистра представлен на рис. 8.8. В нем по каждому из каналов ПДП выделяется по одному биту под признак (флаг) каждого из событий, являющихся потенциальными источниками прерываний (в МК подсемейства *ARM Cortex-M0* такими событиями являются *Transfer Complete, Half Transfer* или *Transfer Error*, см. ранее). Например, единичное состояние бита *TCIF3* служит признаком завершения сеанса передачи по 3-му каналу и т. п. Кроме того, для идентификации канала, вызвавшего прерывание, в регистр *DMA_ISR* введены биты *GIFx* (*Channel x global interrupt flag*). Единичное состояние бита *GIFx* является признаком наличия одного из событий – источников прерываний по каналу *x*. Например, если бит *GIF5* равен единице, это означает, что в 5-м канале произошло одно из 3-х вышеперечисленных событий.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF5	HTIF5	TCIF5	GIF5
												г	г	г	г
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
г	г	г	г	г	г	г	г	г	г	г	г	г	г	г	г

Рис. 8.8. Формат регистра *DMA_ISR* МК модельного ряда *STM32F030xx* [15]

Идентификация источника прерывания должна производиться в подпрограмме его обслуживания; в общем случае, ее рационально осуществлять в следующем порядке:

- вначале опросом битов *GIFx* определить номер канала, вызвавшего прерывание (данная процедура, естественно, не требуется, если разрешена работа только **одного** канала ПДП);
- затем опросом битов *TCIF*, *HTIF* и *TEIF* соответствующего канала идентифицировать событие, вызвавшее прерывание;
- после идентификации номера канала и события, вызвавших прерывание, выполнить сброс находящихся в активном состоянии битов регистра *DMA_ISR*, для предотвращения повторных запусков подпрограммы обслуживания прерывания после выхода из нее (см. подпункт 7.3.2.16);
- перейти к процедуре обслуживания конкретного события в конкретном канале (например, считывания массива данных из ПД по событию *Transfer Complete*).

Заметим, что все биты регистра *DMA_ISR* доступны только для чтения (см. рис. 8.8), а их сброс осуществляется записью единицы в соответствующие им биты регистра *DMA interrupt flag clear register (DMA_IFCR)*, см. рис. 8.9. Например, сброс бита *TCIF3* в регистре *DMA_ISR* – записью единицы в бит *CTCIF3 (Clear TCIF3)* регистра *DMA_IFCR* и т. п.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTEIF5	CHTIF5	CTCIF5	CGIF5
												w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Рис. 8.9. Формат регистра *DMA_IFCR* МК модельного ряда *STM32F030xx* [15]

Аналогичным образом может быть организовано обслуживание событий *Transfer Complete*, *Half Transfer* или *Transfer Error*, если прерывания по ним запрещены по каким-либо причинам. Разрешение / запрет прерываний по перечисленным событиям осуществляются соответственно установкой / сбросом битов *TCIE (Transfer complete interrupt enable)*, *HTIE (Half transfer interrupt enable)* и *TEIE (Transfer error interrupt enable)* регистра конфигурации соответствующего канала.

Необходимо отметить, что в ряде МК подсемейства *ARM Cortex-M3*, в также в МК подсемейств *ARM Cortex-M4* и выше, состав событий, которые могут быть источниками прерываний от контроллера ПДП, несколько шире, чем в МК подсемейств *ARM Cortex-M0* (см. пункт 8.3.2).

8.3.1.11. Как указано ранее, каналы ПДП, которые планируется использовать в процессе работы МК, предварительно должны быть **сконфигурированы**, т. е. должны быть программно заданы параметры обмена данными по каждому из используемых каналов. В МК подсемейства *ARM Cortex-M0* (а также в МК большинства модельных рядов подсемейства *ARM Cortex-M3*) к параметрам ПДП-обмена данными относятся [15]:

- базовый адрес памяти ПУ;
- базовый адрес ПД;
- число слов данных, подлежащих передаче за сеанс;
- направление передачи (ПУ → память или память → ПУ);
- уровень приоритетности, назначаемый каналу (см. подпункт 8.3.1.4);
- разрядность слова, считываемого из памяти источника, и слова, записываемого в память приемника (см. рис. 8.6 и пояснения к нему);
- разрешение / запрет автоматического увеличения адресов ПД и памяти ПУ после передачи очередного слова данных;
- разрешение / запрет циклического режима работы (см. подпункт 8.3.1.8);
- разрешение / запрет программной инициализации сеансов ПДП (*SW trigger*); см. рис. 8.4 и пояснения к нему;
- разрешение / запрет прерываний по событиям, перечисленным в подпункте 8.3.1.10.

Задание вышеперечисленных параметров ПДП-обмена осуществляется посредством следующих программно-доступных регистров, входящих в состав каждого из каналов контроллера ПДП (сравните с рис. 8.2):

- регистра 32-битового базового адреса памяти ПУ, *DMA channel x peripheral address register (DMA_CPARx)*;
- регистра 32-битового базового адреса памяти ПУ, *DMA channel x memory address register (DMA_CMARx)*;

- регистра количества слов, подлежащего передаче во время сеанса ПДП, *DMA channel x number of data register (DMA_CNDTRx)*;
- регистра конфигурации канала, *DMA channel x configuration register (DMA_CCRx)*, формат которого представлен на рис. 8.10 и посредством которого назначаются все остальные параметры ПДП-сеанса.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEM2 MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 8.10. Формат регистра конфигурации ПДП-канала МК
модельного ряда *STM32F030xx* [15]
(пояснения приведены в тексте)

Назначение битов / битовых полей регистра конфигурации канала (см. рис. 8.10) следующее [15]:

- разрешение / запрет программной инициализации сеансов ПДП (*SW trigger*) выполняется установкой / сбросом бита *MEM2MEM* (см. подпункт 8.3.1.6);

- уровень приоритетности канала задается битовым полем *PL[1:0]*; его значениям 00, 01, 10 и 11 соответствуют уровни «Низкий», «Средний», «Высокий» и «Очень высокий» (см. подпункт 8.3.1.4);

- разрядность слов данных, считываемых из ПД / записываемых в ПД задается битовым полем *MSIZE[1:0]*, а считываемых из памяти ПУ / записываемых в память ПУ – полем *PSIZE[1:0]*; значениям 00, 01 и 10 данных полей соответствуют разрядности 8, 16 и 32 бита (см. подпункт 8.3.1.7); значение 11 зарезервировано;

- разрешение / запрет автоматического увеличения адресов ПД и памяти ПУ после передачи очередного слова данных осуществляется установкой / сбросом битов *MINC* и *PINC* соответственно; напомним, что увеличение (при его разрешении) осуществляется на число байт в слове ПД или памяти ПУ; в частности, в Примере 1 пункта 8.1.10 значение *MINC* должно быть

равно единице, а *PINC* – нулю (считывание осуществляется с одного и того же адреса – из регистра данных АЦП);

- разрешение / запрет циклического режима работы производится установкой / сбросом бита *CIRC* (см. подпункт 8.3.1.8);

- направление передачи данных задается битом *DIR*; его нулевому значению соответствует передача «ПУ → память», а единичному – «память → ПУ» (см. подпункт 8.3.1.3);

- разрешение / запрет прерываний по событиям *Transfer Complete*, *Half Transfer* и *Transfer Error* осуществляется установкой / сбросом битов *TCIE*, *HTIE* и *TEIE* соответственно (см. подпункт 8.3.1.10).

- разрешение / запрет работы канала осуществляются установкой / сбросом бита *EN*.

Необходимо отметить, что разрешение работы канала (установку бита *EN* в единичное состояние) следует производить **только** после задания параметров ПДП-сеанса, т. е. базовых адресов ПД и памяти ПУ, количества слов данных, подлежащих передаче, а также параметров, назначаемых посредством регистра конфигурации канала [15]. При *EN* = 1 запись регистров *DMA_CMARx* и *DMA_CPARx* недопустима, а регистр *DMA_CNDTRx* недоступен для записи.

8.3.1.12. Программный запрос (*SW trigger*) на сеанс ПДП (см. подпункт 8.3.1.2) генерируется записью единицы в бит *EN* при установленном бите *MEM2MEM*. Сеанс ПДП запускается непосредственно после установки в единицу бита *EN*. ПДП-сеанс, инициированный программным запросом, завершается только после передачи количества слов, заданного при конфигурировании канала [15].

При ПДП-обмене данными, инициируемом программными запросами, недопустимо использование циклического режима работы канала [15].

8.3.1.13. Задаваемое регистром *DMA_CNDTRx* максимальное число слов, передаваемых за ПДП-сеанс, равно 65535. Содержимое *DMA_CNDTRx* декрементируется после каждого цикла передачи. После его обнуления сеанс обмена завершается (если циклический

режим работы запрещен) или в регистр *DMA_CNDTRx* автоматически загружается количество слов, заданное при конфигурировании, и начинается новый сеанс (если циклический режим разрешен).

8.3.1.14. Следует также заметить, что, если разрядность слов данных, считываемых из ПД / записываемых в ПД равна 16-и битам, 0-й разряд регистра *DMA_CMARx* игнорируется, а при разрядности указанных слов, равной 32-м битам – игнорируются 0-й и 1-й разряды данного регистра. То же относится и к регистру *DMA_CPARx*, если разрядность слов, считываемых из памяти ПУ / записываемых в память ПУ равна 16-и или 32-м битам соответственно. Данные особенности должны учитываться при задании базовых адресов ПД и памяти ПУ, а также при размещении в ПД слов, считываемых или записываемых в процессе ПДП-сеанса.

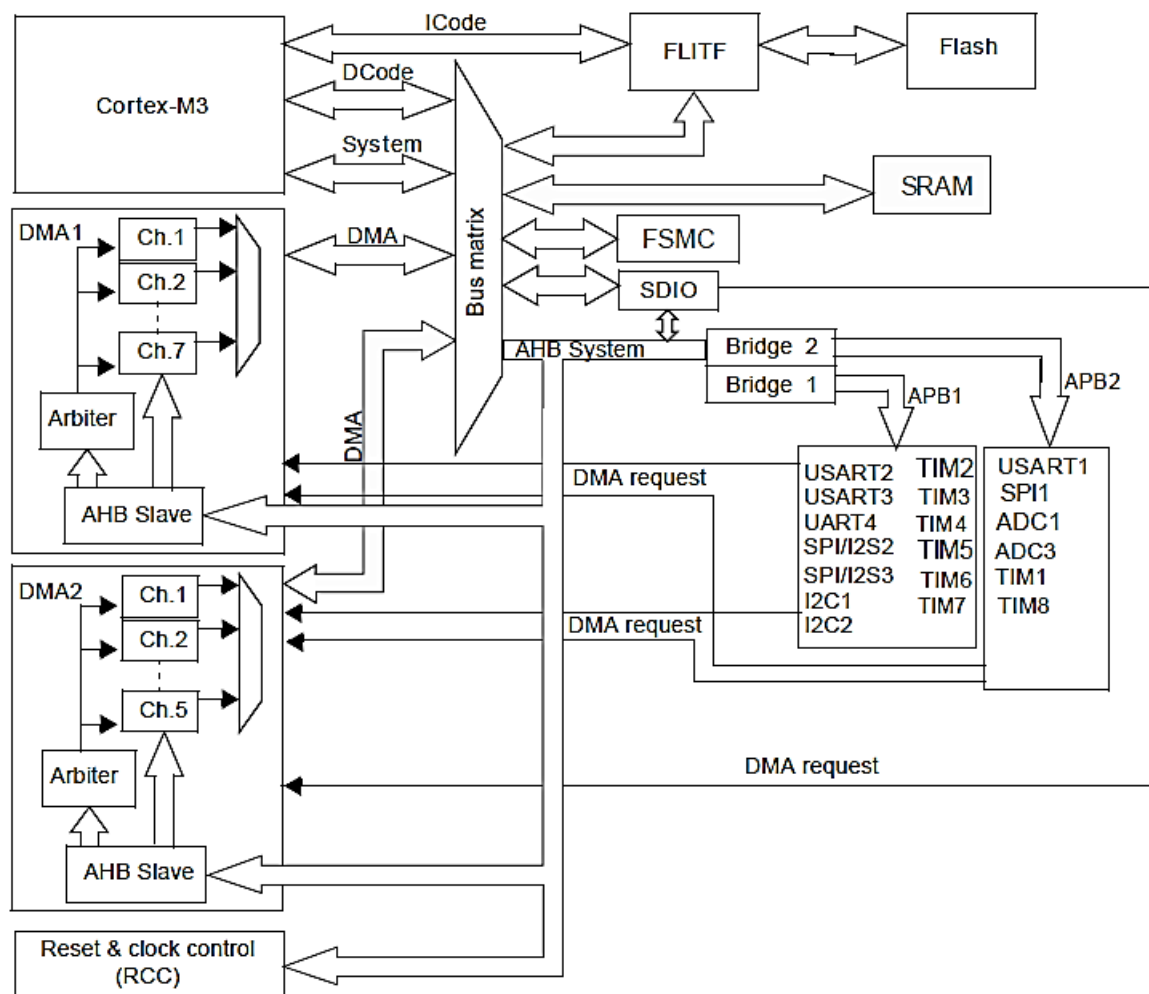
8.3.1.15. При реализации обмена «ПУ → ПУ» в регистры *DMA_CMARx* и *DMA_CPARx* записываются адреса регистров данных соответствующих ПУ. Направление передачи определяется состоянием бита *DIR* регистра конфигурации: при его нулевом значении источником данных служит регистр, адрес которого записан в *DMA_CPARx*, а при единичном – адрес которого указан в *DMA_CMARx*. Биты *MINC* и *PINC* сбрасываются в нулевое состояние.

8.3.2. Особенности структуры и архитектуры подсистемы ПДП МК подсемейств *ARM Cortex-M3* и *ARM Cortex-M4*

8.3.2.1. В настоящем пункте рассматриваются основные отличительные особенности структуры и архитектуры подсистемы ПДП более «продвинутых» по сравнению с *ARM Cortex-M0* подсемейств МК семейства *ARM Cortex-Mx*. Ограничимся наиболее распространенными и относительно несложными подсемействами *ARM Cortex-M3* и *ARM Cortex-M4* [9, 13, 14].

8.3.2.2. На рис. 8.11 представлена обобщенная структурная схема подсистемы ПДП МК популярного модельного ряда *STM32F10xx* [13], принадлежащего к подсемейству *ARM Cortex-M3*. В данном модельном ряду выделяют группы МК малой, средней,

повышенной и высокой сложности (*low-, medium-, high- and XL-density lines*). В группах *low-* и *medium-density* 2-й контроллер ПДП **отсутствует**, а структура имеющегося контроллера отличается от рассмотренных в пункте 8.3.1 структурных решений контроллера ПДП подсемейства *ARM Cortex-M0* только второстепенными деталями (например, наличием 7-и, а не 5-и каналов ПДП). Формирование запросов по каналам реализуется по схеме, которая аналогична представленной на рис. 8.4.



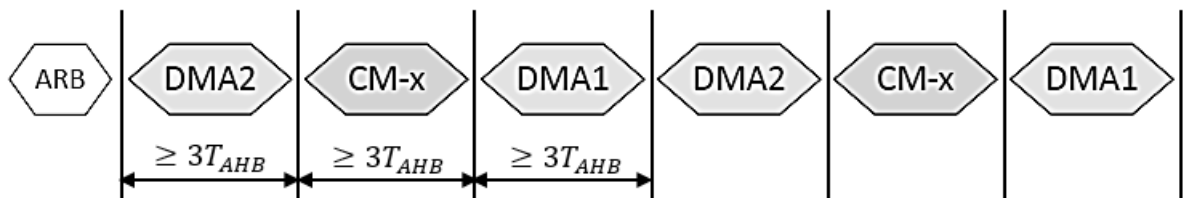
FLITF – Flash Memory Interface

FSMC - Flexible static memory controller

Рис. 8.11. Обобщенная структурная схема подсистемы ПДП МК модельного ряда *STM32F10xx* [13]

Подсистема ПДП тех МК модельного ряда *STM32F10xx*, которые относятся к группам *high-* и *XL-density*, содержит два

контроллера ПДП (см. рис. 8.11), конфигурируемых и работающих независимо друг от друга. При одновременной работе обоих контроллеров распределение времени доступа к *АНВ* и ПД между ними и ЦП реализуется в соответствии с временными диаграммами, приведенными на рис. 8.12 [46]. При этом 1-й контроллер обладает более высоким приоритетом, чем 2-й [13].



ARB – интервал обработки запроса на ПДП

CM-x, *DMA1*, *DMA2* – интервалы времени, в течение которых доступ к ПД и *АНВ* предоставлен ЦП, 1-му и 2-му контроллерам ПДП соответственно

T_{AHB} – период тактовой частоты *АНВ*-домена

Рис. 8.12. Распределение времени доступа к *АНВ* и ПД между ЦП и контроллерами ПДП при их одновременной работе [46]

8.3.2.3. Все основные элементы архитектуры контроллеров ПДП МК модельного ряда *STM32F10xx*, а именно:

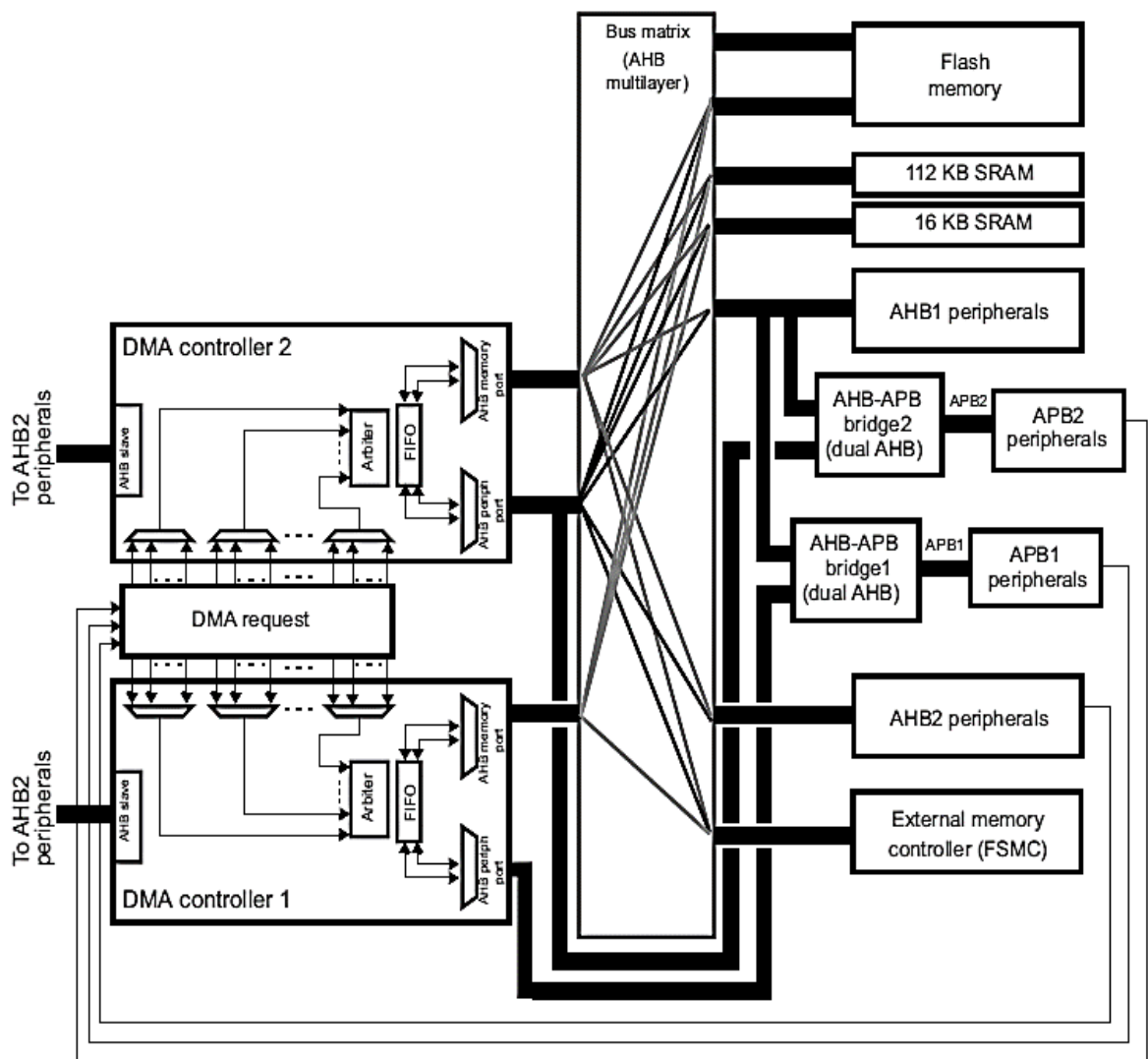
- режимы и принципы работы;
- способы инициализации ПДП-сеансов;
- принципы приоритетного обслуживания запросов
- возможные сочетания разрядностей и форматов слов источника и приемника в процессе ПДП-обмена данными (см. рис. 8.6);

- источники прерываний;
- состав программно-доступных регистров контроллера ПДП и назначения их битов / битовых полей;

аналогичны соответствующим элементам архитектуры МК модельного ряда *STM32F030xx* (см. пункт 8.3.1).

8.3.2.4. Существенно отличаются от ранее рассмотренных структурно-архитектурные решения подсистемы ПДП более «продвинутых» модельных рядов МК подсемейства *ARM Cortex-M3*, например, *STM32F20x* и *STM32F21x* [48], а также МК подсемейства *ARM Cortex-M4* (см., например, [14]). Структурные схемы

подсистемы ПДП МК модельных рядов *STM32F20x* и *STM32F21x* и их контроллера ПДП представлены на рис. 8.13 и рис. 8.14 соответственно. Практически идентичны им структурные схемы подсистемы ПДП и контроллера ПДП большинства модельных рядов МК подсемейства *ARM Cortex-M4*, в частности, *STM32F405xx/07xx* и *STM32F415xx/17xx* [14].



AHB slave – порт ведомого шины *AHB*

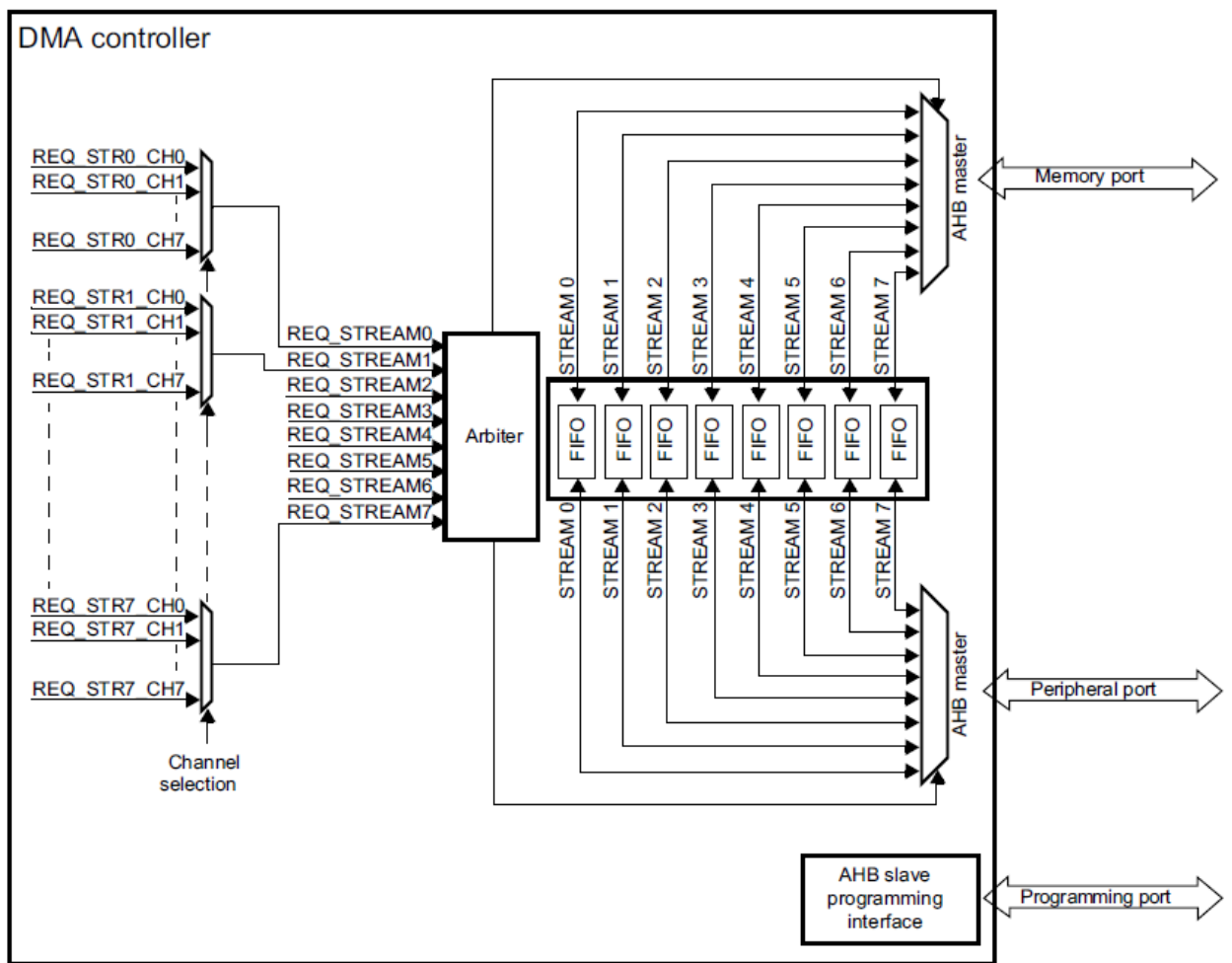
DMA request – запросы на ПДП

FIFO – буферная память типа *First In – First Out*
(«Первым пришел – первым вышел»)

Bus matrix (AHB multilayer) – матрица шин (многоуровневая шина *AHB*)

AHB-APB bridge – мост между шинами *AHB* и *APB* (см. подраздел 1.3)

Рис. 8.13. Структурная схема подсистемы ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48]



REQ_STRx_CHy – запрос на ПДП по каналу y потока x

$REQ_STREAMx$ – запрос на ПДП по потоку x

AHB master – порт ведущего шины *AHB*

Рис. 8.14. Структурная схема контроллера ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48]

Рассмотрим подробнее особенности структуры и архитектуры подсистемы ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48]. Отметим, что они в равной мере относятся и к структурно-архитектурным решениям подсистемы ПДП модельных рядов *STM32F405xx/07xx* и *STM32F415xx/17xx* [14], принадлежащих подсемейству *ARM Cortex-M4*.

8.3.2.5. Каналу ПДП более простых МК (см. рис. 8.3, 8.4 и 8.11) в архитектуре модельных рядов *STM32F20x* и *STM32F21x* (а также МК подсемейства *ARM Cortex-M4*) соответствует понятие «**поток**»

(*Stream*); при этом **каналами** называются линии запросов на ПДП от ПУ МК (см. рис. 8.14).

Режимы и параметры ПДП-обмена по каждому из потоков задаются независимо, посредством набора регистров, в состав которого входят:

- регистры *DMA_SxM0AR* и *DMA_SxM1AR* базовых адресов памяти устройства, подключенного к порту *AHB memory port* контроллера ПДП (наличие 2-х таких регистров необходимо для возможности реализации двойного буфера, см. подпункт 8.3.2.9);
- регистр *DMA_SxPAR* базового адреса памяти устройства, подключенного к порту *AHB peripheral port* контроллера ПДП;
- регистр конфигурации потока (*DMA_SxCR*);
- регистр управления буфером *FIFO* (*DMA_SxFCR*);

где *x* – номер потока (например, *DMA_S3CR*, *DMA_S5FCR* и т. п.). Назначение данных регистров и их битов (битовых полей) будут рассмотрены по ходу последующего изложения.

Запросы на ПДП от ПУ по каждому из потоков объединяются не по схеме ИЛИ (см. рис. 8.4), а посредством **мультиплексоров** (см. рис. 8.14), управляемых битовым полем «*Channel selection*» программно-доступных регистров конфигурации потоков (что, в частности, устраняет проблему идентификации источников запросов, см. подпункт 8.3.1.2).

Схема формирования запросов на ПДП по потоку *x* контроллера ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48], а также *STM32F405xx/07xx* и *STM32F415xx/17xx* [14] приведена на рис. 8.15. Запросы по каждому из потоков поступают через мультиплексор от 8-и источников (с 8-и каналов). При этом в конкретном потоке число реально задействованных каналов может быть меньше 8-и (см. рис. 8.16). Выбор источника запросов по потоку осуществляется битовым полем *CHSEL[2:0]* регистра *DMA_SxCR* конфигурации соответствующего потока (см. также рис. 8.22).

На рис. 8.16 в качестве примера представлена таблица источников запросов 1-го контроллера ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48].

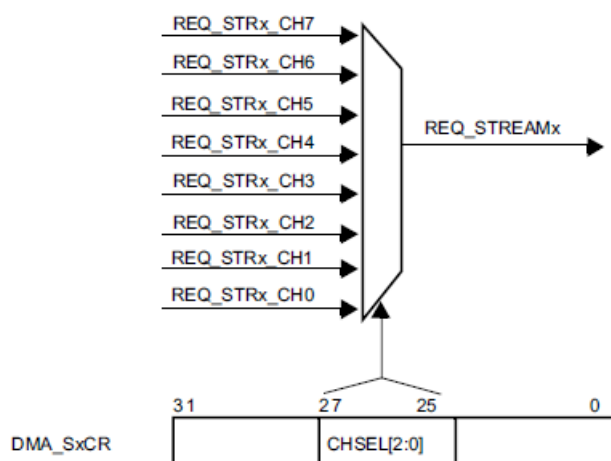


Рис. 8.15. Схема формирования запросов по потоку x контроллера ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48], *STM32F405xx/07xx* и *STM32F415xx/17xx* [14]

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX	-	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	-	SPI3_TX
Channel 1	I2C1_RX	-	TIM7_UP	-	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1	-	I2S3_ext_RX	TIM4_CH2	I2S2_ext_TX	I2S3_ext_TX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_ext_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_ext_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5	-	-	TIM3_CH4 TIM3_UP	-	TIM3_CH1 TIM3_TRIG	TIM3_CH2	-	TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
Channel 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

Рис. 8.16. Таблица источников запросов 1-го контроллера ПДП МК модельных рядов *STM32F20x* и *STM32F21x* [48]

8.3.2.6. Функции устройства, инициирующего завершение сеанса ПДП (*Flow controller*) могут выполнять как контроллер ПДП, так и те из ПУ, которые обладают свойством вырабатывать запрос на окончание сеанса ПДП при передаче / приеме последнего слова данных. Данное свойство используется, если число слов, подлежащих передаче за сеанс, изначально неизвестно. В МК модельных рядов *STM32F20x* и *STM32F21x* таким свойством обладает только блок интерфейса *SDIO* (*Secure digital input/output*

interface). Назначение устройства, выполняющего функцию *Flow controller* по некоторому потоку, осуществляется битом *PFCTRL* (*Peripheral flow controller*) регистра конфигурации соответствующего потока (см. рис. 8.22). При нулевом значении данного бита функцию *Flow controller* выполняет контроллер ПДП, а при единичном – ПУ, способное выполнять данную функцию.

8.3.2.7. Подсистемой ПДП рассматриваемых модельных рядов МК поддерживаются варианты обмена «ПУ → память», «память → ПУ» и «память → память» (последний – только 2-м контроллером ПДП). При первых двух вариантах обмена статусом ПУ обладает устройство, подключенное к порту *AHB peripheral port* контроллера ПДП, а статусом памяти – подключенное к порту *AHB memory port*. При обмене «память → память» устройство, подключенное к порту *AHB peripheral port*, служит источником данных, а подключенное к порту *AHB memory port* – приемником. Задание вариантов обмена по каждому из потоков осуществляется битовым полем *DIR[1:0]* регистра конфигурации потока (см. рис. 8.22); его значениям 00, 01 и 10 соответствуют режимы обмена «ПУ → память», «память → ПУ» и «память → память»; значение 11 зарезервировано.

В отличие от регистровых моделей контроллеров ПДП, рассмотренных в пункте 8.3.1 и в подпунктах 8.3.2.2 и 8.3.2.3, в составе регистров каждого из ПДП-потоков МК модельных рядов *STM32F20x* и *STM32F21x* (а также *STM32F405xx/07xx* и *STM32F415xx/17xx*) имеется два регистра базового адреса памяти устройства, подключенного к порту *AHB memory port* – *DMA_SxM0AR* и *DMA_SxM1AR*. Оба данных регистра задействуются только в режиме **двойного буфера** (см. подпункт 8.3.2.9); если данный режим не используется, задействуется только регистр *DMA_SxM0AR*. Регистр базового адреса памяти ПУ (т. е. устройства, подключенного к *AHB peripheral port*) один на поток, его имя – *DMA_SxPAR* (где *x* – номер потока).

В режимах обмена «ПУ → память» и «память → ПУ» базовый адрес памяти ПУ указывается в регистре *DMA_SxPAR*; базовый адрес устройства, подключенного к порту *AHB memory port*, если не используется режим двойного буфера – в регистре *DMA_SxM0AR*. При использовании данного режима задействуются две области

памяти устройства, подключенного к порту *AHB memory port*, базовые адреса которых указываются в регистрах *DMA_SxM0AR* и *DMA_SxM1AR* (подробнее – см. подпункт 8.3.2.9).

При обмене «память → память»:

- базовый адрес области памяти, служащей источником данных, должен быть записан в регистр *DMA_SxPAR*;

- при использовании режима двойного буфера базовые адреса областей памяти, служащих приемниками данных, записываются в регистры *DMA_SxM0AR* и *DMA_SxM1AR*; если же режим двойного буфера не используется, базовый адрес области памяти, служащей приемником данных, должен быть записан в регистр *DMA_SxM0AR*.

Разрешение / запрет автоматического увеличения адресов памяти устройств, подключенных к портам *AHB memory port* и *AHB peripheral port*, после очередного цикла ПДП, осуществляется установкой / сбросом соответственно битов *MINC* и *PINC* регистра конфигурации потока (см. также рис. 8.22). Значения, на которое увеличиваются адреса, зависят от разрядности данных, передаваемых через соответствующие порты (подробнее – см. подпункт 8.3.2.11).

Инициализация очередного цикла ПДП-сеанса обмена данными в режимах «ПУ → память» и «память → ПУ» осуществляется запросами от ПУ (см. рис. 8.14 – 8.16), аналогично тому, как это происходит в подсистемах ПДП, рассмотренных в пункте 8.3.1 и в подпунктах 8.3.2.2 и 8.3.2.3. **Завершение** сеанса ПДП в режиме обмена «ПУ → память» и «память → ПУ» происходит при одном из следующих событий:

- обнулении счетчика передаваемых слов, т. е. после передачи количества слов, указанного в регистре *DMA stream x number of data register (DMA_SxNDTR)*;

- по запросу на окончание сеанса ПДП от ПУ, способного выполнять функцию *Flow Controller* (если оно назначено в качестве устройства, управляющего сеансом ПДП, при конфигурировании потока, см. подпункт 8.3.2.6);

- при программном сбросе в ноль бита разрешения (*EN*) в регистре конфигурации соответствующего потока.

В режиме «память → память» **запуск** сеанса обмена данными осуществляется программной установкой в единицу бита разрешения (*EN*) в регистре конфигурации соответствующего потока, при предварительном выборе режима «память → память» записью кода 10 в битовое поле *DIR[1:0]* данного регистра. **Завершение** сеанса при этом происходит или после передачи количества слов, указанного в регистре *DMA_SxNDTR*, или при программном сбросе бита *EN* (см. выше).

8.3.2.8. Архитектура подсистемы ПДП рассматриваемых модельных рядов, как и рассмотренных ранее, предоставляет возможность реализации **циклического** режима работы по общим принципам, описанным в подпункте 8.3.1.8. Задание этого режима осуществляется установкой в единицу бита *CIRC* в регистре конфигурации потока (см. рис. 8.22). Использование циклического режима при обмене «память → память» **недопустимо**.

8.3.2.9. В циклическом режиме работы потока имеется возможность организации **двойного буфера** в памяти (только при вариантах обмена «ПУ → память» и «память → ПУ»). При этом необходимо установить в единицу бит *DBM* в регистре конфигурации соответствующего потока (см. рис. 8.22), и зарезервировать две области памяти. Их базовые адреса указываются в регистрах *DMA_SxM0AR* и *DMA_SxM1AR* (см. подпункт 8.3.2.7). По заполнении / считывании одной из них автоматически начинается новый сеанс ПДП, во время которого заполняется / считывается другая область, а ЦП осуществляет считывание / запись области, заполненной / считанной в предыдущем сеансе ПДП, и т. д.

Примечание. В режиме двойного буфера возможно изменение содержимого регистров *DMA_SxM0AR* и *DMA_SxM1AR* в интервалы времени, в которые соответствующий регистр «не активен», т. е. не является указателем базового адреса области памяти, заполняемой / считываемой в процессе обмена данными с ПУ. Признаком того, какой из данных регистров «активен» во время текущего ПДП-сеанса, является состояние бита *CT* (*Current Target*) регистра конфигурации потока (см. рис. 8.22). При *CT* = 0 «активен» регистр *DMA_SxM0AR*, а при *CT* = 1 – *DMA_SxM1AR*. Попытка изменения

содержимого «активного» регистра вызовет событие «*Transfer Error*» (см. подпункт 8.3.2.14).

8.3.2.10. Каждый из потоков снабжен **буферной памятью** типа ***FIFO*** («Первым пришел – первым вышел») объемом 4 слова (16 байт), предназначенной для промежуточного хранения данных в процессе их ПДП-передачи (подробнее – см. подпункт 8.3.2.12).

Обмен «ПУ → память» и «память → ПУ» может также осуществляться в режиме ***Direct mode***, в котором буфер *FIFO* «прозрачен», и не используется для промежуточного хранения данных, а только для их передачи от источника приемнику. Разрешение / запрет режима *Direct mode* осуществляется сбросом / установкой бита *DMDIS* (*Direct mode disable*) в регистре управления буфером *FIFO* соответствующего потока, *DMA stream x FIFO control register* (*DMA_SxFCR*), см. рис. 8.23. При обмене «память → память» применение режима *Direct mode* не допускается; если выбран данный вариант обмена (см. подпункт 8.3.2.6), бит *DMDIS* автоматически устанавливается в единицу на аппаратном уровне.

8.3.2.11. Возможные сочетания **разрядностей и форматов слов**, считываемых из памяти источника и записываемых в память приемника в процессе ПДП-обмена данными, несколько отличаются от описанных в подпункте 8.3.1.7.

Разрядности слов данных, передаваемых через порты *AHB memory port* и *AHB peripheral port* контроллера ПДП, могут быть равны 8-и, 16-и или 32-м битам. Они задаются битовыми полями соответственно *MSIZE[1:0]* и *PSIZE[1:0]* регистра конфигурации потока (см. рис. 8.22). Значениям 00, 01 и 10 данных битовых полей соответствует разрядность 8, 16 и 32 бита; значение 11 зарезервировано.

В режиме работы *Direct mode* (т. е. если буфер *FIFO* «прозрачен» и не используется для хранения данных) разрядности слов, как считываемых из памяти источника, так и записываемых в память приемника, должны быть **одинаковы**. Они задаются битовым полем *PSIZE[1:0]*; поле *MSIZE[1:0]* при этом игнорируется.

При использовании буфера *FIFO* разрядности и форматы слов данных, считываемых из памяти источника и записываемых в

память приемника, могут быть различны. Их возможные сочетания представлены на рис. 8.17.

AHB memory port width	AHB peripheral port width	Number of data items to transfer (NDT)	Memory transfer number	Memory port address / byte lane	Peripheral transfer number	Peripheral port address / byte lane	
						PINCOS = 1	PINCOS = 0
8	8	4	1	0x0 / B0[7:0]	1	0x0 / B0[7:0]	0x0 / B0[7:0]
			2	0x1 / B1[7:0]	2	0x4 / B1[7:0]	0x1 / B1[7:0]
			3	0x2 / B2[7:0]	3	0x8 / B2[7:0]	0x2 / B2[7:0]
			4	0x3 / B3[7:0]	4	0xC / B3[7:0]	0x3 / B3[7:0]
8	16	2	1	0x0 / B0[7:0]	1	0x0 / B1 B0[15:0]	0x0 / B1 B0[15:0]
			2	0x1 / B1[7:0]	2	0x4 / B3 B2[15:0]	0x2 / B3 B2[15:0]
			3	0x2 / B2[7:0]			
			4	0x3 / B3[7:0]			
8	32	1	1	0x0 / B0[7:0]	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]
			2	0x1 / B1[7:0]			
			3	0x2 / B2[7:0]			
			4	0x3 / B3[7:0]			
16	8	4	1	0x0 / B1 B0[15:0]	1	0x0 / B0[7:0]	0x0 / B0[7:0]
			2	0x2 / B3 B2[15:0]	2	0x4 / B1[7:0]	0x1 / B1[7:0]
					3	0x8 / B2[7:0]	0x2 / B2[7:0]
					4	0xC / B3[7:0]	0x3 / B3[7:0]
16	16	2	1	0x0 / B1 B0[15:0]	1	0x0 / B1 B0[15:0]	0x0 / B1 B0[15:0]
			2	0x2 / B1 B0[15:0]	2	0x4 / B3 B2[15:0]	0x2 / B3 B2[15:0]
16	32	1	1	0x0 / B1 B0[15:0]	1	0x0 / B3 B2 B1 B0[31:0]	0x0 / B3 B2 B1 B0[31:0]
			2	0x2 / B3 B2[15:0]			
32	8	4	1	0x0 / B3 B2 B1 B0[31:0]	1	0x0 / B0[7:0]	0x0 / B0[7:0]
					2	0x4 / B1[7:0]	0x1 / B1[7:0]
					3	0x8 / B2[7:0]	0x2 / B2[7:0]
					4	0xC / B3[7:0]	0x3 / B3[7:0]
32	16	2	1	0x0 / B3 B2 B1 B0[31:0]	1	0x0 / B1 B0[15:0]	0x0 / B1 B0[15:0]
					2	0x4 / B3 B2[15:0]	0x2 / B3 B2[15:0]
32	32	1	1	0x0 / B3 B2 B1 B0 [31:0]	1	0x0 / B3 B2 B1 B0 [31:0]	0x0 / B3 B2 B1 B0 [31:0]

Рис. 8.17. Возможные сочетания разрядностей и форматов слов, считываемых из памяти источника и записываемых в память приемника в процессе ПДП-обмена данными в МК модельных рядов *STM32F20x* и *STM32F21x* [48], *STM32F405xx/07xx* и *STM32F415xx/17xx* [14]
(пояснения приведены в тексте)

На рис. 8.17:

- *AHB memory port width, AHB peripheral port width* – разрядность слов данных, передаваемых через порты контроллера ПДП *AHB memory port* и *AHB peripheral port* соответственно (см. рис. 8.14);
- *Number of data items to transfer* – число слов данных, подлежащих передаче;
- *Memory transfer number* – номер слова, передаваемого через *AHB memory port*;
- *Peripheral transfer number* – номер слова, передаваемого через *AHB peripheral port*;
- *Memory port address / byte lane, Peripheral port address / byte lane* – адреса слов данных (относительно адреса 1-го слова передаваемой последовательности) и форматы размещения их байтов в памяти устройств, подключенных к портам *AHB memory port* и *AHB peripheral port* соответственно;
- *PINCOS* – состояние бита *Peripheral increment offset size* в регистре конфигурации потока, *DMA_SxCR* (см. рис. 8.22);
- *Bx* – байт данных с номером *x*.

Примечание. Адреса и форматы слов данных, представленные на рис. 8.17, соответствуют единичному состоянию битов *MINC* и *PINC* в регистре конфигурации потока (после передачи каждого слова данных производится автоматическое увеличение адресов памяти устройств, подключенных как к порту *AHB memory port*, так и к порту *AHB peripheral port*. Увеличение адреса памяти первого из них осуществляется на число байт в словах, передаваемых через *AHB memory port*. Адрес памяти устройства, подключенного к *AHB peripheral port*, увеличивается на число байт в передаваемых через него словах при *PINCOS* = 0, и на 4 – при *PINCOS* = 1 (см. рис. 8.17).

Важно отметить, что, если разрядность слов данных, передаваемых через порт *AHB peripheral port*, меньше, чем передаваемых через порт *AHB memory port* (см., например, 4-ю строку таблицы, представленной на рис. 8.17), может возникнуть ситуация, при которой последнее слово данных передастся не полностью. Во избежание таких ситуаций, число слов данных, подлежащих передаче (*NDT*), задаваемое при конфигурировании потока, должно быть кратно отношению разрядности слов данных, передаваемых через порт *AHB memory port*, к разрядности слов

данных, передаваемых через порт *AHB peripheral port*. Например, при указанных разрядностях, равных 16-и и 8-и битам соответственно, значение *NDT* должно быть кратно 2-м, а равных соответственно 32-м и 8-и битам – 4-м.

8.3.2.12. Как указано ранее, каждый из потоков снабжен **буферной памятью** типа *FIFO*, предназначенной для промежуточного хранения данных в процессе их ПДП-передачи (см. рис. 8.14). Объем буфера *FIFO* – 16 байт (четыре 4-байтовых слова). На рис. 8.18 представлены его структура и примеры форматов последовательностей данных, записываемых в него / считываемых из него при различных сочетаниях разрядностей слов источника и приемника данных (см. также рис. 8.17).

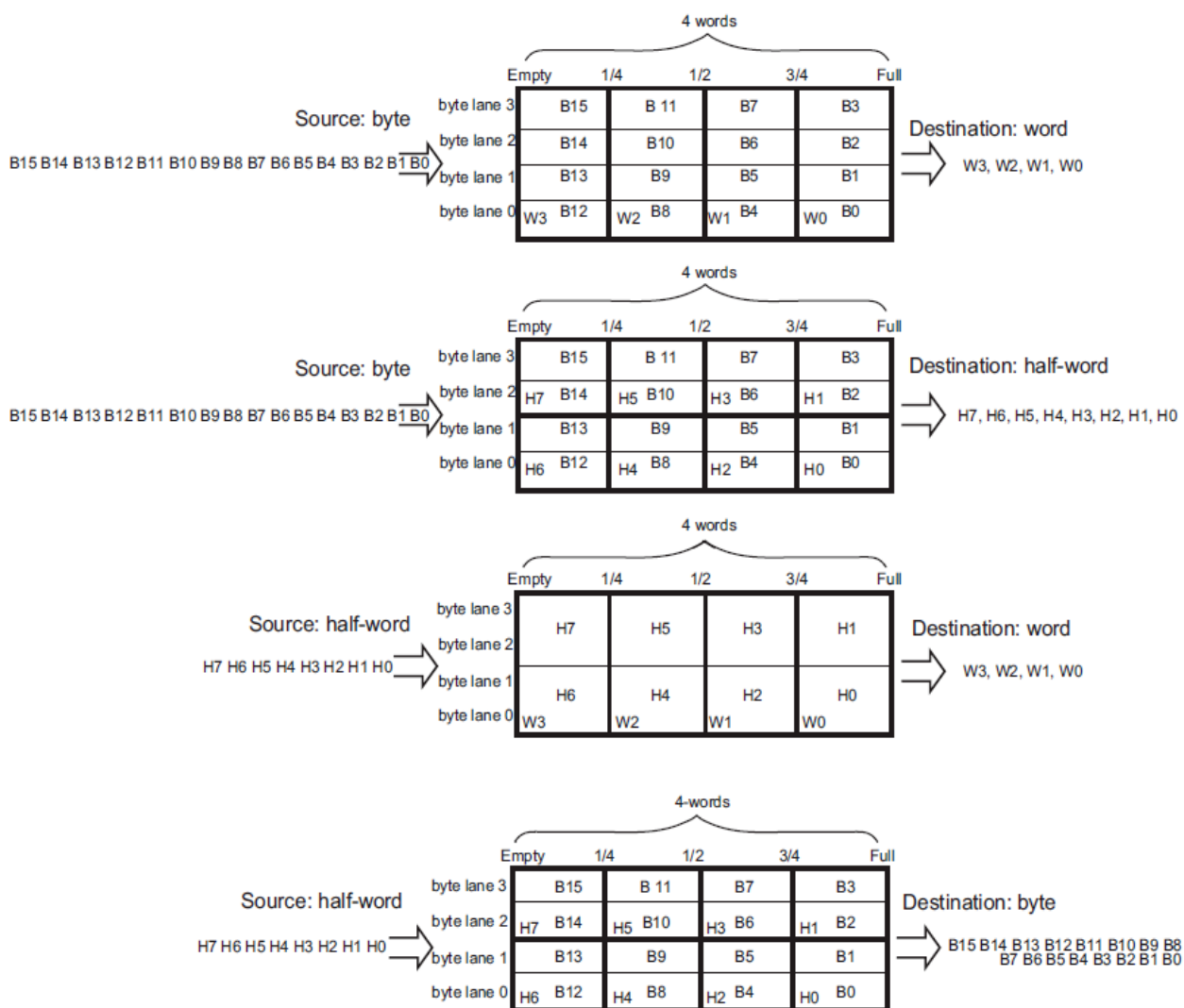
В потоках входных и выходных данных на рис. 8.18 слова данных (*bytes*, *half-words*, *words*) указаны в порядке их поступления в буфер или, соответственно, чтения из буфера по времени: крайнее справа – первое из поступивших / считанных, крайнее слева – последнее.

Разрешение использования буфера *FIFO* потока с номером *x* осуществляется установкой в единицу бита *DMDIS* в регистре управления буфером соответствующего потока, *DMA_SxFCR* (см. рис. 8.23). По умолчанию, данный бит находится в нулевом состоянии; если выбран режим обмена «память → память», данный бит устанавливается в единицу автоматически.

При использовании буфера *FIFO* необходимо установить **порог** его заполнения, т. е. степень заполнения буфера (см. рис. 8.18), по достижении которой данные должны быть считаны из него. Порог определяется соотношением быстродействия источника и приемника данных, и задается таким образом, чтобы не допустить переполнения буфера. Значение порога назначается битовым полем *FTH[1:0]* (*FIFO threshold selection*) регистра *DMA_SxFCR* (см. рис. 8.23). Содержимому данного поля, равному 00, 01, 10 и 11 соответствуют значения порогового уровня заполнения буфера *FIFO*, равные 1/4, 1/2, 3/4 и 4/4 (*Full*). При этом:

- в режиме обмена «ПУ → память» загрузка очередного слова данных в буфер происходит по каждому запросу от ПУ; по

достижении порогового уровня заполнения буфера его содержимое загружается в память;



Source – здесь: разрядность слова источника данных

Destination – здесь: разрядность слова приемника данных

byte lane x – байтовый тракт *x* (фактически – байт с позицией *x* в пределах слова, записанного в буфер *FIFO*)

half-word – слово данных разрядностью 16 бит

word – слово данных разрядностью 32 бита

Bx, *Hx*, *Wx* – соответственно байт, полуслово и слово с номером *x*
Empty (пуст), *1/4*, *1/2*, *3/4*, *Full* (полон) – степени заполнения буфера

Рис. 8.18. Буфер *FIFO* ПДП-потока МК модельных рядов *STM32F20x* и *STM32F21x* [48], *STM32F405xx/07xx* и *STM32F415xx/17xx* [14]

- в режиме обмена «память → ПУ» по каждому запросу от ПУ в его память считывается содержимое буфера, который, в свою

очередь, загружается очередными данными из памяти до достижения порогового уровня;

- в режиме обмена «память → память» заполнение буфера начинается по запуску сеанса ПДП (установкой бита разрешения, *EN*, в регистре конфигурации потока); по достижении порогового уровня содержимое буфера записывается по адресам назначения, а буфер загружается очередными данными из области памяти, служащей источником, до достижения порогового уровня и т. д.

8.3.2.13. Контроллерами ПДП рассматриваемых модельных рядов МК поддерживаются **одиночный** и **пакетный** режим обмена данными (называемые соответственно *single* и *burst transfers*). В первом из них по каждому запросу на ПДП осуществляется передача одного слова данных, во втором – пакета данных из 4-х, 8-и или 16-и слов.

Выбор одиночного или пакетного режима обмена, а также количества слов в пакете производится при конфигурировании потока. Если источником данных является устройство, подключенное к порту *AHB peripheral port*, выбор осуществляется полем *PBURST[1:0]* регистра конфигурации потока, а если устройство, подключенное к порту *AHB memory port* – полем *MBURST[1:0]* того же регистра (см. рис. 8.22). Значению 00 данных полей (устанавливаемому по умолчанию) соответствует одиночный режим обмена, а значениям 01, 10 и 11 – пакетный режим с размером пакета, равным 4, 8 и 16 слов соответственно. Разрядность слов при этом задается соответственно битовыми полями *PSIZE[1:0]* и *MSIZE[1:0]* регистра конфигурации потока (см. подпункт 8.3.2.11).

Указываемое в регистре *DMA_SxNDTR* число слов данных, подлежащих передаче в течение сеанса, **должно быть кратно** произведению размера пакета (в словах) на число байт в слове.

Пакетный обмен данными возможен **только** при использовании буфера *FIFO*. Если выбран режим прямой передачи, *Direct mode* (см. подпункт 8.3.2.10), битовые поля *PBURST[1:0]* и *MBURST[1:0]* автоматически сбрасываются в нулевое состояние, т. е. устанавливается одиночный режим обмена.

При вариантах передачи «ПУ → память» и «память → ПУ» возможен пакетный обмен данными в режиме **двойного буфера**.

Передаваемые пакеты данных загружаются в буфер *FIFO*. При этом выбор порога его заполнения (см. подпункт 8.3.2.12), а также размер пакета (в словах) и число байт в слове должны выбираться с учетом следующих **ограничений**:

- произведение размера пакета (в словах) на число байт в слове не должно превышать объем буфера *FIFO*, во избежание его переполнения;
- порог заполнения буфера должен соответствовать целому числу пакетов.

Допустимые сочетания размера пакета в словах, их разрядностей и порога заполнения буфера *FIFO* представлены в табл. 8.1.

Задание адресов *DMA_SxPA*, *DMA_SxM0A* и *DMA_SxM1A* должно осуществляться с учетом разрядности слов данных и форматов их размещения в памяти устройств, подключенных к портам *АНВ memory port* и *АНВ peripheral port* контроллера ПДП (см. рис. 8.17).

Таблица 8.1

Допустимые сочетания размера пакета, разрядности слов и порога заполнения буфера FIFO контроллера ПДП МК модельных рядов STM32F20x и STM32F21x [48], STM32F405xx/07xx и STM32F415xx/17xx [14] при пакетном режиме обмена

Разрядность слова	Порог заполнения буфера	Число пакетов / слов / байт в буфере при размере пакета:		
		4 слова	8 слов	16 слов
8 бит (1 байт)	1/4	1 / 4 / 4	Н / д ¹⁾	Н / д ¹⁾
	1/2	2 / 8 / 8	1 / 8 / 8	
	3/4	3 / 12 / 12	Н / д ¹⁾	
	Full	4 / 16 / 16	2 / 16 / 16	1 / 16 / 16
16 бит (2 байта)	1/4	Н / д ¹⁾	Н / д ¹⁾	Н / д ¹⁾
	1/2	1 / 4 / 8		
	3/4	Н / д ¹⁾		
	Full	2 / 8 / 16	1 / 8 / 16	
32 бита (4 байт)	1/4	Н / д ¹⁾	Н / д ¹⁾	
	1/2			
	3/4			
	Full	1 / 4 / 16		
¹⁾ Н / д – не допустимо				

8.3.2.14. В табл. 8.2 сведены возможные и недопустимые сочетания режимов работы контроллеров ПДП модельных рядов *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* МК [14, 48].

Таблица 8.2

Возможные и недопустимые сочетания режимов работы контроллеров ПДП модельных рядов STM32F20x, STM32F21x, STM32F405xx/07xx и STM32F415xx/17xx МК [14, 48].

Вариант обмена	Flow controller	Циклический режим	Режим передачи	Direct mode	Режим двойного буфера
ПУ → память (AHB peripheral port → AHB memory port)	Контроллер ПДП	Д	Одиночный	Д	Д
			Пакетный	Н / д	
	ПУ	Н / д	Одиночный	Д	Н / д
			Пакетный	Н / д	
Память → ПУ (AHB memory port → AHB peripheral port)	Контроллер ПДП	Д	Одиночный	Д	Д
			Пакетный	Н / д	
	ПУ	Н / д	Одиночный	Д	Н / д
			Пакетный	Н / д	
Память → память (AHB peripheral port → AHB memory port)	Контроллер ПДП	Н / д	Одиночный	Н / д	Н / д
			Пакетный		
Примечание: Д – допустимо, Н / д – не допустимо.					

8.3.2.15. Как и в МК подсемейства *ARM Cortex-M0*, контроллеры ПДП рассматриваемых модельных рядов МК во время ПДП-сеанса осуществляют мониторинг отсутствия / наличия ошибок обмена

данными. В архитектуре подсистемы ПДП модельных рядов *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* различаются три вида ошибок ПДП-обмена:

- ошибка передачи по потоку (*Transfer Error*);
- ошибка чтения / записи буфера *FIFO* (*FIFO Error*);
- ошибка обмена в режиме *Direct mode* (*Direct Mode Error*).

При этом:

- источником ошибки передачи по потоку может быть сбой на шине при ПДП-обмене, в т. ч. обращение по физически не существующему адресу, а также попытка изменения содержимого «активного» регистра базового адреса памяти в режиме двойного буфера (см. подпункт 8.3.2.9);

- ошибка чтения / записи буфера *FIFO* возникает при его перегрузке (*overrun*) или недогрузке (*underrun*), а также при несоответствии порога заполнения буфера целому числу пакетов (*FIFO level error*, см. табл. 8.1);

- ошибка обмена в режиме *Direct mode* может возникнуть при направлении передачи «ПУ → память» и нулевом состоянии бита *MINC* в регистре конфигурации потока (т. е. при записи данных в процессе обмена по одному и тому же адресу памяти); условие возникновения ошибки – поступление очередного запроса от ПУ на ПДП-передачу до того, как предыдущее слово данных загрузилось по адресу назначения.

По обнаружении какой-либо из вышеперечисленных ошибок устанавливается в единицу бит ее признака в регистре статуса контроллера ПДП и вырабатывается запрос на прерывание по соответствующей ошибке, если оно разрешено (см. подпункт 8.3.2.15). При возникновении ошибки передачи по потоку (*Transfer Error*) или по событию *FIFO level error* работа вызвавшего ее потока автоматически останавливается на аппаратном уровне, сбросом бита разрешения (*EN*) в регистре конфигурации потока. При ошибках обмена в режиме *Direct Mode* или по событиям *FIFO overrun / underrun* работа ПДП-потока не останавливается; обработка ошибки должна производиться на программном уровне.

8.3.2.16. Аналогично каналам контроллеров ПДП МК подсемейства *ARM Cortex-M0*, каждый из потоков контроллеров

ПДП МК модельных рядов *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* может быть источником **запросов на прерывания**. При этом состав событий, вызывающих прерывания (при условии, что они разрешены), шире, чем у МК подсемейства *ARM Cortex-M0*, и, кроме прерываний по событиям:

- завершение сеанса передачи массива данных по потоку (*Transfer Complete Interrupt, TCI*);
 - завершение передачи половины массива данных по потоку (*Half Transfer Interrupt, HTI*);
 - ошибка передачи по потоку (*Transfer Error Interrupt, TEI*);
- (см. подпункт 8.3.1.10) источниками прерываний также могут быть:
- ошибка чтения / записи буфера *FIFO* (*FIFO Error Interrupt, FEI*);
 - ошибка обмена в режиме *Direct mode* (*Direct Mode Error Interrupt*).

Как и в МК подсемейства *ARM Cortex-M0*, в таблице векторов прерываний не выделяется специальных позиций под прерывания по каждому из данных событий; отдельные позиции назначаются только каждому из потоков контроллера ПДП, под прерывания по какому-либо из вышеперечисленных событий в соответствующем потоке (см. рис. 8.19).

Position	Priority	Type of priority	Acronym	Description	Address
11	18	settable	DMA1_Stream0	DMA1 Stream0 global interrupt	0x0000_006C
12	19	settable	DMA1_Stream1	DMA1 Stream1 global interrupt	0x0000_0070
13	20	settable	DMA1_Stream2	DMA1 Stream2 global interrupt	0x0000_0074
14	21	settable	DMA1_Stream3	DMA1 Stream3 global interrupt	0x0000_0078
15	22	settable	DMA1_Stream4	DMA1 Stream4 global interrupt	0x0000_007C
16	23	settable	DMA1_Stream5	DMA1 Stream5 global interrupt	0x0000_0080
17	24	settable	DMA1_Stream6	DMA1 Stream6 global interrupt	0x0000_0084

Рис. 8.19. Фрагмент таблицы векторов прерываний МК модельных рядов *STM32F20x* и *STM32F21x* [48]

Для возможности идентификации конкретного события, вызвавшего прерывание от контроллера ПДП, в его составе имеются регистры признаков событий, которые могут быть вызывать прерывания. Ввиду того, что по каждому из потоков прерывание может быть вызвано 5-ю различными событиями (а не 3-мя, как в МК подсемейства *ARM Cortex-M0*), под признаки данных событий выделено по два регистра в каждом контроллере ПДП: *DMA low interrupt status register (DMA_LISR)* и *DMA high interrupt status register (DMA_HISR)*. Их форматы представлены на рис. 8.20. На нем: *TCIFx*, *HTIFx*, *TEIFx*, *DMEIFx*, *FEIFx* – биты признаков возникновения в потоке с номером *x* событий «*Transfer Complete*», «*Half Transfer*», «*Transfer Error*», «*Direct Mode Error*» и «*FIFO Error*» соответственно (активное состояние данных битов – единичное).

Разрешение прерываний по первым 4-м из вышеперечисленных событий осуществляется установкой в единицу битов *TCIE*, *HTIE*, *TEIE* и *DMEIE* в регистре конфигурации соответствующего потока (см. рис. 8.22), а прерывания по событию «*FIFO Error*» - битом *FEIE* в регистре *DMA stream x FIFO control register (DMA_SxFCR)* (см. рис. 8.23).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF3	HTIF3	TEIF3	DMEIF3	Reserved	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Reserved	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TCIF1	HTIF1	TEIF1	DMEIF1	Reserved	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Reserved	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

a)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF7	HTIF7	TEIF7	DMEIF7	Reserved	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Reserved	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TCIF5	HTIF5	TEIF5	DMEIF5	Reserved	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Reserved	FEIF4
				r	r	r	r		r	r	r	r	r		r

б)

Рис. 8.20. Форматы регистров *DMA_LISR* (а) и *DMA_HISR* (б) модельных рядов МК *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* [48] и *STM32F415xx/17xx* [14] (см. пояснения в тексте)

При запрете прерывания по какому-либо событию, его возникновение может отслеживаться программным опросом соответствующего бита регистра *DMA_LISR* или *DMA_HISR* (что в большинстве практических случаев не рационально).

Как и в МК подсемейства *ARM Cortex-M0*, биты признаков событий – потенциальных источников прерываний от контроллера ПДП доступны только для чтения, а их сброс осуществляется записью единицы в одноименный бит регистра *DMA low interrupt flag clear register (DMA_LIFCR)* или *DMA high interrupt flag clear register (DMA_HIFCR)* (см. рис. 8.21).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Reserved	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Reserved	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Reserved	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Reserved	CFEIF0
				w	w	w	w		w	w	w	w	w		w

а)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Reserved	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Reserved	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Reserved	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Reserved	CFEIF4
				w	w	w	w		w	w	w	w	w		w

б)

Рис. 8.21. Форматы регистров *DMA_LIFCR* (а) и *DMA_HIFCR* (б) модельных рядов МК *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* [14, 48]
(см. также рис. 8.20)

Обслуживание прерывания по некоторому потоку *x* контроллера ПДП, в общем случае, рационально осуществлять в следующем порядке:

- опросом битов *TCIF_x*, *HTIF_x*, *TEIF_x*, *DMEIF_x*, *FEIF_x* соответствующего потока идентифицировать событие, вызвавшее прерывание;
- после идентификации события, вызвавшего прерывание, выполнить сброс находящихся в активном состоянии битов регистров *DMA_LISR* и *DMA_HISR*, для предотвращения повторных

запусков подпрограммы обслуживания прерывания после выхода из нее (см. подпункт 7.3.2.16);

- перейти к процедуре обслуживания конкретного события в конкретном канале (например, считывания массива данных из ПД по событию *Transfer Complete*).

Примечание. Перед установкой в единицу бита разрешения работы потока x (EN) в регистре его конфигурации биты $TCIFx$, $HTIFx$, $TEIFx$, $DMEIFx$ и $FEIFx$ должны быть сброшены в нулевое состояние. Если какой-либо из этих битов будет в единичном состоянии при установке в единицу бита EN , то непосредственно после нее произойдет прерывание по событию, признаком которого он является [14, 48].

8.3.2.17. Приоритетное обслуживание запросов на ПДП в МК модельных рядов *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* осуществляется аналогично приоритезации запросов на ПДП в МК подсемейства *ARM Cortex-M0* (см. подпункт 8.3.1.4):

- каждому из потоков программно назначается один из 4-х уровней приоритетности («Очень высокий», «Высокий», «Средний» или «Низкий»); назначение осуществляется присвоением значений 11, 10, 01 или 00 соответственно битовому полю $PL[1:0]$ (*Priority level*) регистра конфигурации потока (см. рис. 8.22);

- в пределах группы потоков с одинаковым уровнем приоритетности, чем меньше номер потока, тем выше его приоритет.

Примечание. Поток, сконфигурированному на режим работы *Direct mode*, рекомендуется присваивать наивысший приоритет, во избежание потерь данных, так как в данном режиме промежуточное хранение данных в буфере *FIFO* не осуществляется.

8.3.2.18. В состав программно-доступных регистров контроллера ПДП входят:

- DMA_LISR и DMA_HISR (см. рис. 8.20);
- DMA_LIFCR и DMA_HIFCR (см. рис. 8.21);
- регистры потоков, перечень которых представлен в подпункте 8.3.2.5; форматы регистра конфигурации потока (DMA_SxCR) и регистра управления буфером *FIFO* потока (DMA_SxFCR) представлены на рис. 8.22 и 8.23 соответственно.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Reserved	CT	DBM	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис. 8.22. Формат регистра конфигурации потока с номером x (DMA_SxCR) МК модельных рядов МК $STM32F20x$, $STM32F21x$, $STM32F405xx/07xx$ и $STM32F415xx/17xx$ [14, 48]
(см. пояснения в тексте)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
								rw		r	r	r	rw	rw	rw

Рис. 8.23. Формат регистра управления буфером *FIFO* потока с номером x (DMA_SxFCR) МК модельных рядов МК $STM32F20x$, $STM32F21x$, $STM32F405xx/07xx$ и $STM32F415xx/17xx$ [14, 48]
(см. пояснения в тексте)

Назначение битов (битовых полей) регистра DMA_SxCR :

- $CHSEL[2:0]$ – выбор источника запросов по потоку (см. рис. 8.15);

- $PBURST[1:0]$, $MBURST[1:0]$ – выбор числа слов в пакете (см. подпункт 8.3.2.13);

- CT – указатель «активного» адресного регистра в режиме двойного буфера (см. подпункт 8.3.2.9);

- DBM – разрешение / запрет режима двойного буфера (см. подпункт 8.3.2.9);

- $PL[1:0]$ – выбор уровня приоритетности потока (см. подпункт 8.3.2.17);

- $PINCOS$ – бит, определяющий приращение адреса памяти устройства, подключенного к порту *AHB peripheral port*, при $PINC = 1$; если $PINCOS = 0$, приращение равно числу байт в словах данных, передаваемых через *AHB peripheral port*; если $PINCOS = 1$, приращение равно 4-м, независимо от разрядности указанных слов (см. рис. 8.17 и пояснения к нему);

- *MSIZE[1:0]* и *PSIZE[1:0]* – битовые поля, определяющие разрядность слов данных, передаваемых через порты *AHB memory port* и *AHB peripheral port* соответственно (см. подпункт 8.3.2.11);
 - *MINC* и *PINC* – биты разрешения / запрета автоматического увеличения адресов памяти устройств, подключенных к портам *AHB memory port* и *AHB peripheral port*, после очередного цикла ПДП (см. подпункт 8.3.2.7, а также пояснения к рис. 8.17);
 - *CIRC* – разрешение / запрет циклического режима обмена (см. подпункт 8.3.2.8);
 - *DIR[1:0]* – выбор направления передачи данных (см. подпункт 8.3.2.7);
 - *PFCTRL* – назначение устройства, выполняющего функцию *Flow controller* по соответствующему потоку (см. подпункт 8.3.2.6);
 - *TCIE*, *HTIE*, *TEIE* и *DMEIE* – разрешение / запрет прерываний по событиям «*Transfer Complete*», «*Half Transfer*», «*Transfer Error*», «*Direct Mode Error*» соответственно (см. подпункт 8.3.2.16);
 - *EN* – разрешение / запрет работы потока.
- Назначение битов (битовых полей) регистра *DMA_SxFCR* (см. рис. 8.23):
- *FEIE* – разрешение / запрет прерываний по событию «*FIFO Error*» (см. подпункт 8.3.2.16);
 - *FS[2:0]* – статус буфера (степень его заполнения);
 - *DMDIS* – запрет / разрешение режима *Direct mode* (см. подпункт 8.3.2.10);
 - *FTH[1:0]* – задание порога заполнения буфера (см. подпункт 8.3.2.12).

Примечание. Изменение содержимого программно-доступных регистров каждого из потоков должно производиться только при нулевом состоянии бита *EN* в регистре конфигурации. При единичном состоянии данного бита может производиться только изменение содержимого регистров *DMA_SxM0AR* и *DMA_SxM1AR* в режиме двойного буфера (см. подпункт 8.3.2.9). Работа потока может быть разрешена только после выполнения всех процедур его конфигурирования (задания базовых адресов памяти источника и приемника данных; числа слов, подлежащего передаче за ПДП-сеанс, режимов работы потока и т. п.) [14, 48]. Также необходимо

отметить, что, если необходимо остановить ПУ, запросы от которого обслуживаются некоторым потоком ПДП, остановка работы данного ПУ (т. е. сброс бита ее разрешения) может быть произведен только после запрета работы потока, т. е. сброса бита *EN* в регистре его конфигурации и проверки того, что он действительно находится в нулевом состоянии [14, 48].

8.3.2.19. С не рассмотренными в подпунктах 8.3.2.4 – 8.3.2.18 деталями архитектуры подсистемы ПДП МК модельных рядов *STM32F20x*, *STM32F21x*, *STM32F405xx/07xx* и *STM32F415xx/17xx* можно ознакомиться по источникам [14] и [48].

8.4 Примеры программирования подсистемы ПДП

Далее представлены примеры программных модулей конфигурирования и запуска контроллеров ПДП МК семейства *ARM Cortex-Mx*, разработанные для решения типовых задач использования обмена данными в режиме ПДП. Уровень программирования – *CMSIS*, представление операндов – числовое (см. подпункт 2.6.1.14 и Приложение В).

8.4.1. Пример 1. Запись в память в режиме ПДП результатов преобразования в режиме **сканирования** по 8-и **регулярным** каналам (с 0-го по 7-й) встроенного АЦП МК модели *STM32F103C8* (подсемейство *ARM Cortex-M3*, модельный ряд *STM32F10xxx* [13]). Сканирование должно производиться с периодом 100 мкс.

Пояснение. Встроенные АЦП МК семейства *ARM Cortex-Mx*, как и большинства других семейств МК, являются многоканальными. При этом их каналам, в отличие от, например, каналов АЦП МК семейства *AVR*, может быть присвоен статус **регулярных** или **инжектированных**. Например, у АЦП МК модели *STM32F103C8* имеется 10 каналов; из них до 4-х могут быть инжектированными. По каждому из инжектированных каналов имеется отдельный регистр данных (т. е. результата преобразования), а по группе регулярных каналов регистр данных только один, общий для всех каналов. При этом, например, у МК модельного ряда *STM32F10xxx* [13] и некоторых других, при

многоканальном преобразовании по регулярным каналам в режиме их **сканирования** единственным способом избежания потерь результатов является их запись в память в режиме ПДП [13], поскольку в режиме сканирования:

- запрос на ПДП вырабатывается по завершении преобразования по **каждому** из каналов;

- установка в единицу признака завершения преобразований (*EOC*) в регистре статуса АЦП и формирование запроса на прерывание по данному событию происходят только по завершении преобразований по **всем** каналам, и поэтому не могут служить признаками завершения преобразования по отдельным каналам.

Подробнее – см. подпункт 10.3.11.8.

В рассматриваемом примере используется ПДП-обмен данными в циклическом режиме (см. подпункт 8.3.1.8). Запуск цикла сканирования каналов (*ADC triggering*) осуществляется по перепаду из 0 в 1 сигнала, формируемого на выходе 4-го канала 4-го таймера МК. После запуска сканирования последовательно выполняется аналого-цифровое преобразование по всем каналам, выбранным при конфигурировании АЦП (в рассматриваемом примере – по каналам с 0-го по 7-й). По выполнении преобразования по очередному каналу:

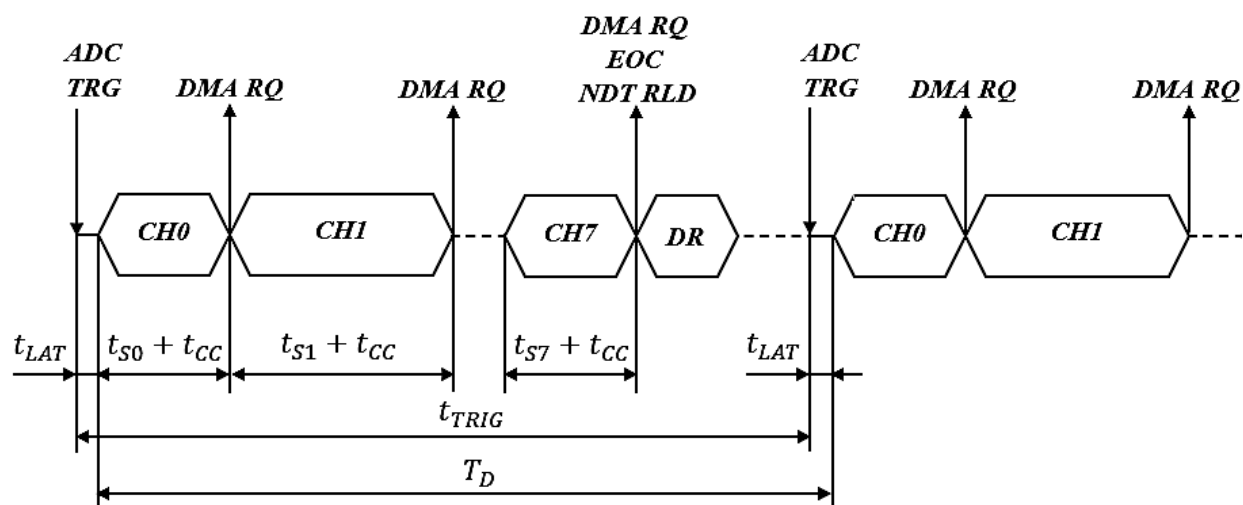
- его результат записывается в регистр данных, общий для всех каналов

- АЦП вырабатывает запрос на ПДП, по которому результат преобразования записывается в ПД;

- происходит автоматический переход к преобразованию по следующему каналу.

После выполнения преобразования по всем каналам устанавливается в единицу признак завершения преобразований (*EOC*) в регистре статуса АЦП, и ЦП производит считывание из ПД результатов преобразований по каналам. Поскольку используется циклический режим ПДП-обмена, в регистр счетчика слов автоматически загружается его начальное значение, равное числу слов, передаваемых за сеанс ПДП (в рассматриваемом примере – 8-и). По очередному сигналу запуска с 4-го таймера начинается новый цикл сканирования каналов и новый сеанс ПДП.

Временные диаграммы, поясняющие вышеописанный процесс сканирования регулярных каналов АЦП, представлены на рис. 8.24.



- ADC TRG* – событие, инициирующее запуск АЦП;
CH0, CH1, ..., CH7 – выборка и преобразование сигналов по каналам с соответствующими номерами;
DMA RQ – запросы от АЦП на ПДП;
EOC – установка признака окончания преобразований;
NDT RLD – перезагрузка счетчика слов;
DR – считывание данных (результатов преобразований по каналам) из ПД центральным процессором;
 t_{LAT} – задержка момента запуска АЦП относительно наступления события *ADC TRG* (см. подпункт 10.3.6.6)
 $t_{S0}, t_{S1}, \dots, t_{S7}$ – время выборки сигнала по соответствующему каналу (см. подпункты 10.3.4.5 и 10.3.11.18);
 t_{CC} – время преобразования в код сигнала, полученного в результате выборки;
 t_{TRIG} – период наступления событий *ADC TRG*
 T_D – период дискретизации сигналов, см. рис. 10.18 (одинаков по всем каналам)

Рис. 8.24. Временные диаграммы, поясняющие процесс сканирования регулярных каналов АЦП
(см. также подпункт 10.3.12.3)

Текст программного модуля (с комментариями), реализующего вышеописанный пример использования обмена в режиме ПДП, представлен ниже.

```

//Подключение файла описания МК модельного ряда stm32f10x
#include "stm32f10x.h"
/////////////////////////////////////////////////////////////////
//Объявление переменной – указателя адреса размещения в ПД результатов
преобразования
char *ad;
//Объявление массива результатов преобразований по каналам
int x[8];
//Объявление индексной переменной
int i;
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
//Основная программа
int main(void)
{
/////////////////////////////////////////////////////////////////
/*
Конфигурирование, в соответствии с подразделом 7.3 руководства [13],
подсистемы синхронизации МК.
Объявление генератора HSE (см. подпункт 4.4.2.1) источником тактирования
МК; в рассматриваемом примере частота HSE равна 8 МГц.
Биты 14 и 15 CFGR также задают тактовую частоту АЦП [13]. По умолчанию
(при их нулевых значениях) она равна тактовой частоте МК, поделенной на 2,
в данном примере – 4 МГц, длительность 1 такта – 0,25 мкс.
Программный фрагмент конфигурирования подсистемы синхронизации
полностью аналогичен приведенному в подпункте 4.4.2.4.
*/
RCC->CR |=0x00010000;
while(((RCC->CR)&(0x00020000))==0x00000000);
RCC->CFGR &= 0xFFFFF0FC;
RCC->CFGR |= 0x00000001;
while(((RCC->CFGR)&(0x0000000C))!=0x00000004);
RCC->CR = (RCC->CR)&(0xFFFFF0FE);
/////////////////////////////////////////////////////////////////
/*
Конфигурирование, в соответствии с подразделом 9.2 руководства [13],
выводов порта A с 0-го по 7-й, задействованных под входы каналов АЦП с
теми же номерами [10].
*/

```

Разрешение тактирования порта A и модуля управления альтернативными функциями портов ($AFIO$)

*/

$$RCC \rightarrow APB2ENR \mid = 0x00000005;$$

/ *

Очистка установок «по умолчанию» битов конфигурации выводов порта А с 0-го по 7-й. Очистка устанавливает их в режим «Аналоговый вход»

*/

```
GPIOA -> CRL = (GPIOA -> CRL)&(0x00000000);
```

////////////////////////////////////

/ *

Конфигурирование и запуск, в соответствии с подразделом 15.4 руководства [13], 4-го таймера, сигнал 4-го канала которого служит для запуска очередного цикла сканирования регулярных каналов АЦП1 (см. рис. 8.24). Запуск осуществляется перепадом из 0 в 1 данного сигнала.

*/

```
//Разрешение тактирования 4-го таймера
```

$$RCC \rightarrow APB1ENR = 0x00000004;$$

/ *

Задание параметров конфигурации 4-го таймера:

- источник тактирования таймера – ГТИ МК (*HSE* частотой 8 МГц, см. выше);
- коэффициент деления предделителя тактовой частоты – 1;
- источник генерируемых импульсов – 4-й канал таймера;
- режим генерации импульсов – асимметричная ШИМ, без инверсии сигнала;
- период генерируемых импульсов и, соответственно, период запуска последовательности преобразований по каналам АЦП и период дискретизации по каждому из каналов – 100 мкс;
- длительность импульсов – 50 мкс.

Подробнее – см. подпункты 9.3.3.2, 9.3.3.4 и подраздел 15.4 руководства [13]

*/

$TIM4 \rightarrow CR1 \mid = 0x0080;$

```
TIM4->CCMR2 |= 0x6000;
```

TIM4->PSC = 0x0000;

$$TIM4 \rightarrow ARR = 800;$$

$TIM4 \rightarrow CCR4 = 400;$

```
TIM4->CCER |= 0x1000;
```

/*

Разрешение работы 4-го таймера установкой в единицу бита *CEN* (*Counter Enable*) его 1-го регистра управления

*/

$TIM4 \rightarrow CR1 \mid = 0x0001;$

/ *

*/

/ *

*/

/ *

*/

/ *

*/

/ *

Задание времени выборки сигнала по каналам (см. подпункт 10.3.11.5). Младшие 3 бита регистра *SMPR2* задают время выборки по 0-му каналу, следующие 3 бита – по 1-му и т. д. По всем каналам, кроме 1-го, значения данных битовых полей заданы равными 000 (по умолчанию), что соответствует времени выборки, равному 1,5 периодам синхросигнала АЦП [13]. Значение битового поля, определяющее время выборки по 1-му каналу, задано равным 001, что соответствует 7,5 периодам синхросигнала АЦП [13], и обусловлено особенностями источника сигнала, подключенного к 1-му каналу (см. подпункт 10.3.4.5). При этом в рассматриваемом примере:

- период синхросигнала АЦП равен 0,25 мкс;
- время выборки по каждому из каналов, кроме 1-го – 0,375 мкс; по 1-му каналу – 1,875 мкс;
- время аналого-цифрового преобразования (t_{CC} на рис. 8.24) – 12,5 периодов синхросигнала АЦП [10], т. е. 3,125 мкс;
- сумма времени выборки и преобразования (см. рис. 8.24) по каждому из каналов, кроме 1-го равна 3,5 мкс, по 1-му каналу – 5 мкс;
- суммарное время сканирования всех каналов – 29,5 мкс;
- длительность интервала времени от окончания текущей до запуска следующей процедуры сканирования, выделяемого для считывания и обработки результатов преобразований, равна 100 мкс минус 29,5 мкс, т. е. 70,5 мкс.

ADC1->SMPR2 = 0x00000008;

```
ADC1->CR2 |= 0x00000100;
```

```
ADC1->CR2 |= 0x00000001;
```

/ *

*/

```
//Разрешение тактирования DMA1
```

```
//Конфигурирование 1-го канала DMA1, обслуживающего запросы от АЦП1.
```

Задание адреса регистра данных ПУ. В нашем примере это регистр данных АЦП. Его адрес равен сумме базового адреса регистров АЦП1, *0x40012400* [10], и равного *0x4C* смещения адреса регистра данных АЦП1 относительно базового адреса, [13].

DMA1 Channel1->CPAR = 0x4001244C;

Задание начального адреса области резидентного статического ОЗУ, выделяемой под результаты преобразования по каналам. Адрес отведен не в начальной области ОЗУ, т. к. она может быть использована при компиляции программы (например, под объявляемые переменные).

DMA1 Channell->CMAR = 0x20000400;

```
DMA1_Channel1->CNDTR = 0x00000008;
```

```
/*
```

Программирование регистра конфигурации 1-го канала *DMA1*, формат которого полностью идентичен представленному на рис. 8.10:

- задание направления передачи ПУ → память (бит *DIR* равен 0);
- бит *MINC* равен 1, а бит *PINC* – 0, что соответствует увеличению адреса ОЗУ на разрядность (в байтах) передаваемого слова в каждом цикле ПДП, при неизменном адресе регистра данных ПУ;
- задание циклического режима обмена установкой в единицу бита *CIRC*;
- задание разрядностей считываемых из ПУ и записываемых в память слов, равными 2-м байтам, записью значения 01 в битовые поля *PSIZE[1:0]* и *MSIZE[1:0]* соответственно.

```
*/
```

```
DMA1_Channel1->CCR = 0x000005A0;
```

```
//Разрешение работы 1-го канала DMA1
```

```
DMA1_Channel1->CCR |= 0x00000001;
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
/*
```

Начальный сброс бита признака окончания преобразования (*EOC*) в регистре статуса АЦП1, в рассматриваемом примере данный бит является признаком завершения цикла сканирования каналов (см. рис. 8.24)

```
*/
```

```
ADC1->SR &= 0xFFFFFFF0;
```

```
/*
```

Начальный сброс признака завершения ПДП-сеанса (*TCIF*) по 1-му каналу *DMA1* (сброс битов регистра *IFCR* производится записью в них единицы [13]).

```
*/
```

```
DMA1->IFCR |= 0x00000002;
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//"Бесконечный" цикл
```

```
while (1)
```

```
{
```

```
//Ожидание установки бита EOC в регистре статуса АЦП1 (см. выше).
```

```
while(((ADC1->SR)&(0x00000002))!=0x00000000);
```

```
//Сброс бита EOC
```

```
ADC1->SR &= 0xFFFFFFF0;
```

```
//Сброс признака завершения ПДП-сеанса по 1-му каналу DMA1 (см. выше).
```

```
DMA1->IFCR |= 0x00000002;
```

```
/*
```

Установка указателя адреса ОЗУ на начальный адрес массива результатов аналого-цифрового преобразования по каналам

```
*/
```

```
ad = (char*)0x20000400;
```

```
/*
```

Считывание результатов преобразования по каналам с 0-го по 7-й и присвоение их значений переменным $x[0] \dots x[7]$ соответственно. После очередного считывания указатель адреса увеличивается **на 2**, поскольку разрядность результатов преобразования равна 2-м байтам.

```
*/
```

```
for(i=0;i<8;i++)
```

```
{
```

```
x[i]=*ad;
```

```
ad=ad + 2;
```

```
}
```

```
////////////////////////////////////////////////////////////////
```

```
//Подпрограмма обработки результатов преобразования
```

```
...
```

```
}
```

```
}
```

8.4.2. Пример 2. Реализация генератора гармонического сигнала некоторой формы (например, гармонического) способом прямого цифрового синтеза (*Direct Digital Synthesis, DDS*). Цифровые отсчеты сигнала хранятся в ПД и в моменты времени, задаваемые одним из таймеров, в режиме ПДП записываются в регистр данных встроенного ЦАП МК.

Модель МК – *STM32F407VG* [14, 29]. Используется 1-й ЦАП. Текст программного модуля с необходимыми комментариями представлен ниже.

```
//Подключение файла описания МК модельного ряда stm32f4xx
```

```
#include "stm32f4xx.h"
```

```
//Объявление функции конфигурирования 1-го ЦАП
```

```
void dac(void);
```

```
// Объявление функции конфигурирования таймера, тактирующего ЦАП
```

```
void tim_dac(void);
```

```
//Объявление функции конфигурирования контроллера ПДП
```

```
void dma(void);
```

```
//Объявление функции конфигурирования порта ввода / вывода
```

```
void gpio(void);
```

```
//Объявление переменной – указателя адреса размещения в ПД отсчетов генерируемого сигнала
```

```
char *ad;
```

```

//Объявление индексной переменной и массива отсчетов косинусоиды
int i;
unsigned char cs[20];
/////////////////////////////////////////////////////////////////
//Функция конфигурирования 1-го ЦАП (см. подраздел 14.5 руководства [14])
void dac(void)
{
//Разрешение тактирования 1-го ЦАП
RCC->APB1ENR |= 0x20000000;
/*
Программирование регистра управления 1-м и 2-м ЦАП.
Установка в единицу 2-го бита (TEN1). Единичное состояние данного бита
разрешает загрузку выходного регистра 1-го ЦАП по внешнему стробу.
Выбор источника внешнего строб-сигнала 1-го ЦАП. Выбор осуществляется
посредством битового поля TSEL1[2:0] (битов с 3-го по 5-й). В данном
примере в него записан код 100, что соответствует выбору выходного 2-го
таймера в качестве источника строб-сигнала.
*/
DAC->CR = 0x00000024;
/*
Разрешение формирования 1-м ЦАП запросов на ПДП установкой в единицу
12-го бита (DMAEN1) регистра управления. Запросы генерируются по фронту
строб-сигнала ЦАП. Если ЦАП работает не в режиме внешнего
стробирования, формирование запросов на ПДП от ЦАП невозможно [14].
*/
DAC->CR |= 0x00001000;
/*
Разрешение работы 1-го ЦАП установкой в единицу 0-го бита (EN1) регистра
управления ЦАП.
*/
DAC->CR |= 0x00000001;
}
/////////////////////////////////////////////////////////////////
/*
Функция конфигурирования 2-го таймера, служащего источником строб-
сигнала 1-го ЦАП (см. подраздел 18.4 руководства [14]).
*/
void tim_dac(void)
{
//Разрешение тактирования 2-го таймера
RCC->APB1ENR |= 0x00000001;

```


начальной области ОЗУ, т. к. она может быть использована при компиляции программы (например, под объявляемые переменные).

*/

DMA1_Stream5->M0AR = 0x20000400;

/*

Задание числа слов данных, подлежащих передаче за сеанс ПДП, равного числу отсчетов генерируемого сигнала за период. В рассматриваемом примере оно равно 16-ти.

*/

DMA1_Stream5->NDTR = 0x10;

/*

Назначение номера канала, обслуживаемого по 5-му потоку ПДП. Номер канала задается битовым полем *CHSEL[2:0]* регистра конфигурации потока (см. рис. 8.15). В рассматриваемом примере содержимое данного битового поля установлено равным 111 в двоичной системе (7-ми в десятичной), т. к. запросы на ПДП от 1-го ЦАП поступают по 7-му каналу 5-го потока.

*/

DMA1_Stream5->CR |= 0x0E000000;

/*

Задание параметров ПДП-обмена, посредством регистра конфигурации 5-го потока. Формат данного регистра представлен на рис. 8.22. В рассматриваемом примере:

- значения битовых полей *PSIZE[1:0]* и *MSIZE[1:0]* равны 00, что соответствует разрядностям считываемых из памяти и записываемых в ПУ слов, равным одному байту
- бит *MINC* равен 1, а бит *PINC* – 0, что соответствует увеличению адреса ОЗУ на разрядность (в байтах) передаваемого слова в каждом цикле ПДП, при неизменном адресе регистра данных ПУ (1-го ЦАП);
- состояние бита *CIRC* – единичное, что соответствует циклическому режиму работы (после формирования очередного периода генерируемого сигнала автоматически начинается формирование следующего периода);
- *DIR[1:0] = 01*, что соответствует направлению передачи память → ПУ.

Все остальные биты находятся в состоянии «по умолчанию».

Буфер *FIFO* не используется, поэтому программирование регистра управления им (см. рис. 8.23) не производится, его содержимое оставлено «по умолчанию», в том числе состояние бита запрета режима *Direct mode (DMDIS)* – нулевое.

*/

DMA1_Stream5->CR |= 0x0000540;

//Разрешение работы 1-го контроллера ПДП (см. рис. 8.22)

DMA1_Stream5->CR |= 0x00000001;

}

```

//*****
/*
Конфигурирование порта A, 4-й вывод которого используется по
альтернативной функции «Выход 1-го ЦАП» (см. подраздел 8.4 руководства
[14])
*/
void gpio(void)
{
//Разрешение тактирования порта A
RCC->AHB1ENR |= 0x00000001;
//Задание режима работы «Аналоговый вывод» по 4-му выводу порта A
GPIOA->MODER |= 0x00000300;
}
//*****
//Основная программа
int main(void)
{
/*
Формирование массива отсчетов гармонического сигнала (косинусоиды со
смещением на величину амплитуды средним уровнем). Отсчеты
представлены 8-битовыми числами без знака.
*/
cs[0]=0xFF;
cs[1]=0xF5;
cs[2]=0xDA;
cs[3]=0xB1;
cs[4]=0x80;
cs[5]=0x4F;
cs[6]=0x26;
cs[7]=0x0B;
cs[8]=0x01;
cs[9]=0x0B;
cs[10]=0x26;
cs[11]=0x4F;
cs[12]=0x80;
cs[13]=0xB1;
cs[14]=0xDA;
cs[15]=0xF5;

```

```

/*
Запись массива отсчетов в резидентное статическое ОЗУ, начиная с адреса
0x20000400
*/
ad = (char*)0x20000400;
for(i=0;i<16;i++)
{
    *ad = cs[i];
    ad = ad + 1;
}
//Конфигурирование порта A
gpio();
//Конфигурирование 2-го таймера
tim_dac();
//Конфигурирование контроллера ПДП
dma();
//Конфигурирование ЦАП
dac();
/*
DDS-формирование косинусоиды со смещенным средним уровнем в
«бесконечном» цикле
*/
while(1)
{
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
}

```

Примечание. Предполагается, что тактирование МК осуществляется источником «по умолчанию» - генератором HSI частотой 16 МГц.

8.5 Выводы по разделу 8

8.5.1. Назначение подсистемы ПДП (*DMA*) – управление обменом массивами данных между памятью и ПУ, 2-мя ПУ или 2-мя областями памяти в режиме прямого доступа к памяти (*Direct Memory Access*). Использование ПДП при обмене одиночными словами не рационально.

8.5.2. Режим ПДП характеризуется следующими основными отличительными особенностями:

- сеанс обмена данными может быть инициирован как ПУ, так и ЦП;
- разрешение на сеанс выдает ЦП;
- собственно процедура обмена данными реализуется без участия ЦП, под управлением контроллера ПДП, который до сеанса обмена должен быть сконфигурирован ПО МК;
- все необходимые в процессе обмена операции (в том числе формирование и модификация адресов источника и приемника, сигналов управления записью и чтением и т. п.) реализуются контроллером ПДП на аппаратном уровне;
- по окончании сеанса обмена данными контроллер ПДП выставляет в активное состояние признак завершения сеанса, располагаемый в регистре статуса контроллера; как правило, при этом также вырабатывается запрос на прерывание по событию «Завершение передачи».

8.5.3. К преимуществам режима ПДП по сравнению с режимом обмена по прерыванию и с обменом, инициируемым ПО МК обычно относят:

- более эффективное использование ЦП за счет возможности выполнения им других задач во время сеанса передачи данных;
- более высокую скорость передачи массивов данных, за счет неучастия ЦП в данном процессе.

Первое из вышеперечисленных преимуществ является бесспорным. Выигрыш в скорости обмена данными, вопреки широко распространенному мнению, не всегда имеет место; его наличие или отсутствие, а также его степень зависят от архитектуры подсистемы ПДП и вычислительной системы в целом (см. пункты 8.1.4 – 8.1.7). Тем не менее, во многих практических случаях применение ПДП оправдано и при отсутствии существенного выигрыша в скорости передачи данных, что обусловлено факторами, перечисленными в пункте 8.1.8. В некоторых частных случаях без применения режима ПДП корректный обмен данными невозможен (см. пример в пункте 8.4.1).

8.5.4. Сеансу обмена данными в режиме ПДП предшествует относительно сложная подготовительная процедура, включающая в себя: прием центральным процессором запроса на сеанс ПДП, анализ уровня приоритета источника запроса, выделение канала обмена данными, передачу источнику запроса подтверждения о его выделении и т. п. Поэтому преимущества режима ПДП проявляются только, если длительность подготовительной процедуры существенно меньше, чем длительность собственно сеанса обмена данными. Это имеет место при обмене массивами данных, причем тем в большей степени, чем больше объем массива. Использование ПДП при обмене одиночными словами данных не рационально.

8.5.5. Возможными вариантами обмена данными в режиме ПДП являются:

- ПУ → память;
- память → ПУ;
- память → память;
- ПУ → ПУ.

8.5.6. Обмен данными в режиме ПДП осуществляется под управлением контроллера ПДП, являющегося центральным функциональным блоком подсистемы ПДП. «Классическая» структура и архитектура контроллера ПДП описаны в подразделе 8.2. Структурно-архитектурные решения контроллеров ПДП современных МК, в целом, аналогичны «классическим», и отличаются от них рядом второстепенных особенностей (см. подраздел 8.3).

8.5.7. При реализации обмена в режиме ПДП каждый канал или поток (см. рис. 8.2, 8.4 и 8.14) контроллера ПДП перед его использованием должен быть настроен на выполнение процедуры обмена данными, планируемой для реализации по данному каналу. К основным параметрам настройки относятся:

- направление обмена (ПУ → память, память → ПУ или память → память);
- базовые адреса памяти источника и приемника данных, начиная с которых данные считываются / записываются в сеансе связи по соответствующему каналу (потoku);

- значения, на которые изменяются указанные адреса после записи / чтения очередного слова данных;
- число слов данных, передаваемых во время сеанса связи по каналу;
- наличие / отсутствие циклического режима работы (см. подпункты 8.3.1.8 и 8.3.2.8).

Архитектура контроллеров ПДП современных МК также позволяет задавать разрядность и форматы слов данных источника и приемника (см. подпункты 8.3.1.7 и 8.3.2.11).

Перечисленные параметры указываются в регистрах конфигурирования каждого из каналов (поток), см. подпункты 8.3.1.11 и 8.3.2.18.

8.5.8. Архитектура подсистемы ПДП ряда подсемейств / модельных рядов современных МК предоставляет возможность использования пакетного режима передачи и буферной памяти в процессе обмена данными, для повышения его скорости (см. подпункты 8.3.2.12 и 8.3.2.13).

8.5.9. Архитектура подсистем ПДП современных МК допускает аппаратную и программную инициализацию обмена данных.

При аппаратной инициализации передача каждого слова / пакета данных происходит по запросу на ПДП от ПУ. Запросы вырабатываются по определенным событиям (см., например, подпункт 8.3.1.6). Разрешение / запрет запросов по каждому из них осуществляется программно (см. примеры в подразделе 8.4).

Программная инициализация ПДП-сеанса осуществляется ПО МК. Ее типовые процедуры описаны в подпунктах 8.3.1.6 и 8.3.2.7.

8.5.10. Обработка запросов на ПДП осуществляется в соответствии с присвоенным им приоритетом. В подсистемах ПДП большинства распространенных семейств МК используется способ многоуровневой квазистатической приоритезации:

- уровень приоритетности назначается программно (например, в МК семейства *ARM Cortex-Mx* – битовым полем *PL[1:0]* регистра конфигурации канала / потока контроллера ПДП, см. рис. 8.10 и 8.22);

- приоритет канала / потока в пределах группы с одинаковым уровнем приоритетности определяется его номером (см. подпункты 8.3.1.4 и 8.3.2.17).

В ряде семейств МП и МК используется также одноуровневая статическая и циклическая приоритезация.

8.5.11. Для оповещения ЦП об окончании ПДП-сеанса, а также о выявленных ошибках обмена архитектурой контроллеров ПДП используется механизм прерываний (см. подпункты 8.3.1.10 и 8.3.2.16). Как правило, в таблице векторов прерываний выделяются отдельные позиции только от глобального прерывания по каналу / потоку (в некоторых модельных рядах – по группе каналов). Идентификация конкретного события, вызвавшего прерывание, осуществляется программно, по битам признаков («флагам») соответствующих событий (см. рис. 8.9 и 8.20).

8.5.12. Типовые структурно-архитектурные решения подсистемы ПДП современных МК (на примере МК семейства *ARM Cortex-Mx*) описаны в подразделе 8.3, примеры программных модулей конфигурирования и запуска контроллеров ПДП МК семейства *ARM Cortex-Mx* – в подразделе 8.4.