14 Цифро-аналоговый преобразователь (ЦАП)

Этот раздел относится только к устройствам STM32F05x, STM32F07x и STM32F09x. Второй канал ЦАП (DAC_OUT2) и некоторые другие функции доступны только на устройствах STM32F07x и STM32F09x.

14.1 Введение

Модуль ЦАП представляет собой 12-разрядный цифро-аналоговый преобразователь с выходным напряжением. ЦАП может быть настроен в 8- или 12-битном режиме и может использоваться вместе с контроллером DMA. В 12-битном режиме данные могут быть выровнены по левому или правому краю. Вход опорного напряжения, VDDA (совместно с АЦП), доступен. Выход может быть дополнительно буферизован для привода с более высоким током.

14.2 Основные функции ЦАП

Устройства интегрируют один 12-битный канал ЦАП DAC_OUT1. Второй канал DAC_OUT2 доступен на устройствах STM32F07x и STM32F09x. Основные характеристики ЦАП:

- Левое или правое выравнивание данных в 12-битном режиме
- Возможность синхронизированного обновления
- Генерация шумовых волн (устройства STM32F07x и STM32F09x)
- Генерация треугольной волны (устройства STM32F07x и STM32F09x)
- Независимые или одновременные преобразования (только двойной режим)
- Возможность прямого доступа к памяти
- Обнаружение ошибки опустошения DMA
- Внешние триггеры для конвертации
- Программируемый внутренний буфер
- Заданное значение входного напряжения, VDDA. На рисунке 44 показана блок-схема канала ЦАП, а в таблице 51 приведено описание контактов.

14.3 Включение выходного буфера ЦАП

В ЦАП встроен один выходной буфер, который можно использовать для уменьшения выходного сопротивления и для непосредственного управления внешними нагрузками без необходимости добавления внешнего операционного усилителя.

Выходные буферы канала DAC могут быть включены и отключены с помощью соответствующего бита BOFFx в регистре DAC_CR.

14.4 Активация канала ЦАП

Каждый канал ЦАП может быть включен путем установки соответствующего бита ENx в регистре DAC_CR. Каждый канал ЦАП затем включается после времени запуска tWAKEUP. Примечание. Бит ENx включает только аналоговую макроячейку DAC Channelx. Цифровой интерфейс DAC Channelx включен, даже если бит ENx сброшен.

Figure 44. DAC block diagram

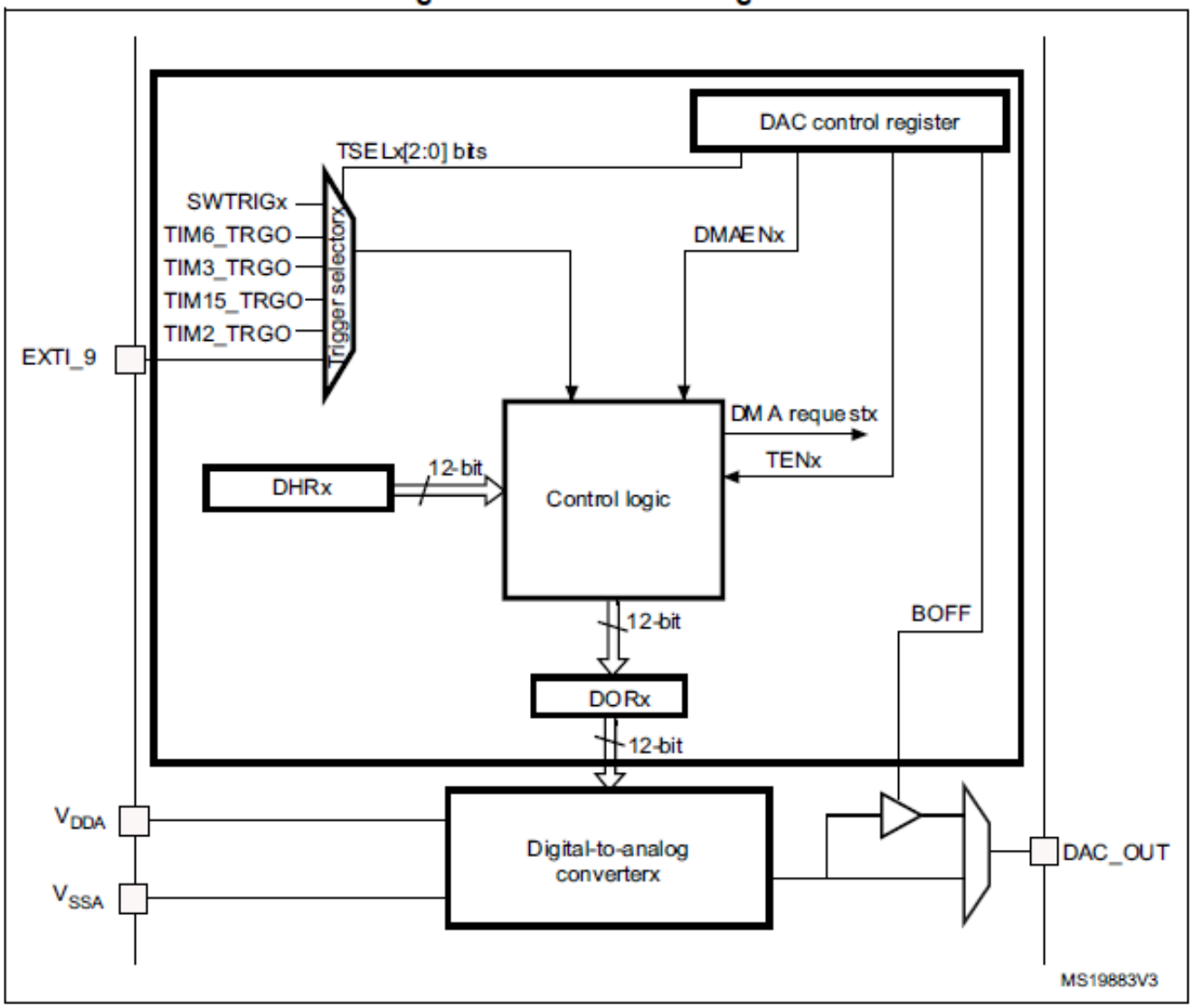


Table 51. DAC pins

Name	Signal type	Remarks
V_{DDA}	Input, analog supply	Analog power supply
V_{SSA}	Input, analog supply ground	Ground for analog power supply
DAC_OUT	Analog output signal	DAC channelx analog output

Примечание. После включения DAC_Channelx соответствующий вывод GPIO (PA4 или PA5) автоматически подключается к выходу аналогового преобразователя (DAC_OUTx). Чтобы избежать паразитного потребления, вывод PA4 или PA5 должен быть сначала настроен на аналоговый (AIN).

14.5 Функциональное описание одномодового режима

14.5.1 Формат данных ЦАП

Есть три варианта:

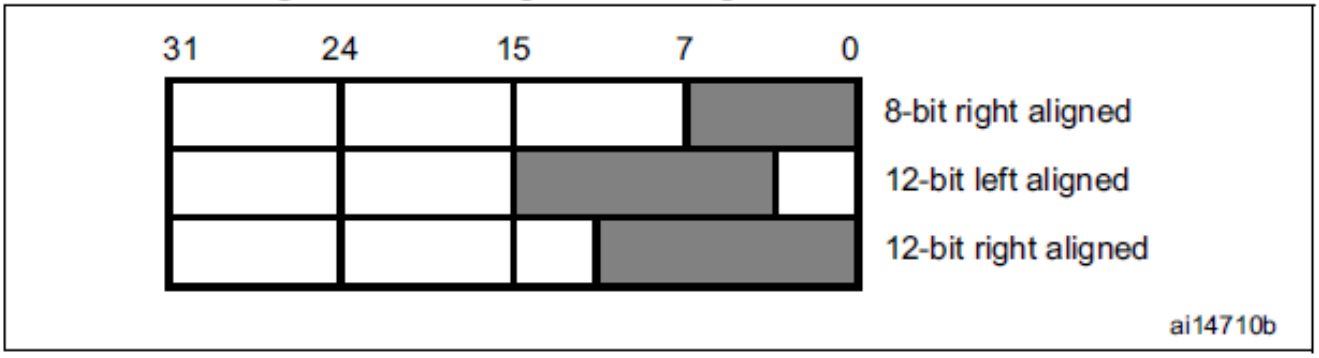
• 8-битное выравнивание по правому краю: программное обеспечение должно загружать данные в биты DAC_DHR8Rx [7: 0] (сохраненные в битах DHRx [11: 4])

• 12-битное выравнивание по левому краю: программное обеспечение должно загружать данные в биты DAC_DHR12Lx [15: 4] (сохраненные в битах DHRx [11: 0])

• 12-битное выравнивание по правому краю: программное обеспечение должно загружать данные в биты DAC_DHR12Rx [11: 0] (сохраненные в битах DHRx [11: 0])

В зависимости от загруженного регистра DAC_DHRyyyx данные, записанные пользователем, сдвигаются и сохраняются в соответствующий DHRx (регистр хранения данных, которые являются внутренними не отображаемыми в памяти регистрами). Затем регистр DHRx загружается в регистр DORx автоматически, с помощью программного запуска или с помощью внешнего события.

Figure 45. Data registers in single DAC channel mode



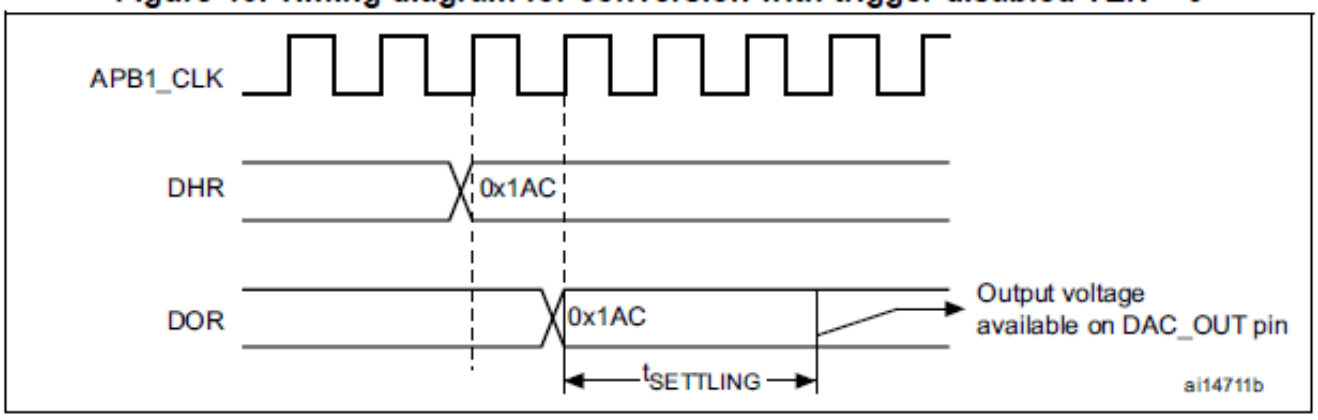
14.5.2 Преобразование канала ЦАП

DAC_DORx не может быть записан напрямую, и любая передача данных на канал DACxx должна выполняться путем загрузки регистра DAC_DHRx (запись в DAC_DHR8Rx, DAC_DHR12Lx, DAC_DHR12Rx).

Данные, хранящиеся в регистре DAC_DHRx, автоматически передаются в регистр DAC_DORx после одного тактового цикла APB, если не выбран аппаратный триггер (бит TENx в регистре DAC_CR сбрасывается). Однако, когда выбран аппаратный триггер (установлен бит TENx в регистре DAC_CR) и происходит триггер, передача выполняется через три тактовых цикла PCLK позже.

Когда в DAC_DORx загружается содержимое DAC_DHRx, напряжение аналогового выхода становится доступным через время tSETTLING, которое зависит от напряжения источника питания и нагрузки аналогового выхода.

Figure 46. Timing diagram for conversion with trigger disabled TEN = 0



Независимый триггер с одиночной генерацией LFSR

Для настройки ЦАП в этом режиме преобразования (см. Раздел 14.7: Генерация шума (устройства STM32F07x и STM32F09x)) требуется следующая последовательность действий:

1.Установите бит разрешения запуска канала ЦАП TENx.

2. Сконфигурируйте источник триггера, установив биты TSELx [2: 0].

3. Настройте биты WAVEx [1: 0] канала ЦАП как «01» и то же значение маски LFSR в битах MAMPx [3: 0]

4. Загрузите данные канала DAC в нужный регистр DAC_DHRx (DHR12RD, DHR12LD или DHR8RD).

При поступлении триггера DAC channelx счетчик LFSRx с той же маской добавляется в регистр DHRx, а сумма передается в DAC_DORx (три тактовых цикла APB позже). Затем счетчик LFSRx обновляется.

Независимый триггер с генерацией одного треугольника

Чтобы настроить ЦАП в этом режиме преобразования (см. Раздел 14.8: Генерация треугольных волн (устройства STM32F07x и STM32F09x)), необходима следующая последовательность действий:

1.Установите биты TENx разрешения запуска триггера ЦАП.

2. Сконфигурируйте источник триггера, установив биты TSELx [2: 0].

3. Настройте биты WACEx [1: 0] канала ЦАП как «1x» и то же значение максимальной амплитуды в битах MAMPx [3: 0]

4. Загрузите данные канала ЦАП в нужный регистр DAC_DHRx. (DHR12RD, DHR12LD или DHR8RD).

Когда поступает триггер канала DAC DAN, счетчик треугольника канала DAC с той же амплитудой треугольника добавляется в регистр DHRx, а сумма передается в DAC_DORx (три тактовых цикла APB позже). Счетчик треугольника канала ЦАП обновляется.

14.5.3 Выходное напряжение ЦАП

Цифровые входы преобразуются в выходные напряжения при линейном преобразовании между 0 и VDDA. Напряжения аналогового выхода на каждом выводе канала ЦАП определяются следующим уравнением:

$$DACoutput = VDDA \times DOR / 4096$$

14.5.4 Выбор триггера ЦАП

Если бит управления TENx установлен, преобразование может быть инициировано внешним событием (счетчик таймера, внешняя линия прерывания). Управляющие биты TSELx [2: 0] определяют, какие возможные события будут запускать преобразование, как показано в таблице 52.

Каждый раз, когда интерфейс ЦАП обнаруживает нарастающий фронт на выбранном выходе TRGO таймера или на выбранной линии 9 внешнего прерывания, последние данные, сохраненные в регистре DAC_DHRx, передаются в регистр DAC_DORx. Регистр DAC_DORx обновляется через три цикла APB после срабатывания триггера.

Table 52. External triggers

Source	Type	TSEL[2:0]
TIM6_TRGO event	Internal signal from on-chip timers	000
TIM3_TRGO event		001
TIM7_TRGO event		010
TIM15_TRGO event		011
TIM2_TRGO event		100
Reserved		101
EXTI line9	External pin	110
SWTRIG	Software control bit	111

Если выбран программный триггер, преобразование начинается после установки бита SWTRIG. SWTRIG сбрасывается аппаратно после загрузки регистра DAC_DORx с содержимым регистра DAC_DHRx.

Примечание. Бит TSELx [2: 0] нельзя изменить, если установлен бит ENx. Когда выбран программный триггер, передача из регистра DAC_DHRx в регистр DAC_DORx занимает только один тактовый такт APB.

14.6 Dual-mode functional description (STM32F07x and STM32F09x devices)

14.7 Noise generation(STM32F07x and STM32F09x devices)

14.8 Triangle-wave generation (STM32F07x and STM32F09xdevices)

14.9 запрос DMA

Каждый канал ЦАП имеет возможность DMA. Два канала DMA используются для обслуживания запросов DMA канала DAC.

Запрос DAC DMA генерируется, когда происходит внешний запуск (но не программный запуск), когда бит DMAENx установлен. Затем значение регистра DAC_DHRx передается в регистр DAC_DORx.

В двойном режиме, если установлены оба бита DMAENx, генерируются два запроса DMA. Если требуется только один запрос DMA, пользователь должен установить только соответствующий бит DMAENx. Таким образом, приложение может управлять обоими каналами ЦАП в двойном режиме, используя один запрос DMA и уникальный канал DMA.

Пример кода см. В разделе Приложения A.8.12: Пример кода инициализации DMA.

Недогрузка буфера DMA

Запрос DAC DMA не ставится в очередь, поэтому, если второй внешний триггер поступает до того, как будет получено подтверждение для первого внешнего триггера (первый запрос), то новый запрос не выдается, и в регистре DAC_SR установлен флаг опустошения DMA channelx DMAUDRx, сообщить

об ошибке. Передача данных DMA затем отключается, и дальнейший запрос DMA не обрабатывается. Канал ЦАП продолжает преобразовывать старые данные.

Программное обеспечение должно очистить флаг DMAUDRx, написав «1», очистить бит DMAEN от используемого потока DMA и повторно инициализировать оба канала DMA и DAC для правильного перезапуска передачи. Программное обеспечение должно изменить частоту преобразования триггера ЦАП или облегчить рабочую нагрузку прямого доступа к памяти, чтобы избежать нового прямого доступа к памяти. Наконец, преобразование ЦАП можно возобновить, включив как передачу данных DMA, так и запуск преобразования.

Для каждого канала ЦАП прерывание также генерируется, если включен соответствующий бит DMAUDRIEx в регистре DAC_CR.

\

14.10 Регистры ЦАП

Обратитесь к Разделу 1.1 на странице 42 для получения списка сокращений, используемых в описаниях регистров.

К периферийным регистрам нужно обращаться по словам (32-битным).

14.10.1 Регистр управления ЦАП (DAC_CR)

Смещение адреса: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
		rw	rw	rw	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
Res.	Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1	EN1
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Сбросить значение: 0x0000 0000

Биты 31: 30 зарезервированы, должны быть сохранены при значении сброса.

Бит 29 **DMAUDRIE2:** разрешение прерывания от опустошения DMA канала 2 DAC.

Этот бит устанавливается и очищается программным обеспечением.

0: DAC channel2 Прерывание опустошения DMA отключено

1: DAC channel2 Прерывание опустошения DMA включено

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Бит 28 **DMAEN2:** включение DMA канала ЦАП2

Этот бит устанавливается и очищается программным обеспечением.

0: DAC канал2 режим DMA отключен

1: DAC канал2 режим DMA включен

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Биты 27:24 **MAMP2** [3: 0]: селектор маски / амплитуды канала ЦАП2

Эти биты записываются программным обеспечением для выбора маски в режиме генерации волны или амплитуды в режиме генерации треугольника.

0000: снимите маску с бит 0 LFSR / амплитуды треугольника, равного 1

0001: снимите маску с бит [1: 0] LFSR / амплитуды треугольника, равные 3

0010: снимите маску с бит [2: 0] LFSR / амплитуды треугольника, равные 7

0011: снимите маску с бит [3: 0] LFSR / амплитуды треугольника, равные 15

0100: снимите маску с бит [4: 0] LFSR / амплитуды треугольника, равные 31

0101: снимите маску с бит [5: 0] LFSR / амплитуды треугольника, равные 63

0110: снимите маску с бит [6: 0] LFSR / амплитуды треугольника, равные 127

0111: снимите маску с бит [7: 0] LFSR / амплитуды треугольника, равные 255

1000: снимите маску с бит [8: 0] LFSR / амплитуды треугольника, равные 511

1001: снимите маску с бит [9: 0] LFSR / амплитуды треугольника, равные 1023

1010: снимите маску с бит[10: 0] LFSR / амплитуды треугольника, равные 2047

≥1011: снимите маску с бит [11: 0] амплитуды LFSR / треугольника, равные 4095

Примечание: эти биты доступны только в двойном режиме, когда поддерживается генерация волны.

В противном случае они зарезервированы и должны быть сохранены на уровне сброса.

Биты 23:22 **WAVE2** [1: 0]: включение генерации шума / треугольной волны ЦАП2 канала

Эти биты устанавливаются / сбрасываются программным обеспечением.

00: генерация волн отключена 01: генерация шумовых волн включена

1x: генерация треугольной волны включена

Примечание. Используется только в том случае, если бит TEN2 = 1 (триггер ЦАП 2 включен)

Эти биты доступны только в двойном режиме, когда поддерживается генерация волны.

В противном случае они зарезервированы и должны быть сохранены на уровне сброса.

Биты 21:19 **TSEL2** [2: 0]: выбор триггера канала ЦАП2

Эти биты выбирают внешнее событие, используемое для запуска ЦАП channel2

000: событие таймера 6 TRGO

001: событие таймера 3 TRGO

010: событие таймера 7 TRGO

011: событие таймера 15 TRGO

100: событие таймера 2 TRGO

101: зарезервировано

110: EXTI line9

111: программный триггер

Примечание. Используется только в том случае, если бит TEN2 = 1 (триггер ЦАП канала 2 включен).

Эти биты доступны только в двойном режиме. Они зарезервированы в одиночном режиме.

Бит 18 **TEN2:** включение триггера ЦАП для канала 2

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения триггера ЦАП Channel2.

0: триггер DAC channel2 отключен, и данные, записанные в регистр DAC_DHRx, передаются на один цикл APBclock позже в регистр DAC_DOR2

1: триггер DAC channel2 включен, и данные из регистра DAC_DHRx передаются через три тактовых цикла APB в регистр DAC_DOR2

Примечание. Если выбран программный триггер, передача из регистра DAC_DHRx в регистр DAC_DOR2 занимает только один тактовый такт APB.

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Бит 17 **BOFF2:** выходной буфер DAC channel2 отключен

Этот бит устанавливается и очищается программным обеспечением для включения / отключения выходного буфера ЦАП 2-го канала.

0: выходной буфер DAC channel2 включен

1: выходной буфер DAC channel2 отключен

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Бит 16 **EN2:** ЦАП Channel2 включен

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения канала ЦАП2.

0: DAC channel2 отключен

1: DAC channel2 включен

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Биты 15: 14 зарезервированы, должны быть сохранены при значении сброса.

Бит 13 **DMAUDRIE1**: ЦАП канал1, разрешение прерывания от DMA-канала

Этот бит устанавливается и очищается программным обеспечением.

0: DAC channel1 Прерывание DMA Underrun отключено

1: DAC channel1 Прерывание опустошения DMA включено

Бит 12 **DMAEN1**: включение DMA канала ЦАП1

Этот бит устанавливается и очищается программным обеспечением.

0: DAC канал1 Режим DMA отключен

1: DAC канал1 режим DMA включен

Биты 11: 8 **MAMP1** [3: 0]: селектор маски / амплитуды DAC channel1

Эти биты записываются программным обеспечением для выбора маски в режиме генерации волны или амплитуды в режиме генерации треугольника.

0000: снимите маску с бит 0 LFSR / амплитуды треугольника, равного 1

0001: снимите маску с бит [1: 0] LFSR / амплитуды треугольника, равные 3

0010: снимите маску с бит [2: 0] LFSR / амплитуды треугольника, равные 7

0011: снимите маску с бит [3: 0] LFSR / амплитуды треугольника, равные 15

0100: снимите маску с бит [4: 0] LFSR / амплитуды треугольника, равные 31

0101: снимите маску с бит [5: 0] LFSR / амплитуды треугольника, равные 63

0110: снимите маску с бит [6: 0] LFSR / амплитуды треугольника, равные 127

0111: снимите маску с бит [7: 0] LFSR / амплитуды треугольника, равные 255

1000: снимите маску с бит [8: 0] LFSR / амплитуды треугольника, равные 511

1001: снимите маску с бит [9: 0] LFSR / амплитуды треугольника, равные 1023

1010: разобрать биты [10: 0] LFSR / амплитуды треугольника, равные 2047

≥ 1011: разобрать биты [11: 0] амплитуды LFSR / треугольника, равные 4095

Биты 7: 6 **WAVE1** [1: 0]: разрешение генерации шума / треугольника ЦАП канала 1

Эти биты устанавливаются и очищаются программным обеспечением.

00: генерация волн отключена

01: Генерация шумовой волны включена

1x: генерация треугольной волны включена

Примечание. Используется, только если бит TEN1 = 1 (триггер ЦАП 1 включен).

Биты 5: 3 **TSEL1** [2: 0]: выбор триггера канала ЦАП1

Эти биты выбирают внешнее событие, используемое для запуска ЦАП channel1.

000: событие таймера 6 TRGO

001: событие таймера 3 TRGO

010: событие таймера 7 TRGO

011: событие таймера 15 TRGO

100: событие таймера 2 TRGO

101: зарезервировано

110: EXTI line9

111: программный триггер

Примечание. Используется, только если бит TEN1 = 1 (триггер ЦАП 1 включен).

Бит 2 **TEN1**: включение триггера ЦАП для канала 1

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения триггера ЦАП Channel1.

0: триггер DAC channel1 отключен, и данные, записанные в регистр DAC_DHRx, передаются на один тактовый цикл APB позже в регистр DAC_DOR1

1: Триггер DAC channel1 активирован, и данные из регистра DAC_DHRx передаются через три тактовых цикла APB в регистр DAC_DOR1

Примечание. Если выбран программный триггер, передача из регистра DAC_DHRx в регистр DAC_DOR1 занимает только один тактовый такт APB.

Бит 1 **BOFF1**: выходной буфер DAC channel1 отключен

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения выходного буфера ЦАП channel1.

0: выходной буфер DAC channel1 включен

1: выходной буфер DAC channel1 отключен

Бит 0 **EN1**: ЦАП Channel1 включен

Этот бит устанавливается и сбрасывается программным обеспечением для включения / выключения ЦАП channel1.

0: канал1 ЦАП отключен

1: канал1 ЦАП включен

14.10.2 Регистр программного запуска ЦАП (DAC_SWTRIGR)

Смещение адреса: 0x04

Сбросить значение: 0x0000 0000

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.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

Биты 31: 2 зарезервированы, должны быть установлены на значение сброса.

Бит 1 **SWTRIG2:** программный триггер ЦАП канала 2

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения программного запуска.

0: программный триггер отключен

1: программный триггер включен

Примечание. Этот бит очищается аппаратно (один тактовый цикл APB позже) после загрузки значения регистра DAC_DHR2 в регистр DAC_DOR2.

Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Бит 0 **SWTRIG1:** программный запуск ЦАП channel1

Этот бит устанавливается и сбрасывается программным обеспечением для включения / отключения программного запуска.

0: программный триггер отключен

1: программный триггер включен

Примечание. Этот бит очищается аппаратно (один тактовый цикл APB позже) после загрузки значения регистра DAC_DHR1 в регистр DAC_DOR1.

14.10.3 ЦАП канал1 12-битный выровненный по правому краю регистр хранения данных (DAC_DHR12R1)

Смещение адреса: 0x08

Сбросить значение: 0x0000 0000

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.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 12 Зарезервированы, должны быть сохранены в значении сброса.

Биты 11: 0 **DACC1DHR** [11: 0]: ЦАП канал1 12-битные данные с выравниванием по праву Эти биты записываются программным обеспечением, которое задает 12-битные данные для канала ЦАП1.

14.10.4 ЦАП канал1 12-разрядный регистр хранения данных с выравниванием по левому краю (DAC_DHR12L1)

Смещение адреса: 0x0C

Сбросить значение: 0x0000 0000

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.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												v	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

Биты 31: 16 зарезервированы, должны быть сохранены при значении сброса.

Биты 15: 4 **DACC1DHR** [11: 0]: ЦАП channel1 12-битовые данные с выравниванием по левому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП1.

Биты 3: 0 зарезервированы, должны быть сохранены при значении сброса.

14.10.5 ЦАП channel1 8-битный выровненный по правому краю регистр хранения данных (DAC_DHR8R1)

Смещение адреса: 0x10

Сбросить значение: 0x0000 0000

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.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 8 зарезервированы, должны быть сохранены в значении сброса.

Биты 7: 0 **DACC1DHR** [7: 0]: 8-разрядные данные DAC channel1 с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 8-битные данные для канала ЦАП1.

14.10.6 ЦАП channel2 12-битный выровненный по правому краю регистр хранения данных (DAC_DHR12R2)

Смещение адреса: 0x14

Сбросить значение: 0x0000 0000

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.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 12 Зарезервированы, должны быть сохранены в значении сброса.

Биты 11: 0 **DACC2DHR** [11: 0]: ЦАП channel2 12-битные данные с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП2.

14.10.7 ЦАП channel2 12-разрядный регистр хранения данных с выравниванием по левому краю (DAC_DHR12L2)

Смещение адреса: 0x18

Сбросить значение: 0x0000 0000

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.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

Биты 31: 16 зарезервированы, должны быть сохранены при значении сброса.

Биты 15: 4 **DACC2DHR** [11: 0]: ЦАП channel2 12-битовые данные с выравниванием по левому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП2.

Биты 3: 0 зарезервированы, должны быть сохранены при значении сброса

14.10.8 ЦАП channel2 8-битный выровненный по правому краю регистр хранения данных (DAC_DHR8R2)

Смещение адреса: 0x1C

Сбросить значение: 0x0000 0000

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.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 8 зарезервированы, должны быть сохранены в значении сброса.

Биты 7: 0 **DACC2DHR** [7: 0]: ЦАП channel2 8-битные данные с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 8-битные данные для канала ЦАП2.

14.10.9 Двойной ЦАП 12-битный выровненный по правому краю регистр хранения данных (DAC_DHR12RD)

Смещение адреса: 0x20

Сбросить значение: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	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
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Биты 31: 28 Зарезервированы, должны быть сохранены при значении сброса.

Биты 27:16 **DACC2DHR** [11: 0]: ЦАП channel2 12-битные данные с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП2.

Биты 15: 12 зарезервированы, должны быть сохранены при значении сброса.

Биты 11: 0 DACC1DHR [11: 0]: ЦАП channel1 12-битные данные с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП1

14.10.10 Двойной ЦАП 12-битный выровненный по левому краю регистр хранения данных (DAC_DHR12LD)

Смещение адреса: 0x24

Сбросить значение: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rw	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
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

Биты 31:20 **DACC2DHR** [11: 0]: ЦАП channel2 12-битовые данные с выравниванием по левому краю

Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП2.

Биты 19: 16 Зарезервированы, должны быть сохранены при значении сброса.

Биты 15: 4 **DACC1DHR** [11: 0]: ЦАП channel1 12-битовые данные с выравниванием по левому краю

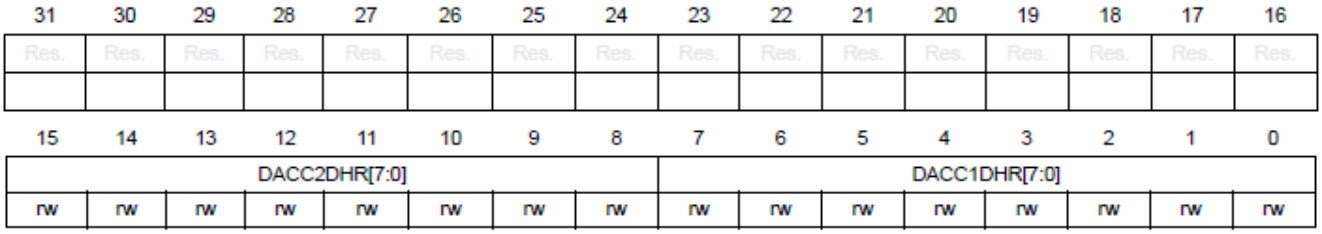
Эти биты записываются программным обеспечением, которое определяет 12-битные данные для канала ЦАП1.

Биты 3: 0 зарезервированы, должны быть сохранены при значении сброса.

14.10.11 Двойной ЦАП 8-битный выровненный по правому краю регистр хранения данных (DAC_DHR8RD)

Смещение адреса: 0x28

Сбросить значение: 0x0000 0000



Биты 31: 16 зарезервированы, должны быть сохранены при значении сброса.

Биты 15: 8 **DACC2DHR** [7: 0]: DAC channel2 8-битные данные с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 8-битные данные для канала ЦАП2.

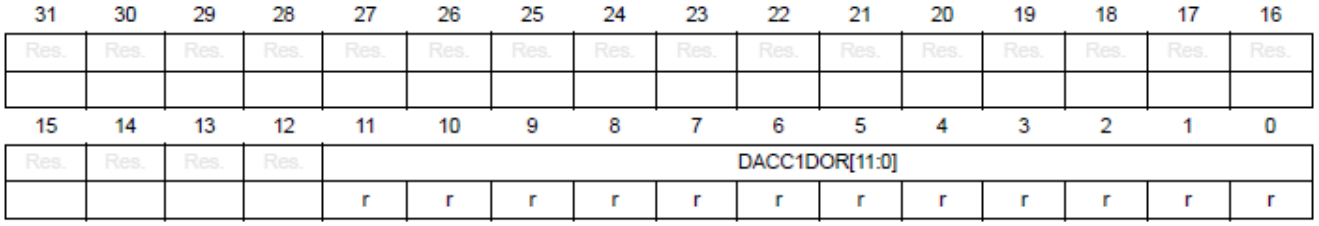
Биты 7: 0 **DACC1DHR** [7: 0]: 8-разрядные данные DAC channel1 с выравниванием по правому краю

Эти биты записываются программным обеспечением, которое определяет 8-битные данные для канала ЦАП1.

14.10.12 Регистр вывода данных DAC channel1 (DAC_DOR1)

Смещение адреса: 0x2C

Сбросить значение: 0x0000 0000



Биты 31: 12 Зарезервированы, должны быть сохранены в значении сброса.

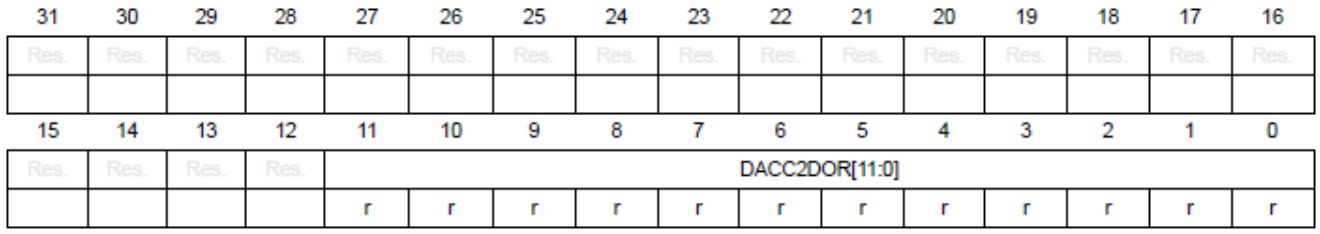
Биты 11: 0 **DACC1DOR** [11: 0]: вывод данных ЦАП channel1

Эти биты доступны только для чтения, они содержат выходные данные для канала ЦАП1.

14.10.13 Регистр вывода данных ЦАП channel2 (DAC_DOR2)

Смещение адреса: 0x30

Сбросить значение: 0x0000 0000



Биты 31: 12 Зарезервированы, должны быть сохранены в значении сброса.

Биты 11: 0 **DACC2DOR** [11: 0]: вывод данных ЦАП channel2

Эти биты предназначены только для чтения, они содержат выходные данные для канала ЦАП2.

14.10.14 Регистр состояния ЦАП (DAC_SR)

Смещение адреса: 0x34

Сбросить значение: 0x0000 0000

Биты 31: 30 зарезервированы, должны быть сохранены при значении сброса.

Бит 29 **DMAUDR2**: флаг опустошения DMA канала 2 DMA. Доступно на устройствах STM32F07x и STM32F09x.

Этот бит устанавливается аппаратно и очищается программно (записывая его в 1).

0: Нет ошибки опустошения DMA для канала DAC2

1: возникла ошибка ошибки опустошения DMA для канала DAC2 (выбранный в настоящий момент триггер управляет преобразованием канала DAC2 на частоте, превышающей скорость возможности услуги DMA)

Примечание. Этот бит доступен только в двойном режиме. Зарезервировано в одиночном режиме.

Биты 28: 14 зарезервированы, должны быть сохранены при значении сброса.

Бит 13 **DMAUDR1**: флаг опустошения DMA channel1 DMA

Этот бит устанавливается аппаратно и очищается программно (записывая его в 1).

0: Нет ошибки опустошения DMA для канала DAC1

1: возникла ошибка ошибки опустошения DMA для канала DAC1 (выбранный в настоящий момент триггер управляет преобразованием канала DAC1 на частоте, превышающей частоту возможностей услуги DMA)

Биты 12: 0 зарезервированы, должны быть установлены на значение сброса.

A.8 DAC

A.8.1 Independent trigger without wave generation code example Пример независимого триггера без кода генерации волны

/* (1) Enable the peripheral clock of the DAC */
/* (1) Включить периферийные часы ЦАП */
/* (2) Enable DMA transfer on DAC ch1 and ch2, enable interrupt on DMA underrun DAC ch1 and ch2, enable the DAC ch1 and ch2, select TIM6 as trigger by keeping 000 in TSEL1 select TIM7 as trigger by writing 010 in TSEL2 */
* (2) Включите передачу DMA на ЦАП ch1 и ch2, включите прерывание на DMA, опустошающей ЦАП ch1 и ch2, включите ЦАП ch1 и ch2, выберите TIM6 как триггер, сохранив 000 в TSEL1, выберите TIM7 как триггер, записав 010 в TSEL2 */

```
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_TSEL2_1 | DAC_CR_DMAUDRIE2 | DAC_CR_DMAEN2
| DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_DMAUDRIE1 | DAC_CR_DMAEN1 | DAC_CR_BOFF1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* Initialize the DAC value on ch1 */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* Initialize the DAC value on ch2 */
```

A.8.2 Independent trigger with single LFSR generation code example Независимый триггер с одним примером кода генерации LFSR

/* (1) Enable the peripheral clock of the DAC */
/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 01 and LFSR mask amplitude (MAMPx) at 1000 for a 511-bits amplitude, enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL2, and select TIM6 as trigger by keeping 000 in TSEL1 */
/* (2) Сконфигурируйте WAVEx на 01 и амплитуду маски LFSR (MAMPx) на 1000 для 511-битной амплитуды, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2, выберите TIM7 как триггер, записав 010 в TSEL2, и выберите TIM6 как триггер с помощью хранение 000 в TSEL1 */

```
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_MAMP1_3
| DAC_CR_MAMP2_3 | DAC_CR_WAVE2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2
| DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_BOFF1 | DAC_CR_TEN1
| DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* Initialize the DAC output value */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* Initialize the DAC output value */
```

A.8.3 Independent trigger with different LFSR generation code example Независимый триггер с другим примером кода генерации LFSR

/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 01 and LFSR mask amplitude (MAMPx) at 1000 for a 511-bits amplitude, LFSR mask amplitude (MAMP2) at 0111 i.e. a 255-bits amplitude for ch2, enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL2, and select TIM6 as trigger by keeping 000 in TSEL1 */
/* (2) Сконфигурируйте WAVEx в 01 и амплитуду маски LFSR (MAMPx) в 1000 для амплитуды

511 битов, амплитуду маски LFSR (MAMP2) в 0111, то есть амплитуду 255 битов для ch2, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2 выберите TIM7 в качестве триггера, написав 010 в TSEL2, и выберите TIM6 в качестве триггера, сохранив 000 в TSEL1 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0 | DAC_CR_MAMP1_3
| DAC_CR_MAMP2_2 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* Initialize DAC output value */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* Initialize DAC output value */

A.8.4 Independent trigger with single triangle generation code example Независимый триггер с примером кода генерации одного треугольника

/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 10 and LFSR mask amplitude (MAMPx) at 1001 for a 1023-bits amplitude, enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL2 and select TIM6 as trigger by keeping 000 in TSEL1 */
/* (2) Настройте WAVEx на 10 и амплитуду маски LFSR (MAMPx) на 1001 для амплитуды 1023-бит, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2, выберите TIM7 в качестве триггера, записав 010 в TSEL2, и выберите TIM6 в качестве триггера, сохранив 000 в TSEL1 */
/* (3) Define the low value of the triangle on channel1 */
/* (4) Define the low value of the triangle on channel2 */
/* (4) Определите нижнее значение треугольника на канале 2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
| DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
| DAC_CR_MAMP2_3 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* (3) */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* (4) */

A.8.5 Independent trigger with different triangle generation code example Независимый триггер с другим примером кода генерации треугольника

/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 10, configure mask amplitude for ch1 (MAMP1) at 1001 for a 1023-bits amplitude, and mask amplitude for ch2 (MAMP1) at 1011 for a 4095-bits amplitude, enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL2, and select TIM6 as trigger by keeping 000 in TSEL1 */
/* (2) Настройте WAVEx на 10, настройте амплитуду маски для ch1 (MAMP1) на 1001 для амплитуды 1023-бит и амплитуду маски для ch2 (MAMP1) на 1011 для амплитуды 4095 бит, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2, выберите TIM7 как триггер, записав 010 в TSEL2, и выберите TIM6 как триггер, сохранив 000 в TSEL1 */
/* (3) Define the low value of the triangle on channel1 */
/* (4) Define the low value of the triangle on channel2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
| DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
| DAC_CR_MAMP2_3 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */

```
DAC->DHR12R1 = DAC_OUT1_VALUE; /* (3) */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* (4) */
```

A.8.6 Simultaneous software start code example
Пример кода одновременного запуска программного обеспечения

```
/* Load the dual DAC channel data to the desired DHR register */
/* Загрузите данные двойного канала ЦАП в нужный регистр DHR*/
DAC->DHR12RD = (uint32_t)((signal1[x] << 16) + signal2[x]);
```

A.8.7 Simultaneous trigger without wave generation code example
Пример одновременного запуска без кода генерации волны

```
/* (1) Enable the peripheral clock of the DAC */
/* (2) Enable DMA transfer on DAC ch1 for both channels, enable the DAC ch1 and ch2, select TIM7
as trigger by writing 010 in TSEL1 and TSEL2 */
/* (2) Включите передачу DMA на ЦАП ch1 для обоих каналов, включите ЦАП ch1 и ch2, выбе-
рите TIM7 в качестве триггера, записав 010 в TSEL1 и TSEL2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_TSEL1_1 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL2_1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* Initialize the dual DAC value */
DAC->DHR12RD = (uint32_t)((2048 << 16) + 2048);
```

A.8.8 Simultaneous trigger with single LFSR generation code example
Одновременный запуск с одним примером кода генерации LFSR

```
/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 01 and LFSR mask amplitude (MAMPx) at 1000 for a 511-bits amplitude,
enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in
TSEL1 and TSEL2 */
/* (2) Настройте WAVEx на 01 и амплитуду маски LFSR (MAMPx) на 1000 для 511-битной ам-
плитуды, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2, выберите TIM7 как триггер,
записав 010 в TSEL1 и TSEL2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0
| DAC_CR_MAMP1_3 | DAC_CR_MAMP2_3
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2
| DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL1_1 | DAC_CR_BOFF1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* Initialize the dual register */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);
```

A.8.9 Simultaneous trigger with different LFSR generation code example
A.8.9 Одновременный запуск с другим примером кода генерации LFSR

```
/* (1) Enable the peripheral clock of the DAC */
/* (1) Включить периферийные часы ЦАП */
/* (2) Configure WAVEx at 01 and LFSR mask amplitude (MAMP1) at 1000 for a 511-bits amplitude,
set LFSR mask amplitude (MAMP2) at 0111 i.e. a 255-bits amplitude for ch2, enable the DAC ch1
and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL1 and TSEL2 */
/* (2) Сконфигурируйте WAVEx в 01 и амплитуду маски LFSR (MAMP1) в 1000 для 511-битной
амплитуды, установите амплитуду маски LFSR (MAMP2) в 0111, то есть 255-битную амплитуду
```

для ch2, включите ЦАП ch1 и ch2, отключите буфер на каналах ch1 и ch2 выберите TIM7 в качестве триггера, записав 010 в TSEL1 и TSEL2*/
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0 | DAC_CR_MAMP1_3
| DAC_CR_MAMP2_2 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2
| DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL1_1| DAC_CR_BOFF1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* Initialize the dual register */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);

A.8.10 Simultaneous trigger with single triangle generation code example
Одновременный запуск с примером кода генерации одного треугольника

/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 10 and LFSR mask amplitude (MAMPx) at 1001 for a 1023-bits amplitude, enable the DAC ch1 and ch2, disable buffer on ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL1 and TSEL2 */
/* (2) Сконфигурируйте WAVEx на 10 и амплитуду маски LFSR (MAMPx) на 1001 для амплитуды 1023 бит, включите ЦАП ch1 и ch2, отключите буфер на ch1 и ch2, выберите TIM7 в качестве триггера, записав 010 в TSEL1 и TSEL2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
| DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
| DAC_CR_MAMP2_3 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_BOFF2
| DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL1_1 | DAC_CR_BOFF1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* Initialize the dual register */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);

A.8.11 Simultaneous trigger with different triangle generation code example
Одновременный триггер с другим примером кода генерации треугольника

/* (1) Enable the peripheral clock of the DAC */
/* (2) Configure WAVEx at 10, configure mask amplitude for ch1 (MAMP1) at 1001 for a 1023-bits amplitude and mask amplitude for ch2 (MAMP1) at 1011 for a 4095-bits amplitude, enable the DAC ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL1 and TSEL2 */
/* (2) Настройте WAVEx на 10, настройте амплитуду маски для ch1 (MAMP1) на 1001 для амплитуды 1023-бит и амплитуду маски для ch2 (MAMP1) на 1011 для амплитуды 4095 бит, включите ЦАП ch1 и ch2, выберите TIM7 в качестве триггера с помощью писать 010 в TSEL1 и TSEL2 */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
| DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
| DAC_CR_MAMP2_3 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
| DAC_CR_TSEL2_1 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL1_1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */

```
/* Initialize the dual register */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);
```

A.8.12 DMA initialization code example
Пример кода инициализации DMA

```
/* (1) Enable DMA transfer on DAC ch1 for both channels, enable interrupt on DMA underrun DAC,
enable the DAC ch1 and ch2, select TIM7 as trigger by writing 010 in TSEL1 and TSEL2 */
/* (1) Включите передачу DMA на ЦАП ch1 для обоих каналов, включите прерывание на ЦАП с
обработкой DMA, активируйте ЦАП ch1 и ch2, выберите TIM7 в качестве триггера, записав 010
в TSEL1 и TSEL2 */
DAC->CR |= DAC_CR_TSEL1_1 | DAC_CR_TEN2 | DAC_CR_EN2
| DAC_CR_TSEL2_1 | DAC_CR_DMAUDRIE1 | DAC_CR_DMAEN1
| DAC_CR_TEN1 | DAC_CR_EN1; /* (1) */
/* (1) Enable the peripheral clock on DMA */
/* (2) Configure the peripheral data register address */
/* (3) Configure the memory address */
/* (4) Configure the number of DMA tranfer to be performs on channel 3 */
/* (5) Configure increment, size (32-bits), interrupts, transfer from
memory to peripheral and circular mode */
/* (6) Enable DMA Channel 3 */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
DMA1_Channel3->CPAR = (uint32_t) (&(DAC->DHR12RD)); /* (2) */
DMA1_Channel3->CMAR = (uint32_t)signal_data; /* (3) */
DMA1_Channel3->CNDTR = SIGNAL_ARRAY_SIZE; /* (4) */
DMA1_Channel3->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_1 | DMA_CCR_PSIZE_1
| DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel3->CCR |= DMA_CCR_EN; /* (6) */
```