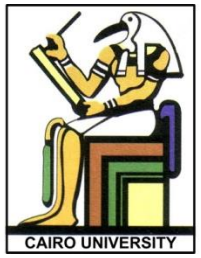


STM32 MICROCONTROLLER: DIGITAL-TO-ANALOG CONVERTER (DAC)

Lecture 6

Prof. Yasser Mostafa Kadah

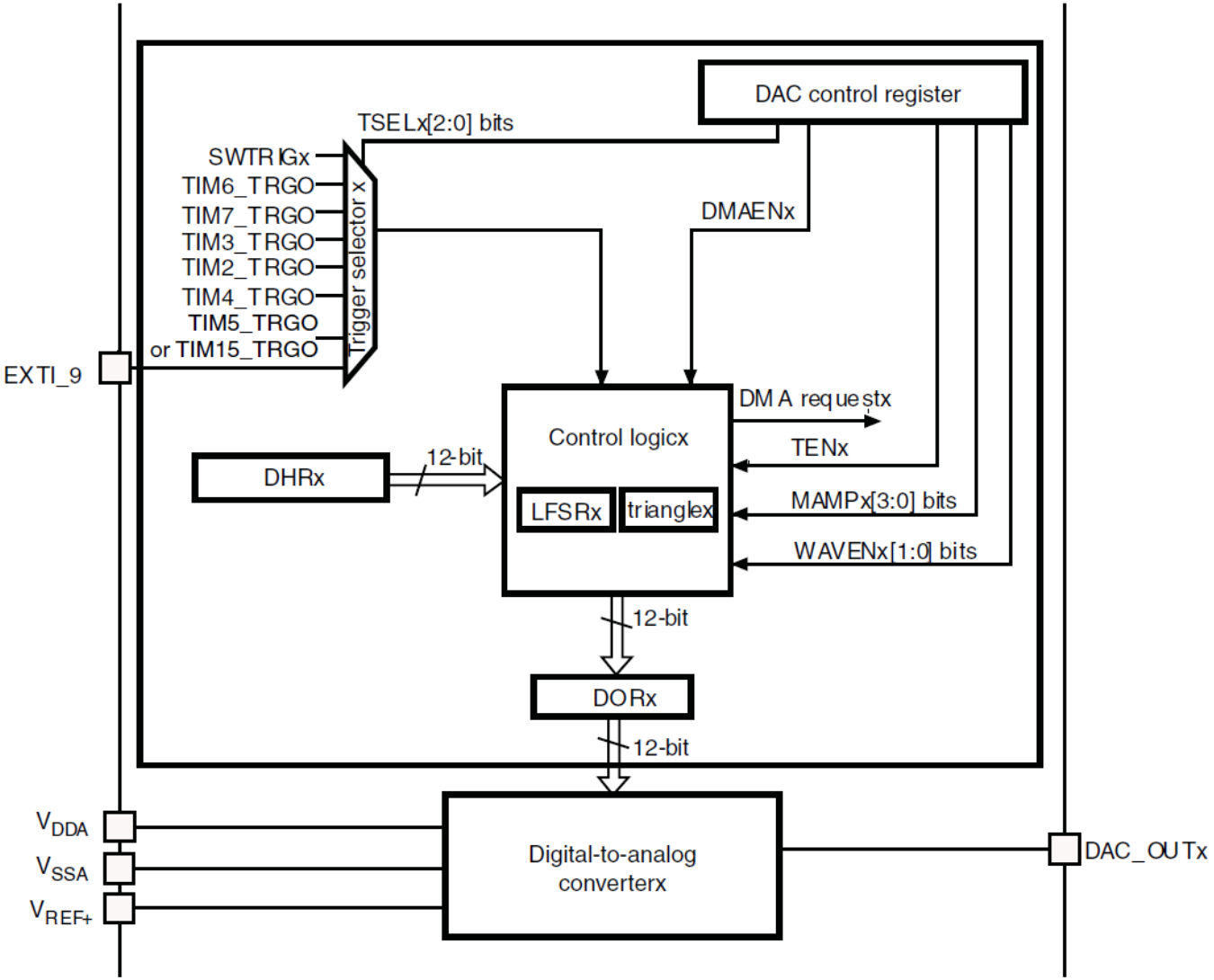
DAC Introduction

- DAC module is a 12-bit, voltage output digital-to-analog converter
- DAC can be configured in 8- or 12-bit mode and may be used in conjunction with the DMA controller
  - In 12-bit mode, the data could be left- or right-aligned
- DAC has two output channels, each with its own converter
  - In dual DAC channel mode, conversions could be done independently or simultaneously when both channels are grouped together for synchronous update operations
- Input reference pin, VREF+ (shared with ADC) is available for better resolution

DAC Main Features

- Two DAC converters: one output channel each
- Left or right data alignment in 12-bit mode
- Synchronized update capability
- Noise-wave generation
- Triangular-wave generation
- Dual DAC channel for independent or simultaneous conversions
- DMA capability for each channel
- DMA underrun error detection
- External triggers for conversion
- Input voltage reference, VREF+

DAC Channel Block Diagram



DAC Pins

- Once the DAC channelx is enabled, the corresponding GPIO pin (PA4 or PA5) is automatically connected to the analog converter output (DAC_OUTx)
- In order to avoid parasitic consumption, the PA4 or PA5 pin should first be configured to analog (AIN)

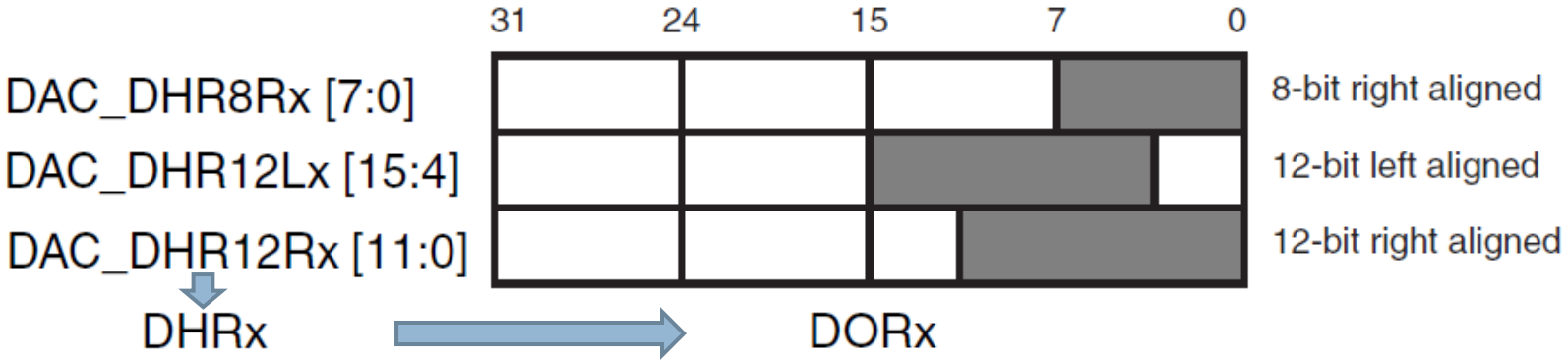
Name	Signal type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the DAC, 2.4 V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}	Input, analog supply	Analog power supply
V_{SSA}	Input, analog supply ground	Ground for analog power supply
DAC_OUTx	Analog output signal	DAC channelx analog output

DAC Functional Description

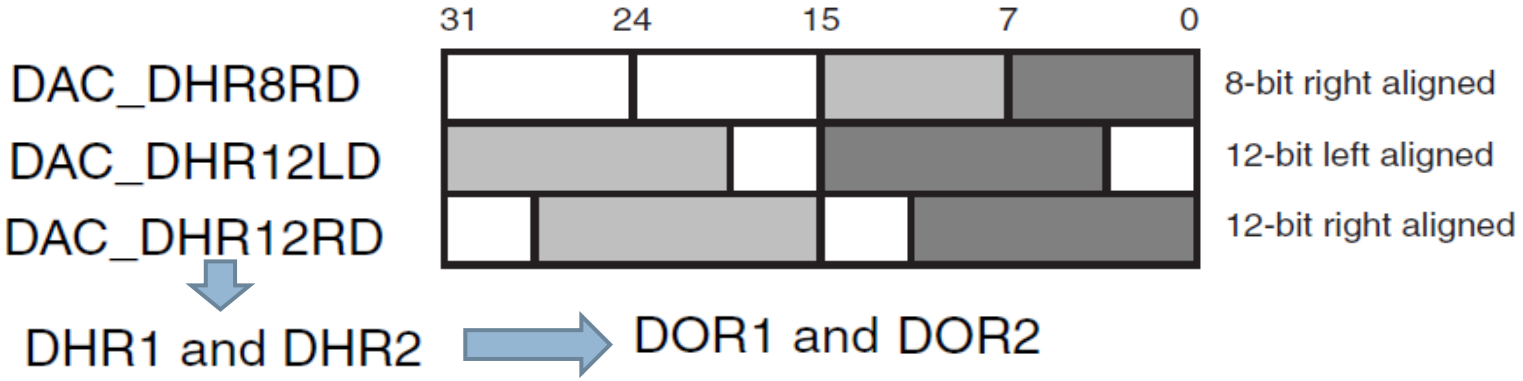
- DAC channel enable
  - Setting its corresponding ENx bit in the DAC_CR register
  - DAC channel is then enabled after a startup time t_{WAKEUP}
- DAC output buffer enable
  - DAC integrates two output buffers to reduce output impedance, and to drive external loads directly without having to add an external op amp
  - Enabled using the corresponding BOFFx bit in the DAC_CR register

DAC Functional Description

- DAC data format
  - Single DAC channelx



  - Dual DAC channels



DAC Functional Description

- DAC output voltage

$$DACoutput = V_{REF} \times \frac{DOR}{4095}$$

- DAC trigger selection
  - Rising edge for HW triggers: conversion after three APB1 clock cycles
  - If the software trigger is selected, the conversion starts once the SWTRIG bit is set. SWTRIG is reset by hardware once the DAC_DORx register has been loaded with the DAC_DHRx register contents

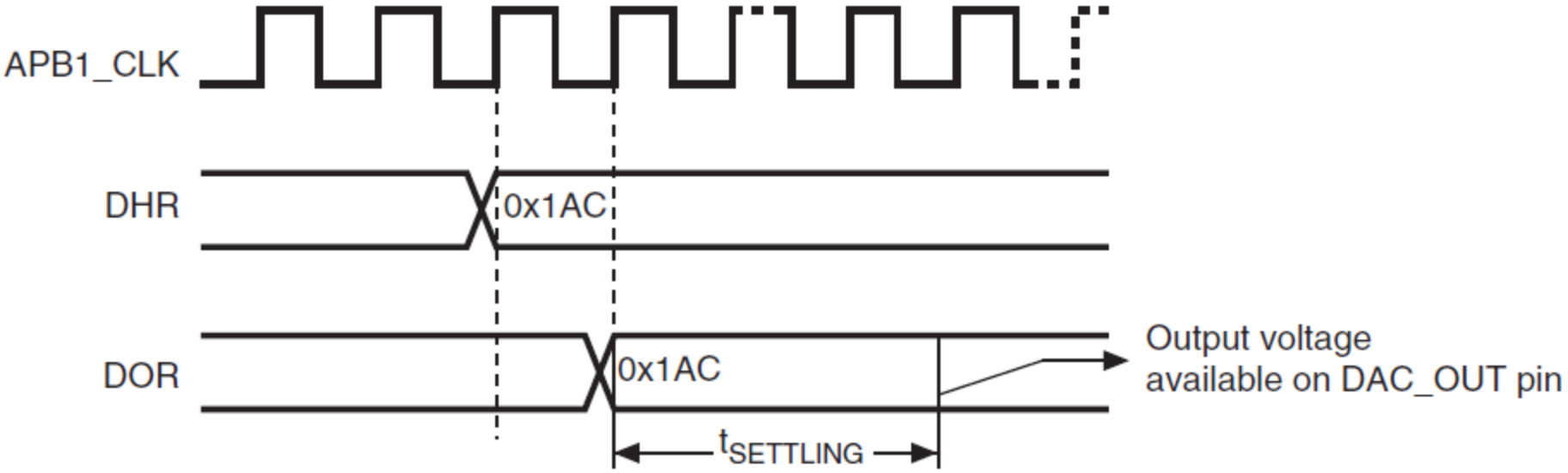
Source	Type	TSEL[2:0]
Timer 6 TRGO event	Internal signal from on-chip timers	000
Timer 3 TRGO event		001
Timer 7 TRGO event		010
Timer 5 or Timer 15 TRGO event		011
Timer 2 TRGO event		100
Timer 4 TRGO event		101
EXTI line9	External pin	110
SWTRIG	Software control bit	111

DAC Conversion

- DAC_DORx cannot be written directly and any data transfer to the DAC channelx must be performed by loading the DAC_DHRx register
- Data stored in the DAC_DHRx register are automatically transferred to the DAC_DORx register after one APB1 clock cycle, if no hardware trigger is selected (TENx bit in DAC_CR register is reset)
- When hardware trigger is selected (TENx bit in DAC_CR register is set) and a trigger occurs, transfer is performed three APB1 clock cycles later
- When DAC_DORx is loaded with the DAC_DHRx contents, the analog output voltage becomes available after a time $t_{SETTLING}$ that depends on power supply voltage and analog output load

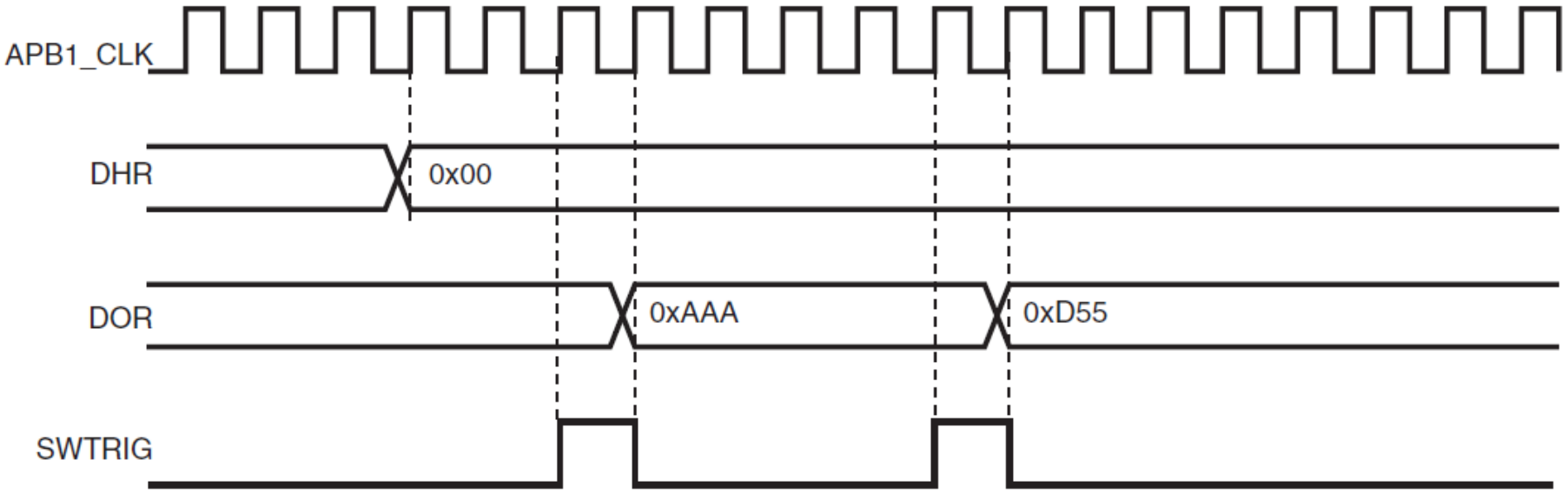
DAC Conversion

- Timing diagram for conversion with trigger disabled TEN = 0



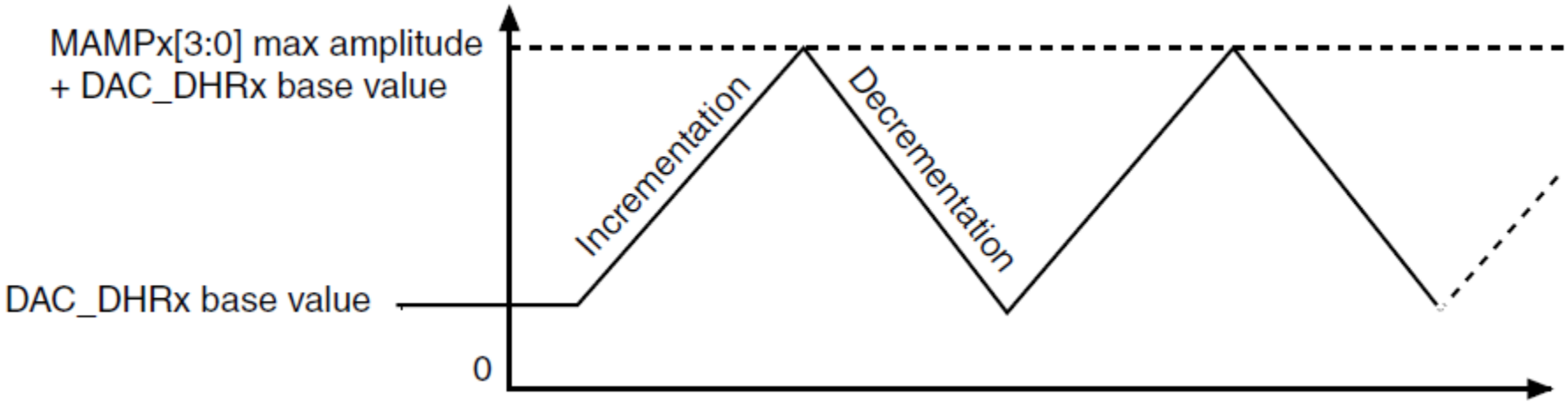
Noise Generation

- In order to generate a variable-amplitude pseudonoise, an LFSR (linear feedback shift register) is available
- Noise generation is selected by setting WAVEx[1:0] to “01”
- The preloaded value in LFSR is 0xAAA and is updated after each trigger event, following a specific calculation algorithm



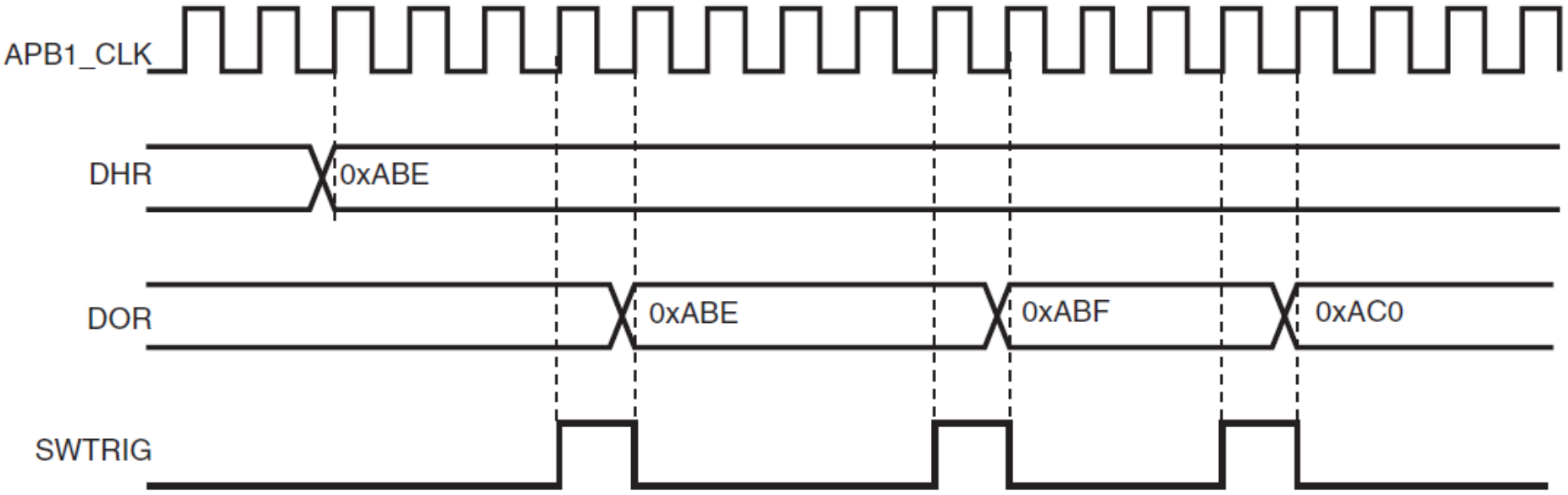
Triangle-Wave Generation

- It is possible to add a small-amplitude triangular waveform on a DC or slowly varying signal.
- DAC triangle-wave generation is selected by setting WAVEx[1:0] to “10”
  - Amplitude is configured through MAMPx[3:0] bits in DAC_CR register
- It is possible to reset triangle wave generation by resetting the WAVEx[1:0] bits



Triangle-Wave Generation

- DAC conversion (SW trigger enabled) with triangle wave generation
  - DAC trigger must be enabled by setting the TENx bit in DAC_CR register
  - The MAMPx[3:0] bits must be configured before enabling the DAC, otherwise they cannot be changed



DAC Control Register (DAC_CR)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		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
Reserved		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

Bits 11:8 **MAMP1[3:0]**: DAC channel1 mask/amplitude selector
These bits are written by software to select mask in wave generation mode or amplitude ir triangle generation mode.
0000: Unmask bit0 of LFSR/ triangle amplitude equal to 1
0001: Unmask bits[1:0] of LFSR/ triangle amplitude equal to 3
0010: Unmask bits[2:0] of LFSR/ triangle amplitude equal to 7
0011: Unmask bits[3:0] of LFSR/ triangle amplitude equal to 15
0100: Unmask bits[4:0] of LFSR/ triangle amplitude equal to 31
0101: Unmask bits[5:0] of LFSR/ triangle amplitude equal to 63
0110: Unmask bits[6:0] of LFSR/ triangle amplitude equal to 127
0111: Unmask bits[7:0] of LFSR/ triangle amplitude equal to 255
1000: Unmask bits[8:0] of LFSR/ triangle amplitude equal to 511
1001: Unmask bits[9:0] of LFSR/ triangle amplitude equal to 1023
1010: Unmask bits[10:0] of LFSR/ triangle amplitude equal to 2047
≥ 1011: Unmask bits[11:0] of LFSR/ triangle amplitude equal to 4095

Bits 7:6 **WAVE1[1:0]**: DAC channel1 noise/triangle wave generation enable
These bits are set and cleared by software.
00: wave generation disabled
01: Noise wave generation enabled
1x: Triangle wave generation enabled

Bits 5:3 **TSEL1[2:0]**: DAC channel1 trigger selection

Bit 2 **TEN1**: DAC channel1 trigger enable

Bit 1 **BOFF1**: DAC channel1 output buffer disable

Bit 0 **EN1**: DAC channel1 enable

DAC Software Trigger Register (DAC_SWTRIGR)

Reset value: 0x0000 0000

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														SWTRIG2	SWTRIG1
														w	w

Bit 1 **SWTRIG2**: DAC channel2 software trigger

This bit is set and cleared by software to enable/disable the software trigger.

0: Software trigger disabled

1: Software trigger enabled

Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_DHR2 register value has been loaded into the DAC_DOR2 register.

Bit 0 **SWTRIG1**: DAC channel1 software trigger

This bit is set and cleared by software to enable/disable the software trigger.

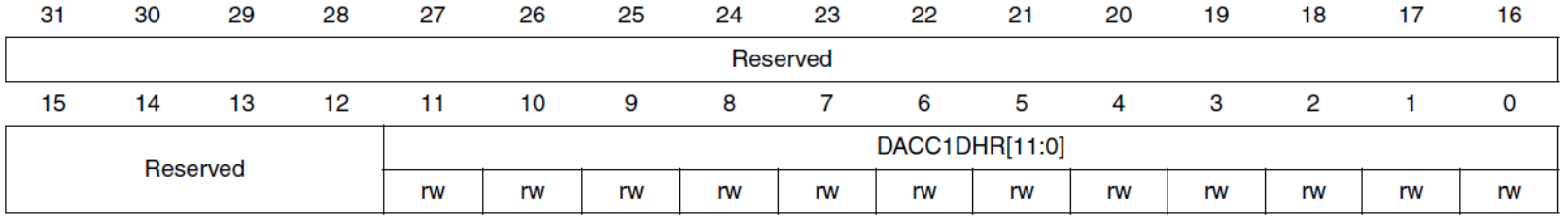
0: Software trigger disabled

1: Software trigger enabled

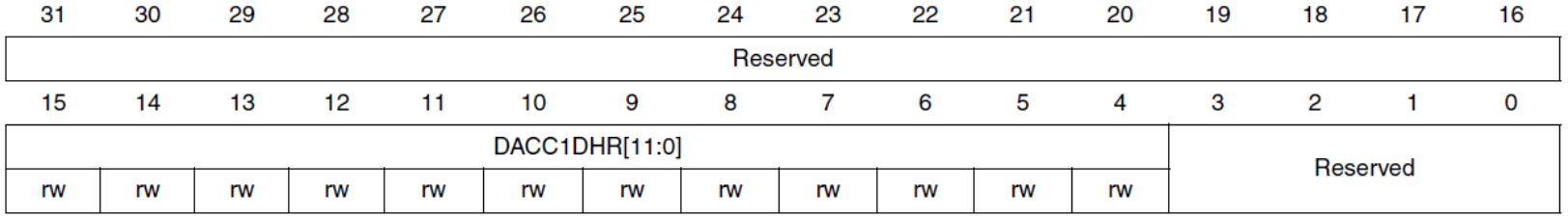
Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_DHR1 register value has been loaded into the DAC_DOR1 register.

DAC Channel 1 Data Registers

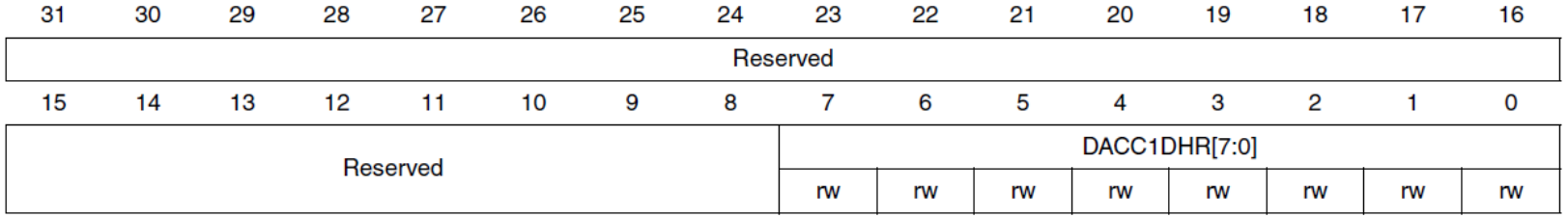
- DAC channel1 12-bit right-aligned data holding register (DAC_DHR12R1)



- DAC channel1 12-bit left aligned data holding register (DAC_DHR12L1)



- DAC channel1 8-bit right aligned data holding register (DAC_DHR8R1)



DAC Channel1 Data Output Register (DAC_DOR1)

Reset value: 0x0000 0000

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				DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:12 Reserved.

Bit 11:0 **DACC1DOR[11:0]**: DAC channel1 data output

These bits are read-only, they contain data output for DAC channel1.

TIMx Control Register 2 (TIMx_CR2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	Reserved		
								rw	rw	rw	rw	rw			

Bits 6:4 **MMS**: Master mode selection

These bits allow to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows:

000: **Reset** - the UG bit from the TIMx_EGR register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode) then the signal on TRGO is delayed compared to the actual reset.

001: **Enable** - the Counter enable signal, CNT_EN, is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode.

When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected (see the MSM bit description in TIMx_SMCR register).

010: **Update** - The update event is selected as trigger output (TRGO). For instance a master timer can then be used as a prescaler for a slave timer.

011: **Compare Pulse** - The trigger output send a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or a compare match occurred. (TRGO)

100: **Compare** - OC1REF signal is used as trigger output (TRGO)

101: **Compare** - OC2REF signal is used as trigger output (TRGO)

110: **Compare** - OC3REF signal is used as trigger output (TRGO)

111: **Compare** - OC4REF signal is used as trigger output (TRGO)

Assignments

- ARM Project #5