**Introduction**

* **12-bit** voltage output DAC
* can be 8-bit or 12-bit mode in conjunction with DMA controller
* two output channels with each converter
* Noise-Wave generation
* Triangular-Wave generation
* Dual DAC channel independent or simultaneous conversions
* DMA capability for each channel
* external triggers for conversion
* Input Voltage reference VREF+­S
* PA4 and PA5 pins should be configured as Analog Input

**Functional Description**

**DAC channel enable**

* setting the ***ENx*** bit in the register ***DAC\_CR***
* takes some time to convert (tWAKEUP)

**DAC output buffer enabled**

* two output buffers to reduce output impedance
* can be enabled by ***BOFFx*** bit in register ***DAC\_CR***

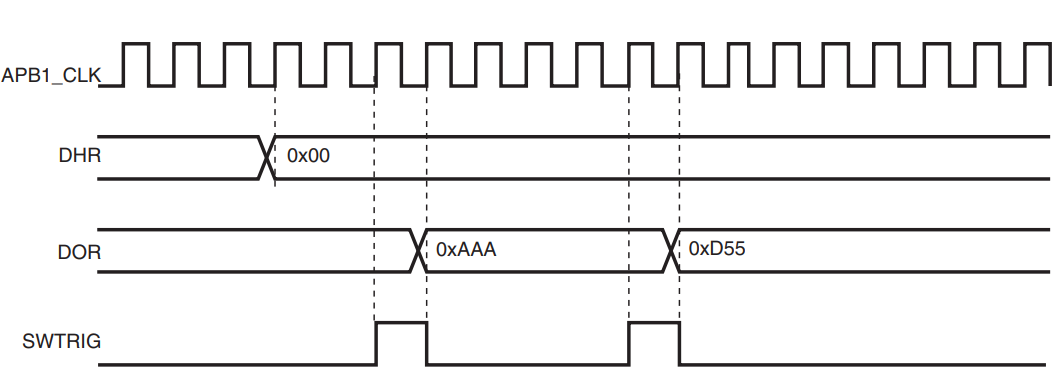
**DAC data format**

* **Single DAC channel**
  1. 8-bit right alignment
     + user has to load data into DAC\_DHR8Rx[7:0] bits stored into DHRx[11:4] bits
  2. 12-bit left alignment
     + user has to load data into DAC\_DHR12Lx[15:4] bits stored into DHRx[11:0] bits
  3. 12-bit right alignment
     + user has to load data into DAC\_DHR12Rx[11:0] bits stored into DHRx[11:0] bits
* **Dual DAC channel**
  1. 8-bit right alignment
     + for DAC channel 1 - user has to load data into DAC\_DHR8RD[7:0] bits stored into DHR1[11:4] bits
     + for DAC channel 2 - user has to load data into DAC\_DHR8RD[15:8] bits stored into DHR2[11:4] bits
  2. 12-bit left alignment
     + for DAC channel 1 - user has to load data into DAC\_DHR12LD[15:4] bits stored into DHR1[11:0] bits
     + for DAC channel 2 - user has to load data into DAC\_DHR12LD[31:20] bits stored into DHR2[11:0] bits
  3. 12-bit right alignment
     + for DAC channel 1 - user has to load data into DAC\_DHR12RD[11:0] bits stored into DHR1[11:0] bits
     + for DAC channel 2 - user has to load data into DAC\_DHR8RD[27:16] bits stored into DHR2[11:0] bits

**DAC Trigger Selection**

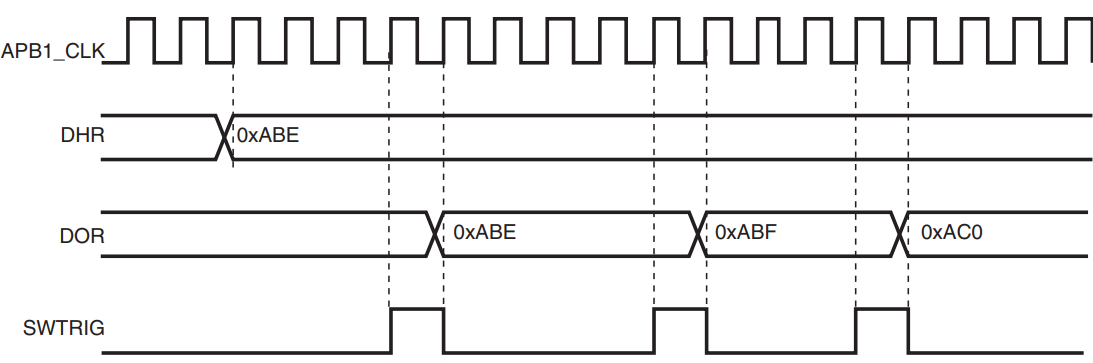
* **TENx** bit is set to trigger the DMA to convert on external events such as timer counter, external interrupt line
* **TSELx[2:0]** determine which possible event to select
* **ENx** must be disabled before changing TSELx[2:0]
* **Note :** software trigger is done by setting ***SWTRIG*** bit

**Noise Generation**



* generate variable-amplitude pseudonoise – a Linear Feedback Shift Register is available
* selected by setting ***WAVEx[1:0]*** bits to 01
* LSFR value can be masked via ***MAPMx[3:0]*** bit in register ***DAC\_CR***
* DAC trigger must be enabled for noise generation by setting ***TENx*** bit in ***DAC\_CR***

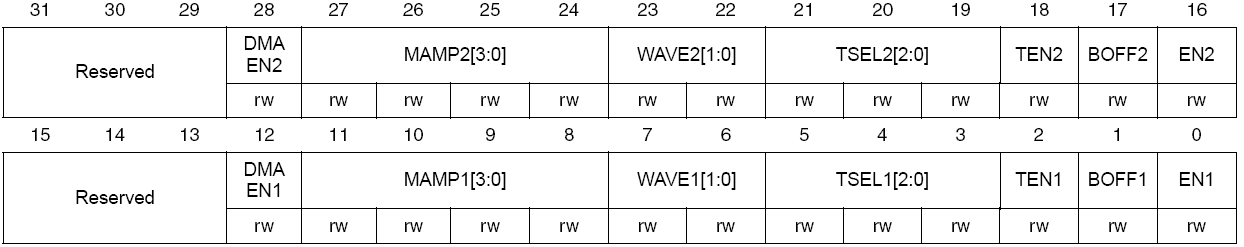
**Triangle-wave Generation**



* small amplitude triangular waveform on DC
* selected by setting ***WAVEx[1:0]*** to 10
* amplitude is configured by ***MAMPx[3:0]***
* DAC trigger must be enabled for noise generation by setting ***TENx*** bit in ***DAC\_CR***

**DAC Registers**

**DAC control register (DAC\_CR)**



**MAMPx[3:0]**

* DAC channel mask/ amplitude selctor
  + select mask in wave generate mode
  + select amplitude in triangle generation mode

|  |  |  |
| --- | --- | --- |
| **MAMPx[3:0]** | **Wave generate mode** | **Triangle generation mode** |
| **0000** | Unmask bit0 of LFSR | Triangle amplitude == 1 |
| **0001** | Unmask bits[1:0] of LFSR | Triangle amplitude == 3 |
| **0010** | Unmask bits[2:0] of LFSR | Triangle amplitude == 7 |
| **0011** | Unmask bits[3:0] of LFSR | Triangle amplitude == 15 |
| **0100** | Unmask bits[4:0] of LFSR | Triangle amplitude == 31 |
| **0101** | Unmask bits[5:0] of LFSR | Triangle amplitude == 63 |
| **0110** | Unmask bits[6:0] of LFSR | Triangle amplitude == 127 |
| **0111** | Unmask bits[7:0] of LFSR | Triangle amplitude == 255 |
| **1000** | Unmask bits[8:0] of LFSR | Triangle amplitude == 511 |
| **1001** | Unmask bits[9:0] of LFSR | Triangle amplitude == 1023 |
| **1010** | Unmask bits[10:0] of LFSR | Triangle amplitude == 2047 |
| **>= 1011** | Unmask bits[11:0] of LFSR | Triangle amplitude == 4095 |

**WAVEx[1:0] – DAC channel x noise/triangle wave generation enable**

* only when TENx bit is 1

|  |  |
| --- | --- |
| **WAVEx[1:0]** | **Wave generation mode selection** |
| **00** | Wave generation disabled |
| **01** | Noise wave generation enabled |
| **1x** | Triangle wave generation enabled |

|  |  |  |
| --- | --- | --- |
| **Source** | **Type** | **TSELx[3:0]** |
| Timer 6 TRGO event | Internal signal from on-chip timer | 000 |
| Timer 8 TRGO event | Internal signal from on-chip timer | 001 |
| Timer 7 TRGO event | Internal signal from on-chip timer | 010 |
| Timer 5 TRGO event | Internal signal from on-chip timer | 011 |
| Timer 2 TRGO event | Internal signal from on-chip timer | 100 |
| Timer 4 TRGO event | Internal signal from on-chip timer | 101 |
| EXTI line9 | External pin | 110 |
| SWTRIG | Software Control bit | 111 |

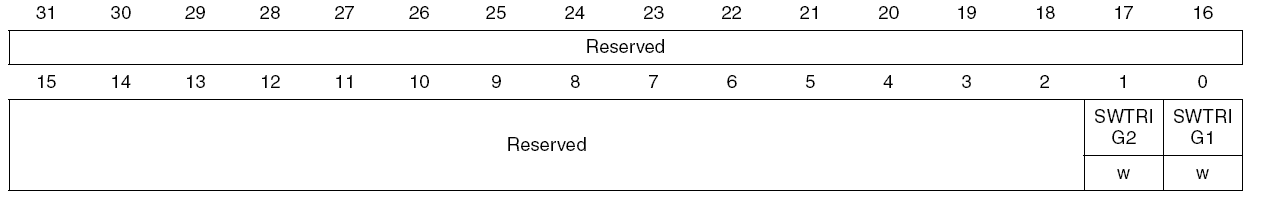
**BOFFx – DAC channel x output buffer enable/disable**

* 1 – output buffer disable

**ENx - DAC channelx ENABLE**

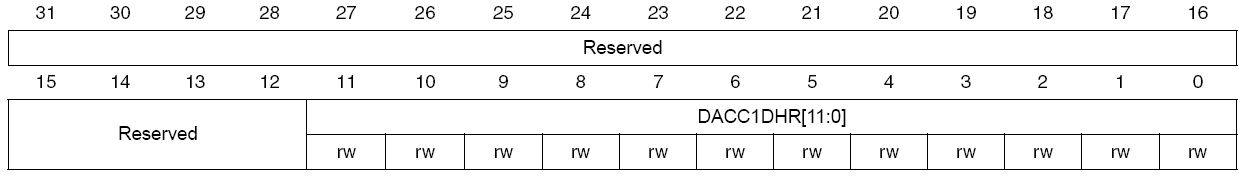
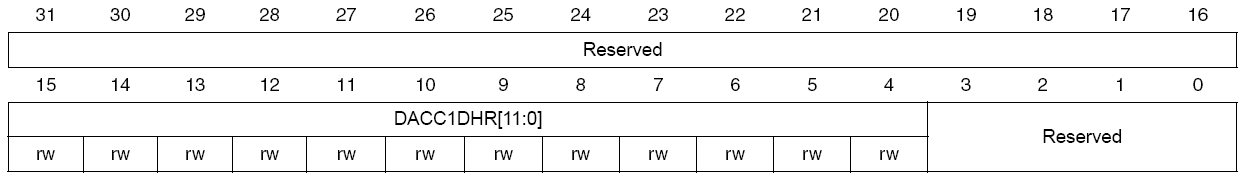
* 1 – DAC channelx disable

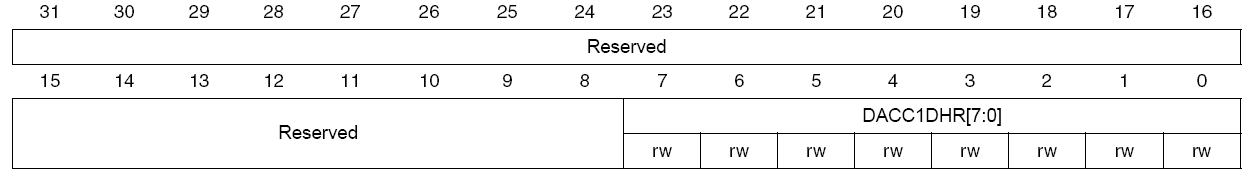
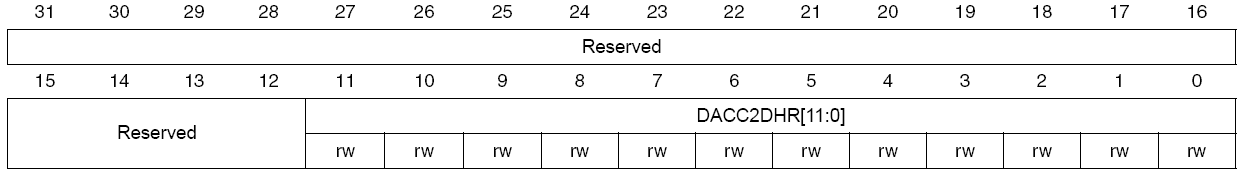
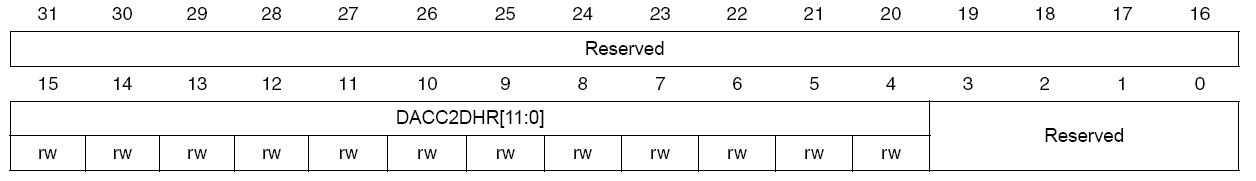
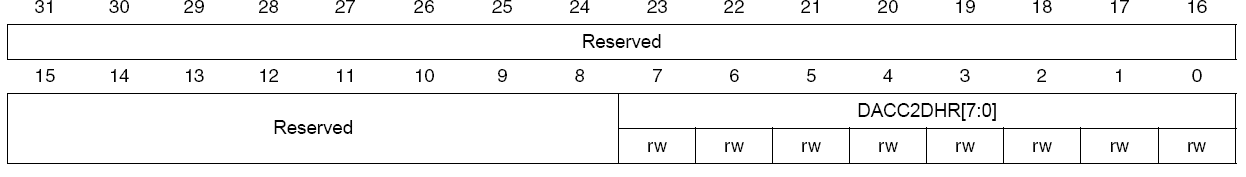
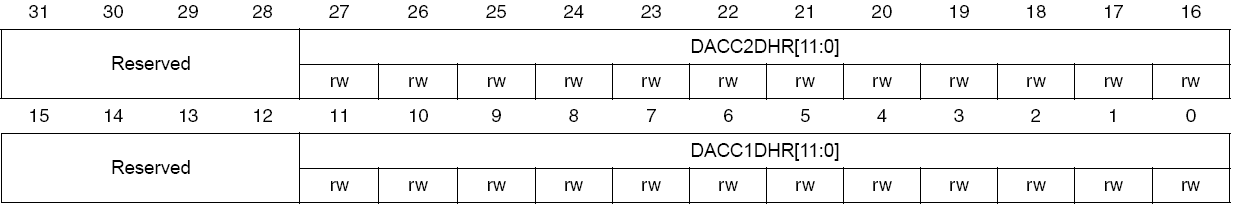
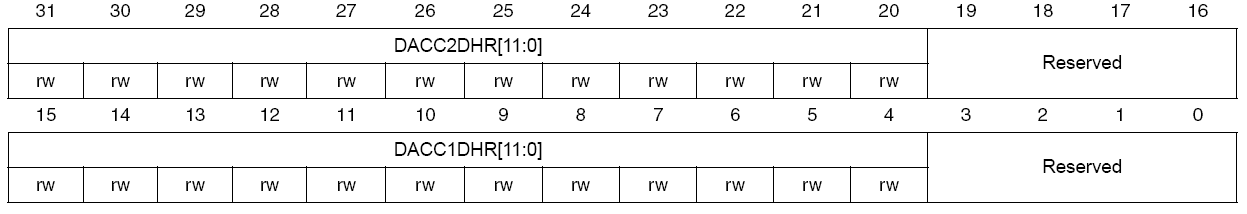
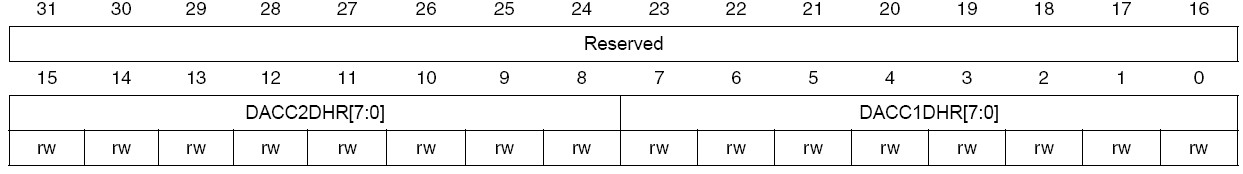
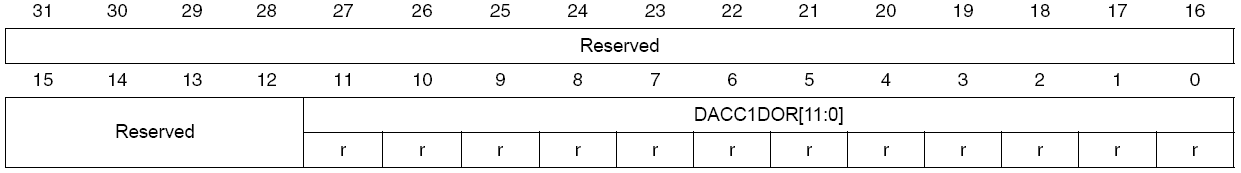
**DAC software trigger register (DAC\_SWTRIGR)**



**SWTRIGx – DAC channelx software trigger**

* set and cleared to disable and enable software trigger
* reset by hardware once DAC does operation
* 1 – Software trigger enabled

**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 channel2 12-bit right aligned data holding register (DAC\_DHR12R2)****DAC channel2 12-bit left aligned data holding register (DAC\_DHR12L2)****DAC channel2 8-bit right-aligned data holding register (DAC\_DHR8R2)****Dual DAC 12-bit right-aligned data holding register (DAC\_DHR12RD)****DAC 12-bit left aligned data holding register (DAC\_DHR12LD)****DUAL DAC 8-bit right aligned data holding register (DAC\_DHR8RD)****DAC channel1 data output register (DAC\_DOR1)****DAC channel2 data output register (DAC\_DOR2)**