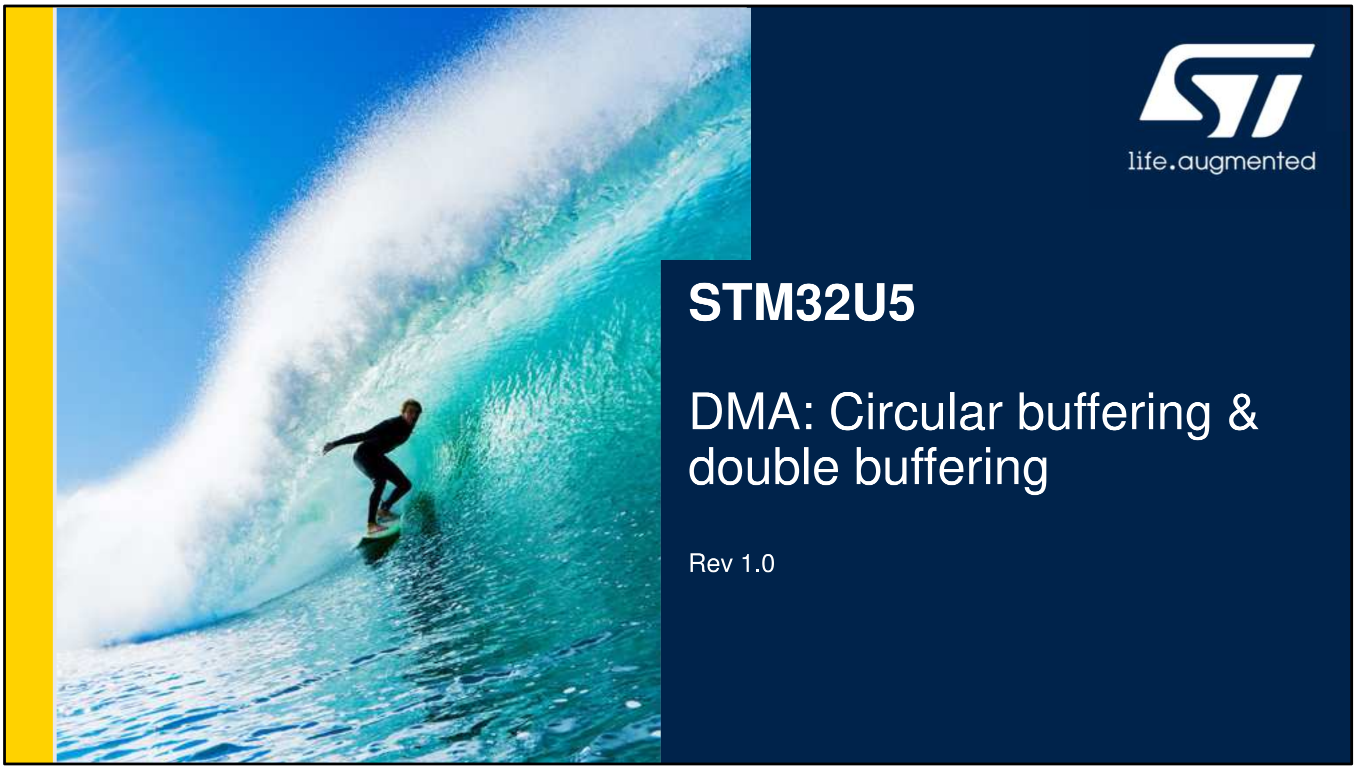
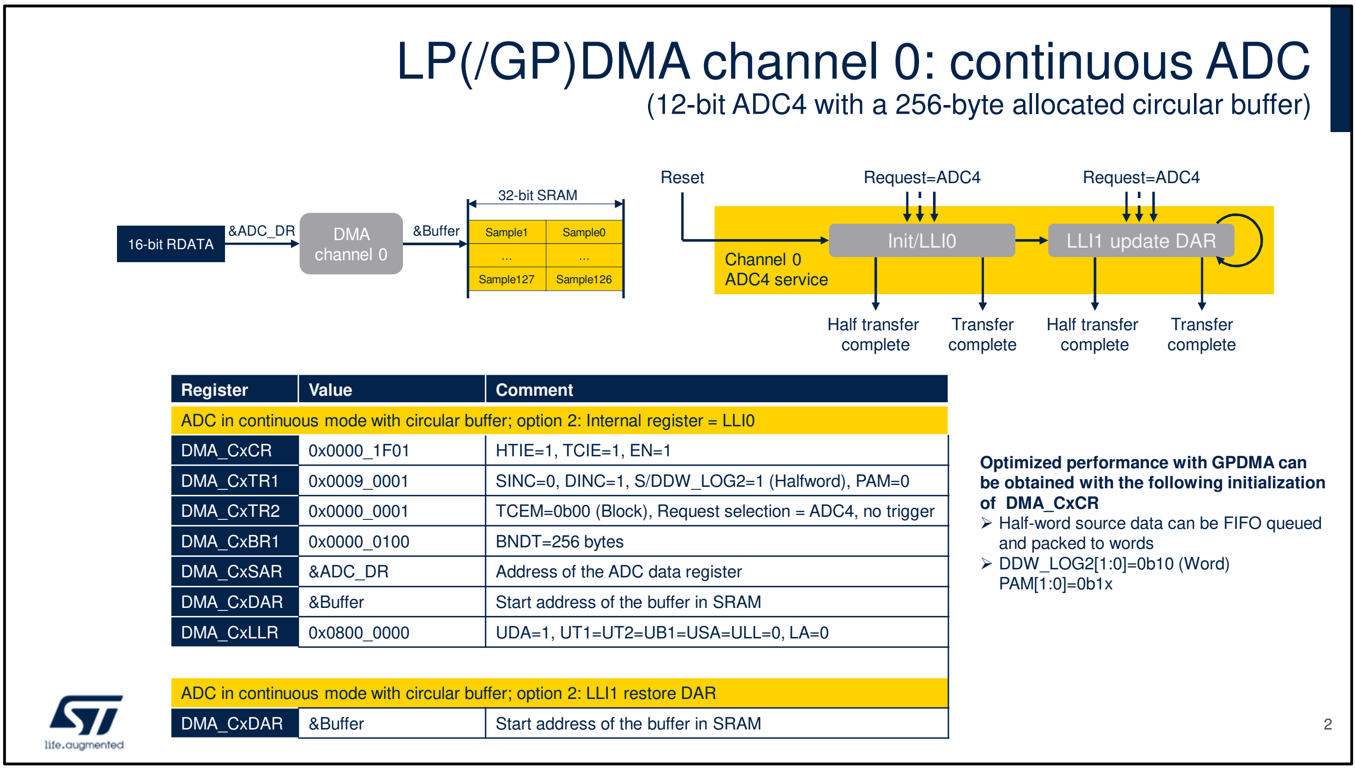


STM32U5

DMA: Circular buffering & double buffering

Rev 1.0

Hello, and welcome to this presentation, which describes the circular buffering and the double buffering supported by the LPDMA and the GPDMA.



LP(/GP)DMA channel 0: continuous ADC

(12-bit ADC4 with a 256-byte allocated circular buffer)

Register	Value	Comment
ADC in continuous mode with circular buffer; option 2: Internal register = LLI0		
DMA_CxCR	0x0000_1F01	HTIE=1, TCIE=1, EN=1
DMA_CxTR1	0x0009_0001	SINC=0, DINC=1, S/DDW_LOG2=1 (Halfword), PAM=0
DMA_CxTR2	0x0000_0001	TCEM=0b00 (Block), Request selection = ADC4, no trigger
DMA_CxBR1	0x0000_0100	BNDT=256 bytes
DMA_CxSAR	&ADC_DR	Address of the ADC data register
DMA_CxDAR	&Buffer	Start address of the buffer in SRAM
DMA_CxLLR	0x0800_0000	UDA=1, UT1=UT2=UB1=USA=ULL=0, LA=0
ADC in continuous mode with circular buffer; option 2: LLI1 restore DAR		
DMA_CxDAR	&Buffer	Start address of the buffer in SRAM

Optimized performance with GPDMA can be obtained with the following initialization of DMA_CxCR

- Half-word source data can be FIFO queued and packed to words
- DDW_LOG2[1:0]=0b10 (Word) PAM[1:0]=0b1x

life.augmented

2

The DMA is used to transfer 16-bit samples acquired by ADC4 to a buffer in SRAM containing 128 samples, with minimum software intervention.

In order to automatically reload the destination address register with the start address of the buffer, a linked-list containing two LLIs is implemented.

LLI0 provides all the required initializations, while LLI1 only restores the destination address register.

The CxLLR register is not updated after LLI0, so that each time the buffer is full, LLI1 restores only the destination address.

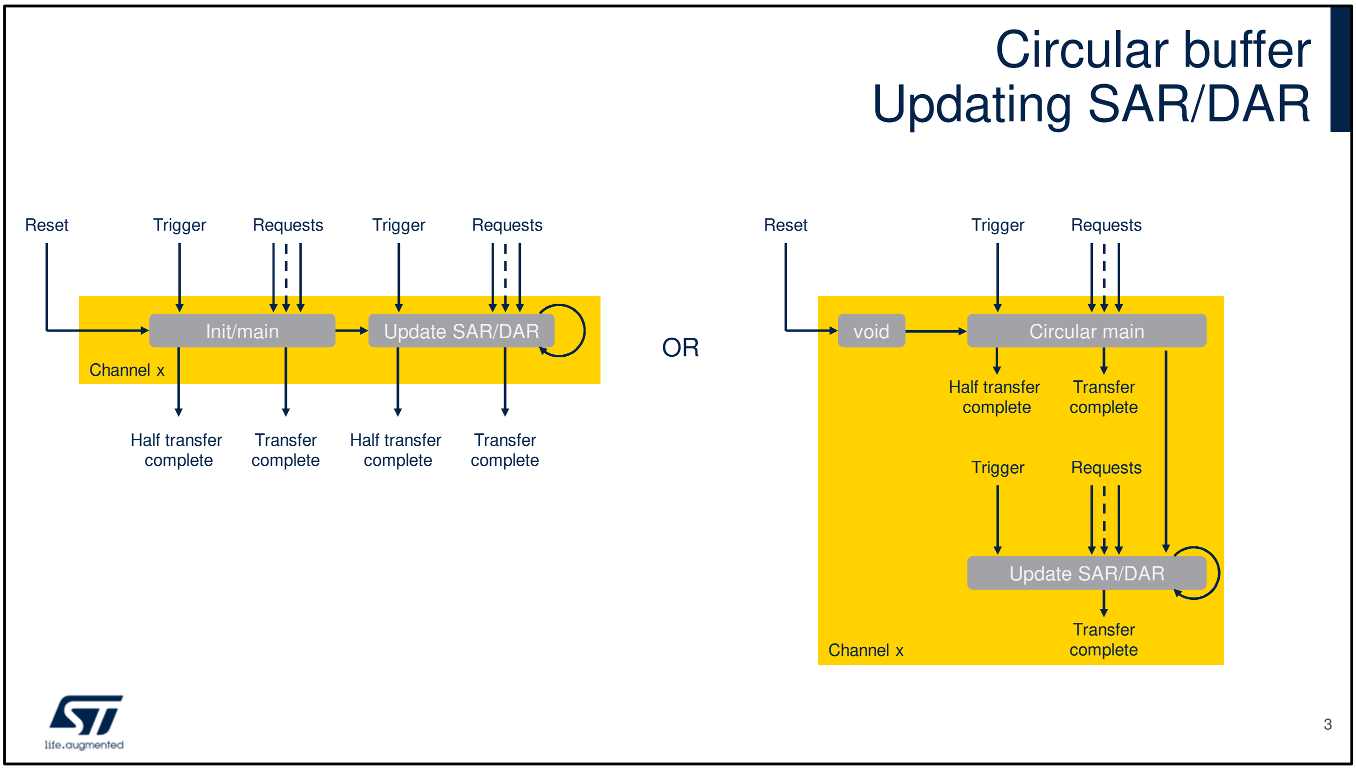
Two interrupts are enabled: half transfer complete and transfer complete.

Let us describe the initializations that apply to both GPDMA and LPDMA:

- In the CxCR register, interrupts are enabled, and the channel is activated
- In the CxTR1 register, the addressing modes are selected: fixed for the source address and incremented for the destination address. The source data and destination data widths are also selected: 16-bit halfword in both cases.
- In the CxTR2 register, the transfer complete event occurs when a block is transferred, the request input is selected: here it is ADC4 while no trigger is used.
- In the CxSAR register, the address of the ADC data register is programmed
- In the CxDAR register, the start address of the buffer in memory is programmed
- In the CxLLR register, the link is configured: only the DAR register is reloaded .

When the GPDMA is used, the configuration can be modified to improve the performance.

Instead of moving halfwords to the buffer in memory, the channel FIFO collects halfwords, packs them into words and writes words to the buffer. This is achieved by programming the destination data width to be a word and setting the PAM field to 1x in order to enable the source data FIFO queuing and packing up to the destination data width.



Circular buffer
Updating SAR/DAR

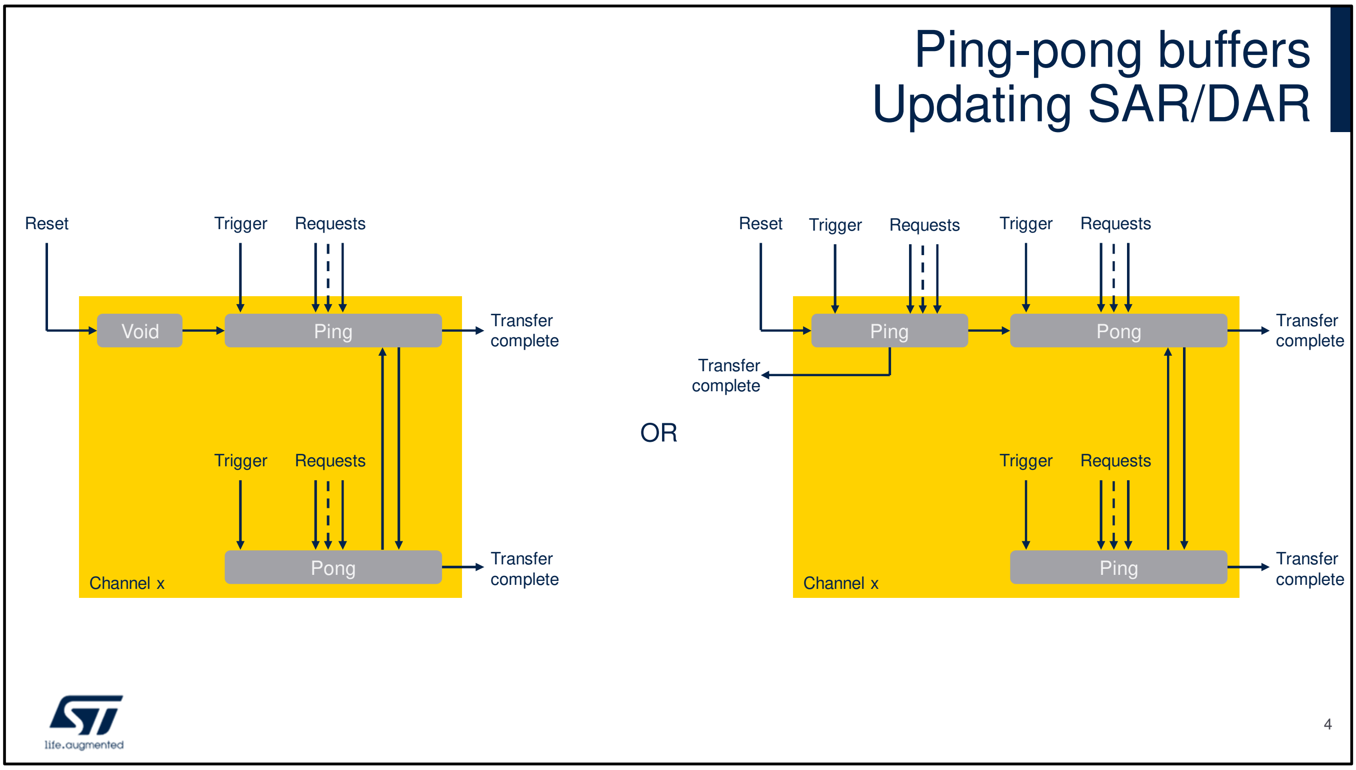
This slide describes the automatic update of the source or destination address register when a unique circular buffer is used.
As a unique buffer is used, the half transfer complete interrupt is required to ensure that software has processed the received samples before the DMA overrides them with the new ones.

The automatic restoration of the source or destination address can be performed by two different approaches. The first one corresponding to the figure on the left was described in the previous slide: the main LLI0 configures directly the linked-list register of the channel. LLI0 data structure is not in memory.
Only LLI1 in memory restores the source or destination

address and is the only one executed.

An alternative approach is to have no data transfer described by the channel x linked-list register but the LLR register pointing to the LLI1.
LLI1 is the main data structure in the memory instead of being directly initialized in the register file and points to LLI2 data structure.
LLI2 only restores the source or destination address and is the only one executed.

If circular buffering must be executed after some other transfers over the shared DMA channel x, this second approach is required.
The penultimate LLI N-1 in memory is needed to configure the first block transfer.
And the last LLI N restores the memory source or destination start address.



Ping-pong buffers
Updating SAR/DAR

Reset

Trigger

Requests

Void

Ping

Transfer complete

Trigger

Requests

Pong

Transfer complete

Channel x

OR

Reset

Trigger

Requests

Trigger

Requests

Ping

Pong

Transfer complete

Transfer complete

Trigger

Requests

Ping

Transfer complete

Channel x

life.augmented

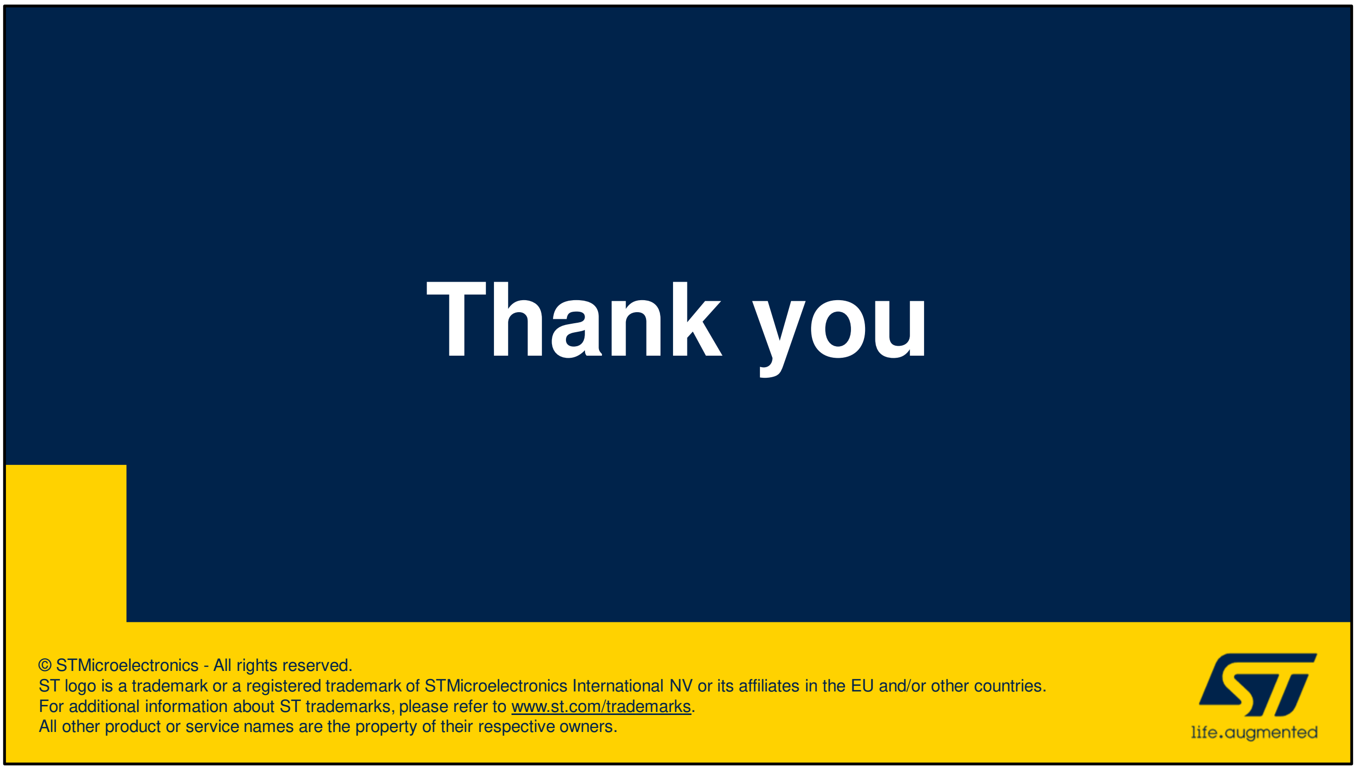
4

The management of a ping / pong buffer pair can be preferred vs a unique circular buffer.
The figure on the left describes the implementation without any first data transfer via the register file and only a link transfer for the loading of the LLI ping data structure.
When the first request and possibly trigger is received, the LLI related to the ping buffer is executed.
Once this buffer is full, the transfer complete interrupt is generated to inform the software that the ping buffer contains data ready to be processed.
The link transfer for loading the LLI related to the pong buffer also occurs when the ping buffer is full.

Once this pong buffer is fully executed, the transfer

complete interrupt informs the software that the pong buffer contains data ready to be processed.
The link to the LLI related to the ping buffer also occurs when the pong buffer is full. Then the same sequence repeats.

The figure on the right describes a sequence in which LLI0 directly configures the DMA to execute the ping buffer and afterwards load the LLI pong buffer from memory.
With the double buffer management case, when the pong buffer is full, a second LLI in memory to describe the ping transfer is always needed.



Thank you

life.augmented

In addition to this presentation, you can refer to the other presentations on the GPDMA and LPDMA:

- DMA overview
- DMA transfers hardware and software views
- Autonomous DMA & low power mode
- DMA linked list
- DMA 2D addressing
- DMA Register file
- DMA Error reporting
- DMA Input-output LLI control.