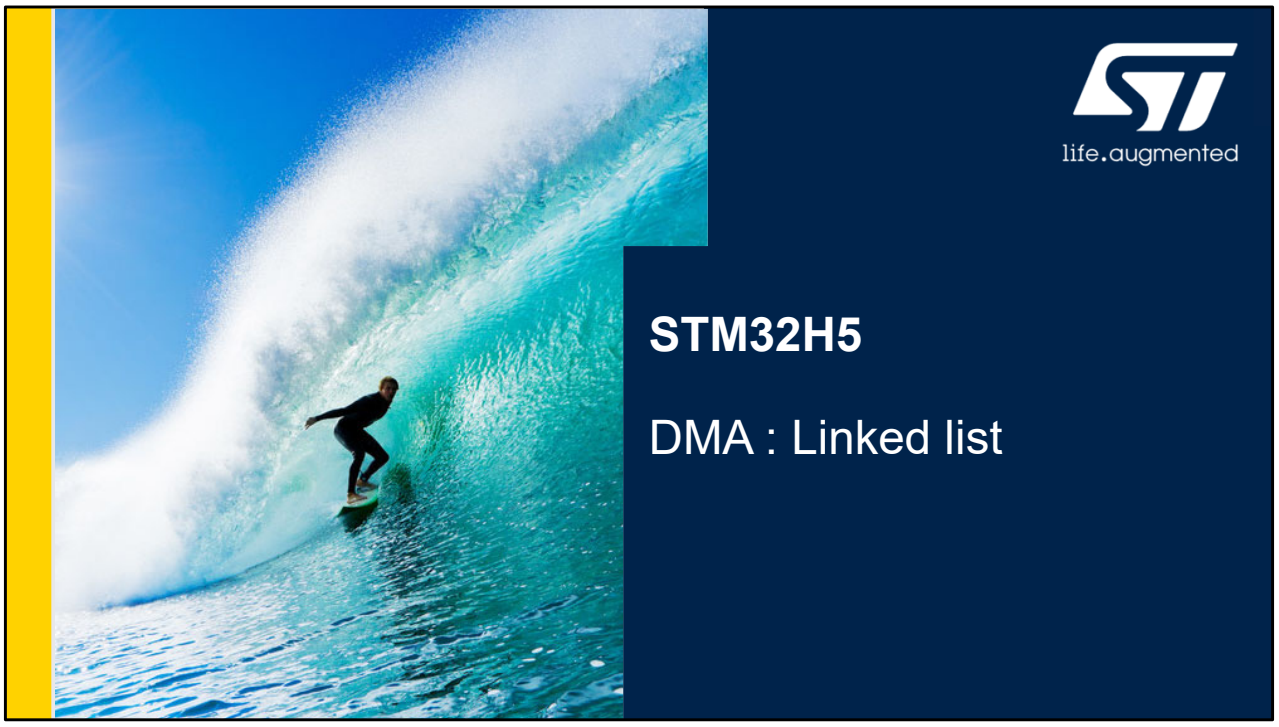


Hello, and welcome to this presentation, that describes the linked list management supported by the GPDMA.

Linked-list

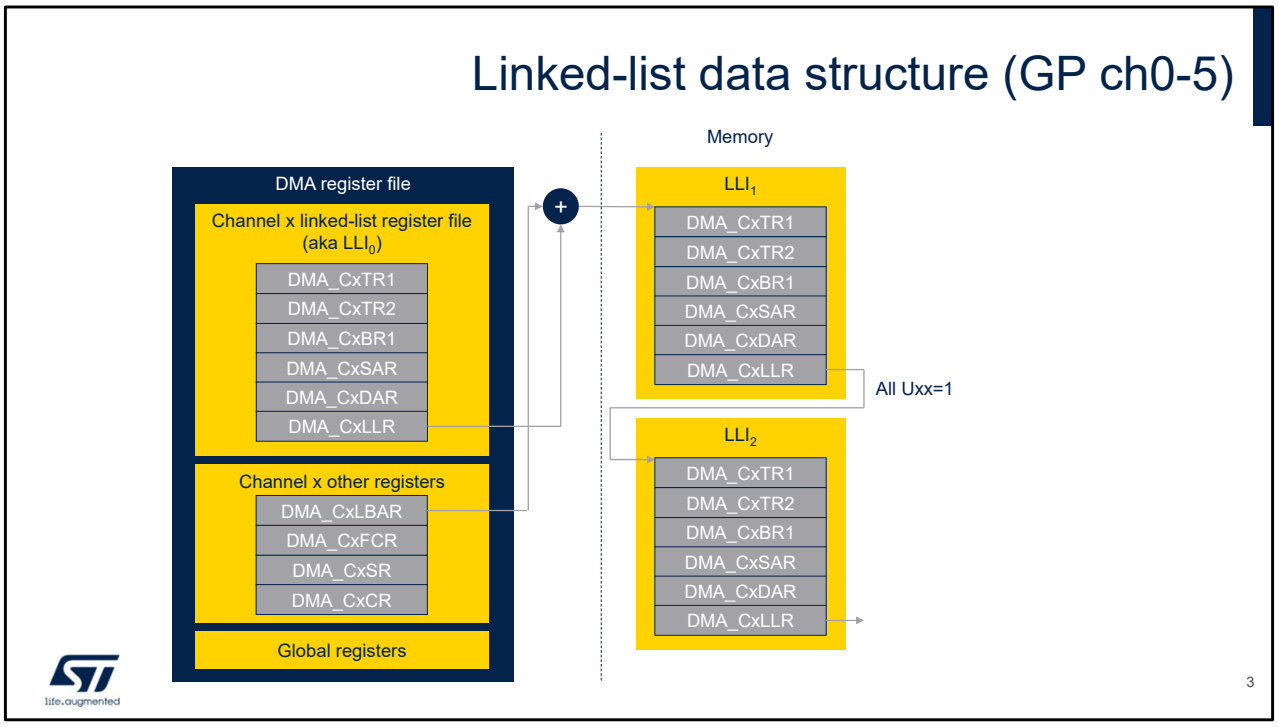
- Alternatively to the direct programming mode, a channel can be programmed by a list of transfers, known as a list of linked-list items (LLI)
  - Each Linked-list Item (a.k.a LLI) is defined by its data structure
  - The base address in memory of the data structure of a next linked-list item (LLIn+1) of a channel x is the sum of:
    - The (static) 64 KB-aligned link base address (i.e. DMA_CxLBAR)
    - The link address offset (LA[15:2] field of DMA_CxLLR) from the previous linked-list item (LLIn)
- A LLI is automatically transferred into the linked-list register file
- The data structure of each LLI may be specific and minimized to the difference between two successive links
  - A linked-list data structure is addressed following the value of the UT1, UT2, UB1, USA, UDA and ULL bits, plus UB2 and UT3 when present, in GPDMA_CxLLR

life.augmented

2

The first approach to configure a DMA transfer consists in initializing the channel memory-mapped registers. This is called direct programming mode.
However, the same channel can only be reconfigured when the ongoing transfer is completed.
In order to decouple software preparation of transfers from their execution, the linked-list approach is more convenient.
Each item in this linked-list is called a Linked-List Item or LLI.
The LLI is mapped in memory and contains an image of the values to be initialized into the DMA channel registers.
Thus, DMA channel registers programming becomes an indirect operation.

The base address in memory of the data structure of a next LLIn+1 of a channel x is the sum of the link base address of the channel x, which is a static value, and the link address offset, which is provided by the previous LLI. The data structure for each LLI may be specific and contains the list of channel registers to be initialized. By using the UT1, UT2, UB1, USA, UDA, ULL, plus UB2 and UT3 when present, the user can select which channel registers will be updated during the link. Therefore, only the difference between two consecutive LLIs has to be programmed.



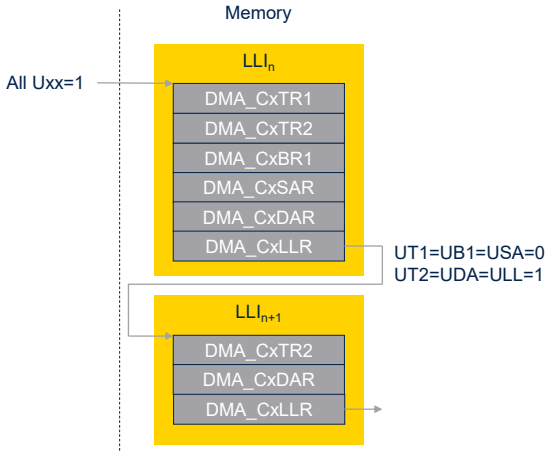
The left part of the figure represents the DMA register file.
The right part represents the LLI 1 and 2 allocated in memory.
This figure describes the link operation when GPDMA channels 0 to 5 are used.
The address of LLI1 is the sum of a base address programmed in the CxLBAR register and the offset present in the CxLLR register.
LLI n provides the offset used to locate LLI n+1.
When the link is performed, the channel x linked-list register file is updated from the values read in LLI1 data structure.
Later, when the transfer related to LLI1 will be completed, the LLI2 data structure will be loaded into the channel x

linked-list register file.
Since all Uxx control bits are assumed to be one, the entire channel x linked-list register file is updated.

Conditional update & compacted memory

Linked-list data structure (GP ch0-5)

Register	Purpose	Control bit that determines whether the register is updated during the link
DMA_CxTR1	Control register 1	DMA_CxLLR[UT1]
DMA_CxTR2	Control register 2	DMA_CxLLR[UT2]
DMA_CxBR1	Block register 1	DMA_CxLLR[UB1]
DMA_CxSAR	Source address register	DMA_CxLLR[USA]
DMA_CxDAR	Destination address register	DMA_CxLLR[UDA]
DMA_CxLLR	Linked-list address register	DMA_CxLLR[ULL]



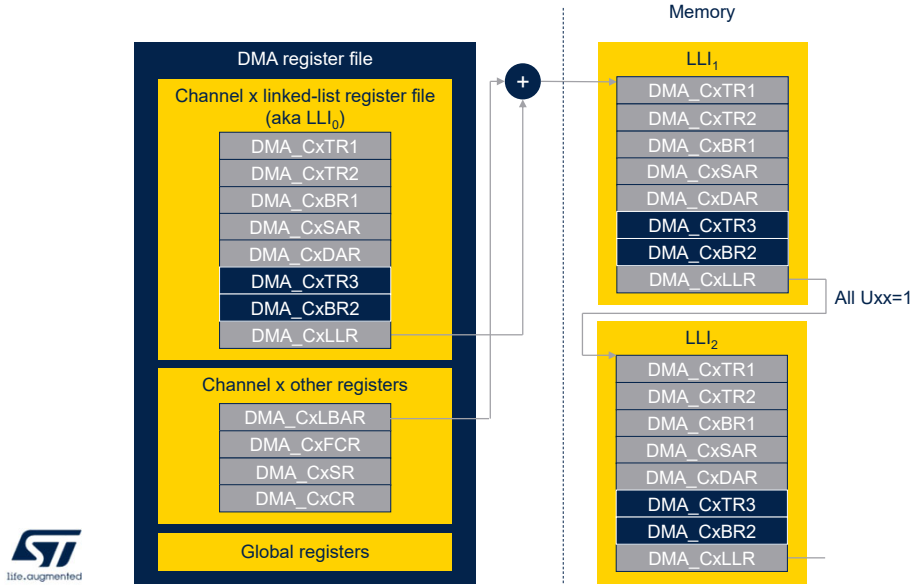
life.augmented

4

The CxLLR register contains control bits that enable the user to select which registers of the channel x linked-list register file will be updated during the link.
In the example on the right, it is assumed that only CxTR2, CxDAR and CxLLR register will be updated when linking from LLIn to LLIn+1.
Control bits related to registers that are not updated are cleared, these are UT1, UB1 and USA.
Control bits related to registers that are updated are set to one, these are UT2, UDA and ULL.
This capability of selecting the registers that are updated is very useful to minimize the memory footprint and accelerate the link when consecutive LLIs share common configuration settings.

For example, when two consecutive transfers are performed from the same source peripheral, it is not necessary to reprogram the source address.

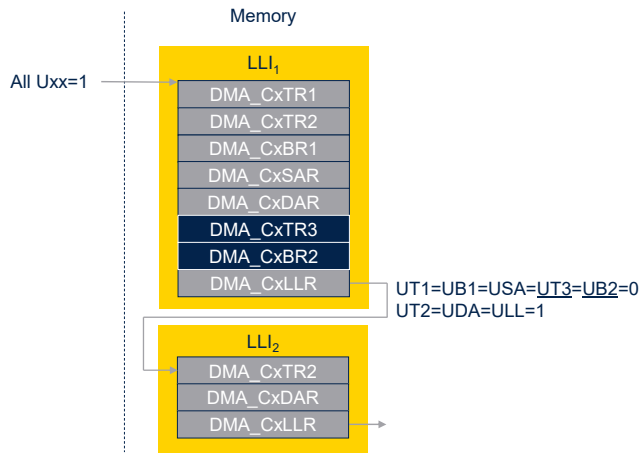
Linked-list data structure (GP ch6-7)



This figure describes the link operation when GPDMA channels 6 to 7 are used.
These channels support additional functionalities, such as block repetition, inter-burst offset and inter-block offset, that rely on two registers called CxTR3 and CxBR2.
Thus, the LLI size is extended to 8 fields.
During the link, if all Uxx bits are set to one, the 8 registers of the Channel x linked-list register file are updated from the LLI present in memory.

Conditional update & compacted memory Linked-list data structure (GP ch6-7)

Register	Purpose	Control bit that determines whether the register is updated during the link
DMA_CxTR1	Control register 1	DMA_CxLLR[UT1]
DMA_CxTR2	Control register 2	DMA_CxLLR[UT2]
DMA_CxBR1	Block register 1	DMA_CxLLR[UB1]
DMA_CxSAR	Source address register	DMA_CxLLR[USA]
DMA_CxDAR	Destination address register	DMA_CxLLR[UDA]
DMA_CxTR3	Control register 3	DMA_CxLLR[UT3]
DMA_CxBR2	Block register 2	DMA_CxLLR[UB2]
DMA_CxLLR	Linked-list address register	DMA_CxLLR[ULL]



life.augmented

6

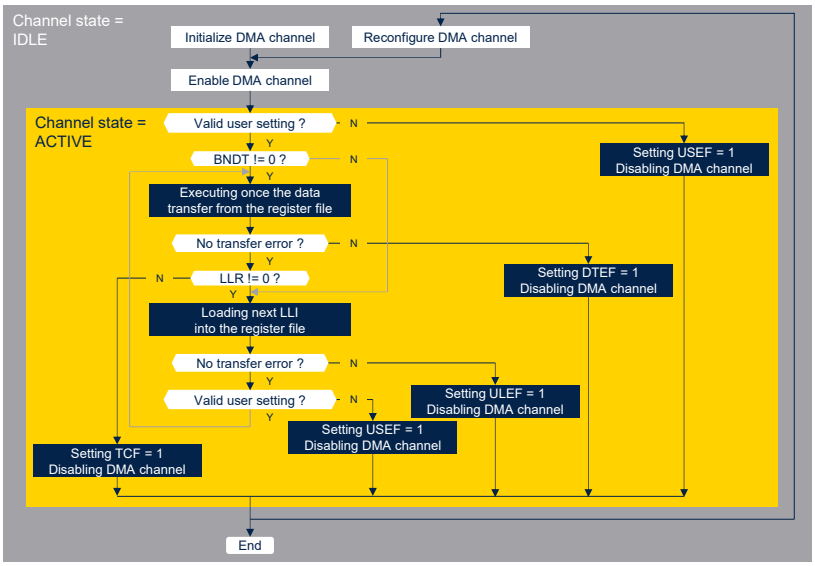
The CxLLR register of channels 6 and 7 supports two additional bits: UT3 and UB2, used to determine whether the CxTR3 and the CxBR2 registers will be updated during the link.

In the example on the right, it is assumed that only CxTR2, CxDAR and CxLLR register will be updated when linking from LLIn to LLIn+1.

Control bits related to registers that are not updated are cleared, these are UT1, UB1, USA, UT3 and UB2.

Control bits related to registers that are updated are set to one, these are UT2, UDA and ULL.

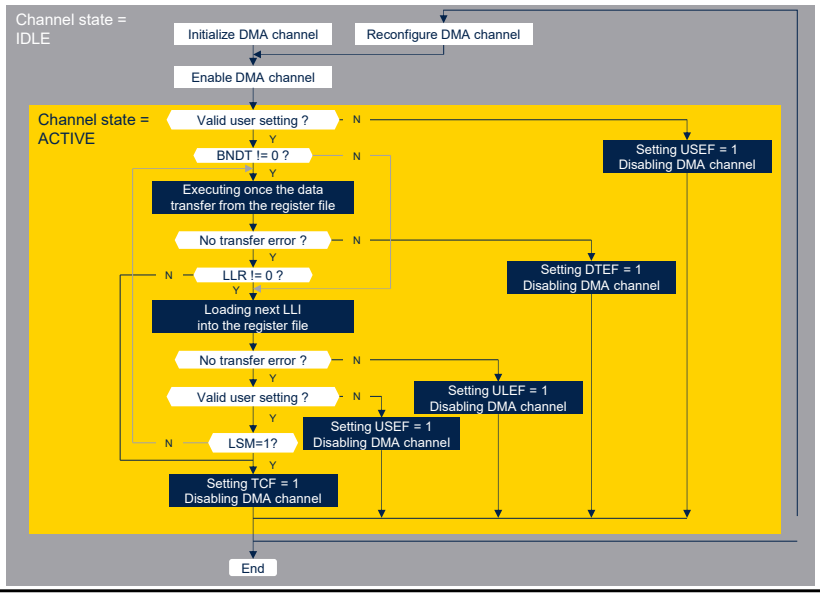
Channel (linked-list) programming Normal/Run-to-completion mode (LSM=0)



This figure depicts the DMA channel execution and its registers programming in run-to-completion mode.
In this mode, a full sequence of LLIs is processed by the DMA, without software intervention.
The first LLI is described by the content of the linked-list register file.
LLI0 may only perform a link transfer, if BNDT is null, but LLR register non null. BNDT stands for Block Number of data bytes to transfer.
LLI0 may consist of a first data transfer if BNDT is not null, followed by the next LLI1 loading if LLR register is not null. Then the next LLI is loaded and an iteration occurs. See the rising arrow between “valid user setting test” and “Executing once the data transfer from the register file”.

Initializing LLR with value 0 completes the channel transfer at the end of the currently active LLI.
Note that the DMA controller parses the initialization performed by software and raises a flag called USEF in case of a setting error.
The DMA controller raises a flag called DTEF when an error occurs during a data transfer and raises a flag called ULEF when an error occurs during a link.
The TCF flag is set to one when the channel transfer completes successfully.

Channel (linked-list) programming Whatever LSM



This figure shows the overall and unified GPDMA linked-list programming, whatever is the execution mode: run to completion or link step.

The software can reconfigure a channel when the channel is disabled and update the execution mode to change from/to run-to-completion mode to/from link step mode.

When LSM equals one, there is no automatic iteration. The channel transfer completes when the data transfer and link transfer, if any, of the current LLI are performed.

Software is in charge of reactivating the channel, after a possible reconfiguration, in order to handle the next LLI.

Thank you

life.augmented

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

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