Hello, and welcome to this presentation, that describes the DMA 2D addressing mode supported by GPDMA.
This is not to be confused with the Chrom-ART Accelerator controller, present in other STM32 microcontrollers, known as DMA2D, that also implements 2D addressing mode.

GPDMA ch6-7
Burst, block and 2D addressing

- Linear addressing mode
- Fixed addressing (typically for peripheral data register)
  - Contiguously-incremented addressing (typically for memory access)
    - Contiguous data blocks in memory (for a given LLIn), scanned from head to tail
  - Blocks up to 64kB (16-bit BNDT)
- 2D addressing mode (GPDMA1 & GPDMA2 ch6-7), additional
  - Repeated block mode: programmable repeated block counter (11-bit BRC, up to 2k blocks)
  - Programmable source/destination signed burst address offset (2x 14bit, up to +/-8kB)
    - Non-contiguous incremented/decremented addressing after each burst
  - Programmable source/destination signed block address offset (2x 17bit, up to +/-64kB)
    - Non-contiguous incremented/decremented addressing after each block

life.augmented

2

This slide sums up the addressing mode supported by GPDMA.
The GPDMA implements the following addressing modes:
- Fixed addressing, typically used to access a peripheral data register
- Contiguously-incremented addressing, typically used to access memory, in ascending address order
- Block maximum size is 64 kilobyte, due to the 16-bit field called BNDT, which stands for block number of data bytes to transfer.

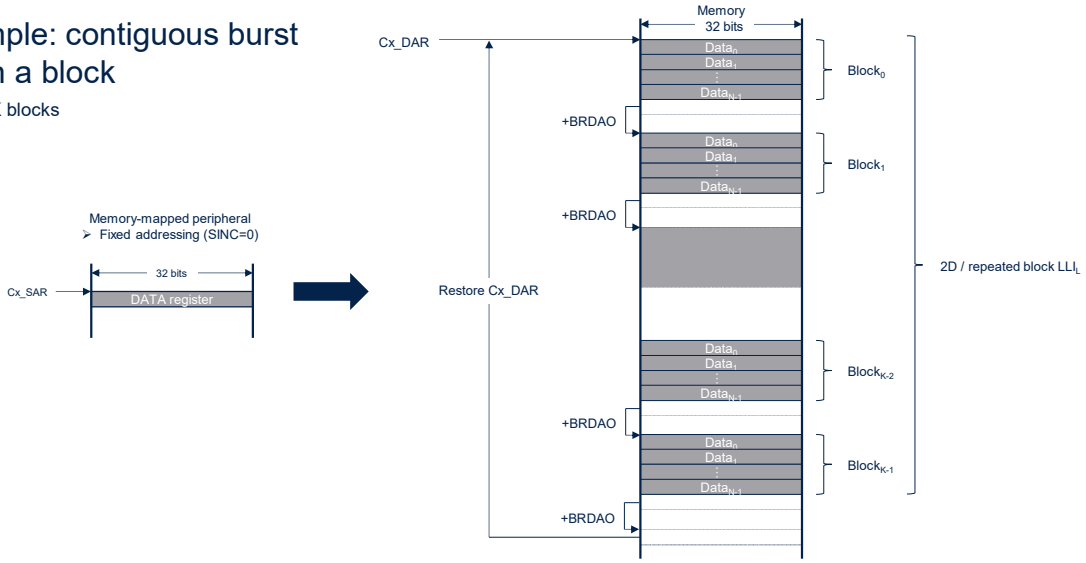
The channels 6 and 7 of the GPDMA1 and GPDMA2 implement additional addressing modes:
- Repeated block mode, based on a programmable counter

- Programmable source and destination signed burst address offset
- Programmable source and destination signed block address offset.

Thus, two programmable strides can be inserted: between consecutive bursts and between consecutive blocks.
It is to added that from a software driver point of view, the GPDMA2 channels 0 to 7 are registered as DMA channels 8 to 15.
From a software driver point of view, the 2D addressing capable DMA channels are the channels 6, 7 , 14 and 15.

2D addressing mode (continuous) (de)interleaving data/channels GP ch6-7

- Example: contiguous burst within a block
  - LLI: K blocks

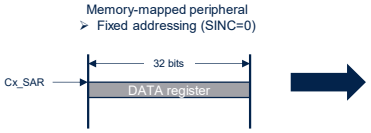


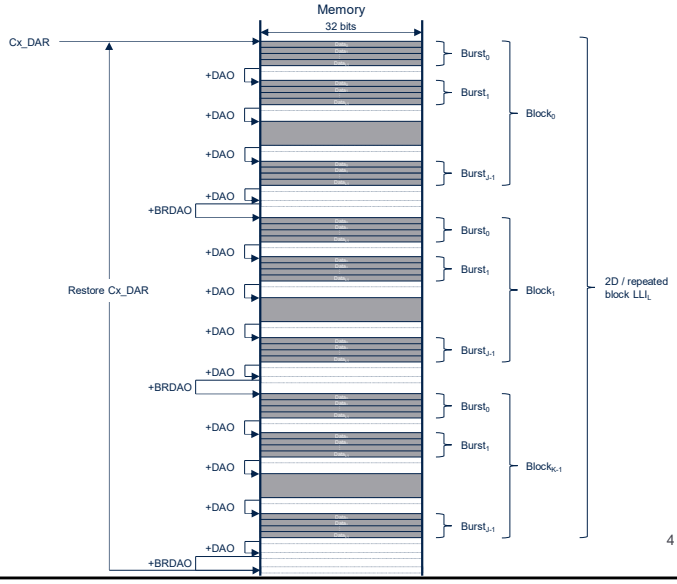
This figure highlights the repeated block mode and the stride between blocks.
This is useful to transfer the contents of a peripheral data register, typically an input FIFO, to non-contiguous buffers in memory.
After filling a block of N words, the DMA channel automatically adds the BDRA0 signed offset to the current address, thus jumping to the next buffer in memory.
At the end of the transfer, when K blocks will have been transferred, the Cx_DAR register, which points to the beginning of the first buffer, can be automatically restored, in order to implement circular buffers.
This automatic restoration requires a link operation.
In this example, bursts are placed contiguously in memory

so the burst destination address offset (the DAO field of the Cx_TR3 register) must be null.
For performance optimization, destination transfers should be programmed as 4-word bursts.

2D addressing mode (continuous) (de)interleaving data/channels GP ch6-7

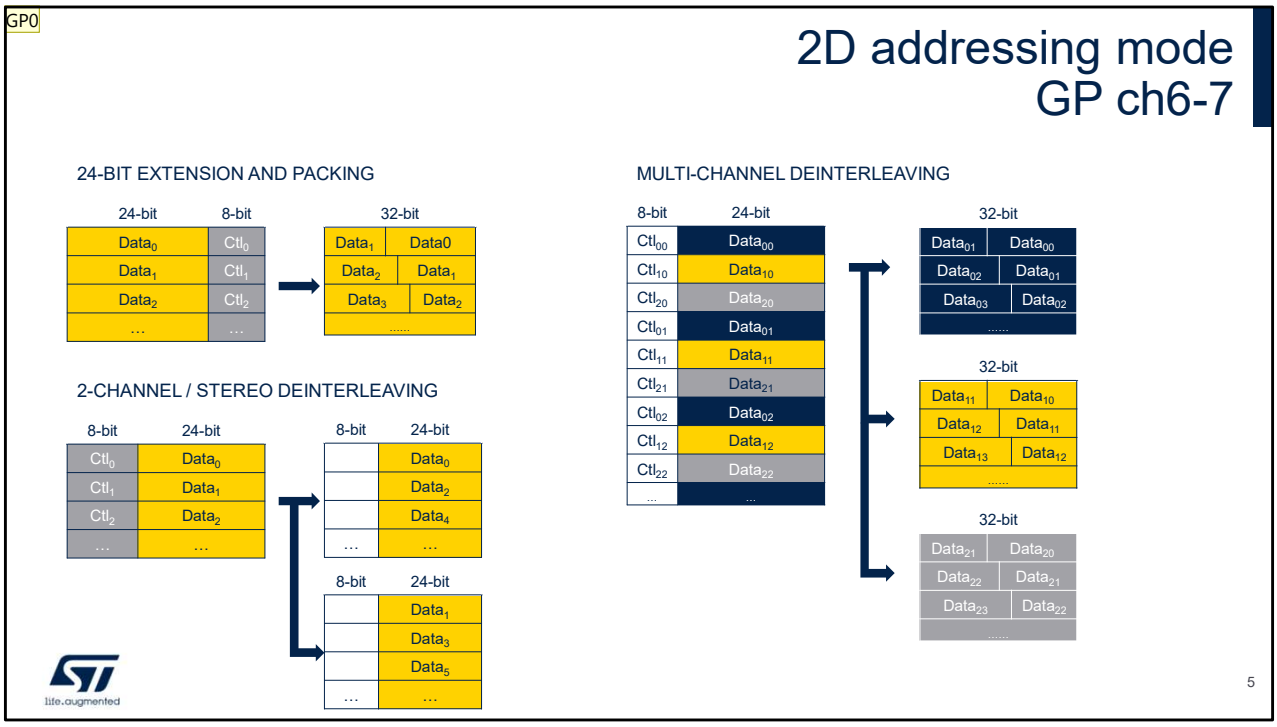
- Example: non-contiguous bursts within a block
  - Burst: I words
  - Block: J bursts
  - LLI: K blocks





4

This figure highlights the repeated block mode, with the stride between bursts and the stride between blocks.
The GPDMA transfers K blocks of J bursts, each burst containing I words.
The stride between blocks is useful to transfer the contents of a peripheral data register, typically an input FIFO, to non-contiguous buffers in memory.
The stride between bursts is useful to interleave or deinterleave data and, also to align data. Some use cases will be described in the next slide.



The first use case of burst stride is 24-bit extension and packing.

A source buffer contains data and control pairs. The GPDMA channel 6 to 7 are capable to extract the data field and to perform packing by storing the resulting 24-bit data back-to-back in the destination buffer.

On source, the following configuration is appropriate: a burst address offset of +1 byte and a burst length of 3 bytes.

No burst stride is needed on destination. Destination can be programmed as 4-word burst for best performances in writes.

The second use case consists in deinterleaving stereo audio samples into two separate buffers, one containing

the right samples, the other one the left samples.
There are several possible implementations.
One may be based on 2 GPDMA channels, one for left samples, one for right samples.
Then if the source buffer contains 2xN samples, the left/even samples may be read by a programmed source 1-word burst, with a source burst address offset of 4, and a source block size of 4xN bytes. And destination can be programmed as 4-word burst for best performances on writes.
An alternative implementation may allocate one single GPDMA channel.
A source with bursts of 1 word and a (source) burst address offset of +4.
A source with 2 blocks (i.e. BRC=1), one (source) block being 2x N bytes.
A destination with 4-word burst optimized writes, optionally with a (destination) block address offset.
In the last use case, data is deinterleaved and packed into 3 buffers. This could be the deinterleaving of RGB pixel color components.
Similarly, a single channel or 3 channels may be used.

Slide 5

GP0 Can we replace "The GPDMA channel 12 to 15 are capable to extract the data field and to perform packing by storing the resulting 24-bit data back-to-back in the destination buffer." with "Any GPDMA channel is capable ..."
GUILLAUME PERON, 2023-05-20T15:10:28.663

FC0 0 Replaced by channel 6 to 7
Francois CLOUTE, 2023-06-28T10:58:23.335

GP0 1 OK
GUILLAUME PERON, 2023-06-30T17:37:37.732

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 linked list
- DMA Circular buffering & double buffering
- DMA Register file
- DMA Error reporting
- DMA Input-output LLI control.