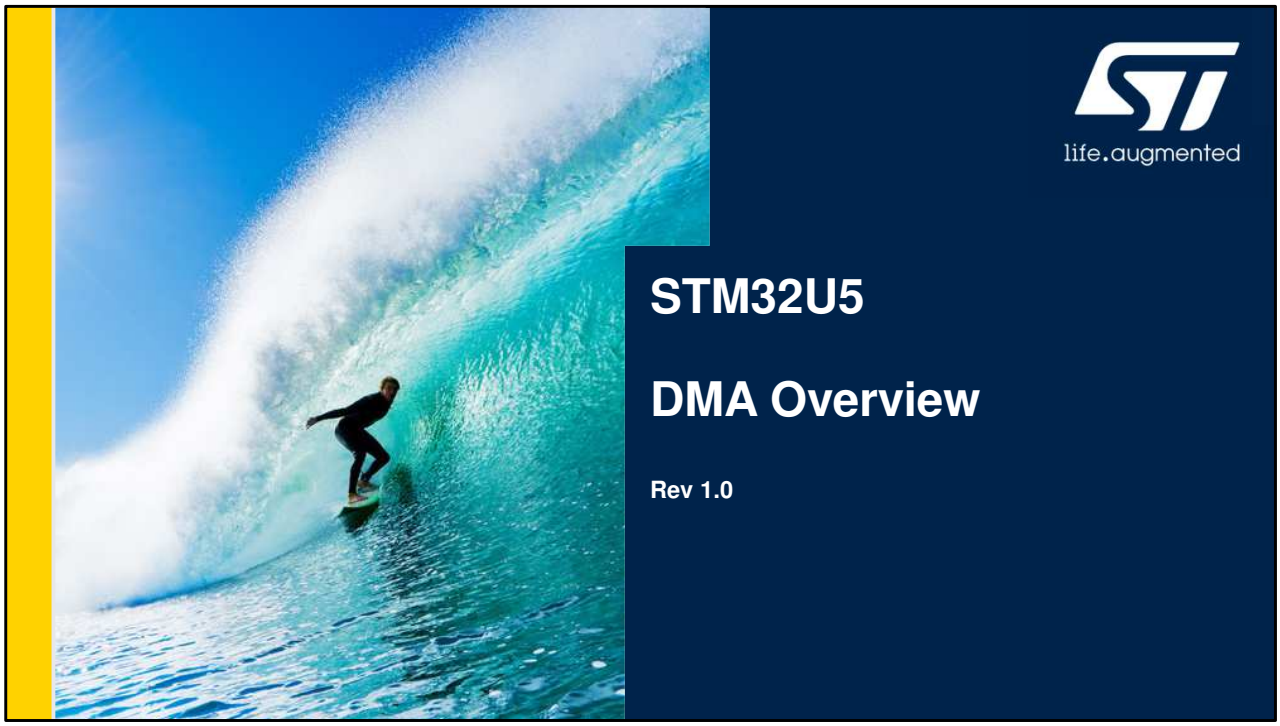
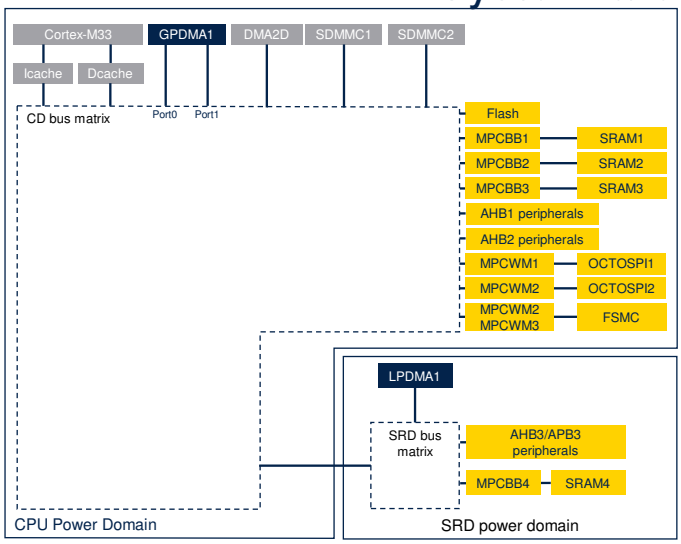


Hello, and welcome to this overview of GPDMA and LPDMA controllers embedded in the STM32U5.

System architecture & DMA overview



Application benefit

- Off-load CPU for data transfers from a memory-mapped source to a memory-mapped destination

- A new DMA module
  - Single DMA driver
  - 2 hardware instances
    - GPDMA
    - LPDMA
  - Linked-list based programming
  - Integrated DMAMUX features
  - Improved autonomy
    - Own clock request management
    - Flexible intra-channel and inter-channel input/output control

life.augmented

2

The STM32U5 embeds two new DMA controller modules: the General Purpose DMA (or GPDMA) and the Low Power DMA (or LPDMA).
The GPDMA belongs to the CPU power domain and supports the low-power background autonomous mode (or LPBAM) in stop 0 and stop 1 modes.
The LPDMA belongs to the Smart Run Domain and supports low-power background autonomous mode (or LPBAM) in stop 0, stop 1 and stop 2 modes.
These DMA controllers are in charge of data movement between memory-mapped locations, memory or peripherals, thus offloading the Cortex-M33 core.
GPDMA and LPDMA are two instances of the same IP, and are therefore controlled by the same software driver.

The DMA multiplexer, connecting the requests generated by peripherals to the channels is integrated in the module.
GPDMA and LPDMA support linked-list base programming to enable buffer chaining.
In LPBAM, these DMA controllers are able to temporarily request the clock when a request is received, in order to perform the transfer.
Also flexible intra-channel and inter-channel transfer chaining is supported, without software intervention.

DMA key features

- Bidirectional AHB master port(s): LPDMA: 1 port, GPDMA: 2 ports
- Memory-mapped data transfers from a source to a destination
  - Peripheral-to-memory
  - Memory-to-peripheral
  - Memory-to-memory
  - Peripheral-to-peripheral
- Autonomous data transfers during Sleep and Stop modes
- Concurrent DMA channels
- Transfers arbitration is based on a 4-grade priority policy
  - One reserved highest priority queue for time-sensitive traffic
  - Three lower priority queues with weighted round robin allocation

life.augmented

3

The LPDMA has a unique 32-bit AHB master port, while the GPDMA has two independent 32-bit AHB ports.
Transfers can be performed from peripheral to memory, from memory to peripheral, from memory to memory and from peripheral to peripheral.
The LPDMA and GPDMA support the LPBAM, that enables autonomous transfers during sleep and stop modes.
Channels are independent of each other and operate concurrently.
They are arbitrated through an algorithm based on a 4-grade priority policy:

- One reserved highest priority queue dedicated to time-sensitive traffic.

- Three lower priority queues implementing a weighted round-robin allocation.

Per-channel programming

- Status
  - Idle state
  - Events flag
    - Half transfer
    - Transfer complete
    - Error
      - User setting error
      - Link transfer error
      - Data transfer error
    - Trigger overrun
    - Suspended
  - Global (masked) interrupt status
  - FIFO level (for GPDMA only)

- Control
  - Start/enable
  - Suspend and resume
  - Reset
  - Abort and restart
  - Event flag clear
  - Interrupt enable (vs any event type)
- Static configuration
  - Security and privilege attributes
  - Priority
  - Link transfer allocated port (GPDMA only)
  - Execution mode
    - Run-to-completion (default)
    - (Single) Link-step mode

life.augmented

4

This slide and the following two describe the per-channel programming features.
The status information is visible to the software, this includes:

- Idle state
- Event flags
- Global masked interrupt status
- FIFO level for GPDMA.

The control information includes:

- A software start enable
- The capability of suspending and resuming a channel, resetting a channel and aborting and restarting a channel
- Event flag clearing

- Interrupt masking.

Some functionalities are statically configured and maybe locked until the next reset. These are:

- The security and privilege attributes
- The port used to access the Linked-list Items (or LLI)
- The execution mode: either run to completion, possibly executing chained transfers without software intervention, or link-step mode that requires a software re-enable each time a new LLI has to be processed.

Per-channel programming

- Linked-list, for chaining DMA services (a.k.a. data transfers) over a single channel
  - Each linked-list item (LLIn) is defined by a (linked-list) data structure in memory
  - Each LLIn execution consists of
    1. The (optional) link transfer: (next) LLIN+1 is loaded by the DMA hardware and updates its (linked-list) register file (aka LLI0)
    2. The (optional) data transfer: block-level transfer

life.augmented

5

Transfer chaining for a particular channel is performed through a linked-list.
Each item of the linked list is called a Linked-List Item or LLI, which is a structure allocated in memory.
The LLI has to be initialized with the values to transfer to the control registers when the chaining occurs.
Each LLI execution consists of:

- The optional link transfer, that loads the next LLI and updates the control registers accordingly
- The optional data transfer, with a granularity called a block.

Per-channel programming

- Per-LLI programming
  - Address
    - Source & destination start address
    - Source & destination addressing mode
    - Source & destination address offset between bursts (GPDMA ch12-15 only)
  - Data
    - Number of data bytes from the source, defining the block level
    - Number of repeated blocks (GPDMA ch12-15 only)
    - Source & destination data width and data handling
    - Source & destination burst length (GPDMA only)
  - Source & destination allocated port (GPDMA ony)
  - Source & destination security attribute
  - Input/output control
    - Peripheral request type and selection
    - Trigger mode and selection
    - Transfer complete event generation

life.augmented

6

For each LLI, the user has to program:

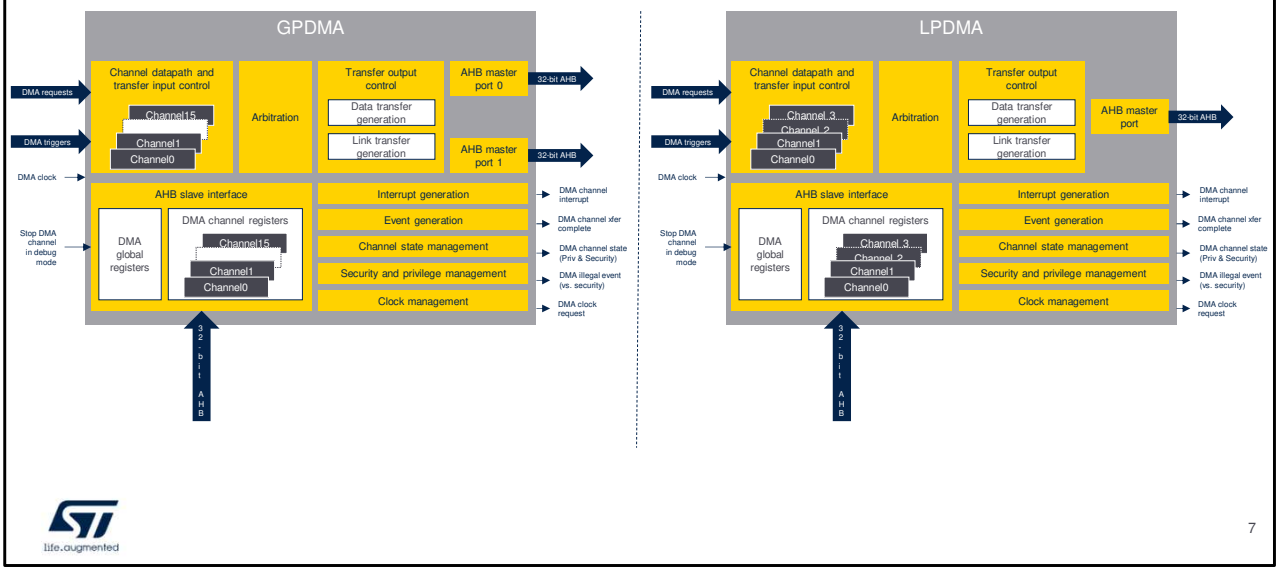
- The source and destination address as well as addressing modes and address offsets between bursts in 2D mode
- The information related to the data transfer: block size, source and destination data width and data handling. The GPDMA supports additional settings: number of repeated blocks for channels 12 to 15 and burst length for all channels.

Regarding the GPDMA, the AHB port used to access the source and destination locations have to be chosen.
The security attribute is also selectable when the channel itself is programmed as secure.
Finally, the input and output signals of the DMA controller

has to be configured:

- Peripheral request type and selection
- Trigger mode and selection
- Transfer complete event generation.

DMA block diagram



The block diagrams of GPDMA and LPDMA are very similar.
Requests and triggers are received on the left. They are used to activate channels that are then arbitrated.
The transfer output control is in charge of generating the data transfer and the link transfer, which consists in loading the linked-list item.
The DMA controllers also have an AHB slave interface to enable an access to the global registers and to the channel registers.
The DMA controller issues interrupt requests and DMA channel transfer complete events.
Each transfer generated by the DMA controller is tagged with a particular privilege and security attributes, which are

fully programmable.
In the case of a security violation, an output of the DMA controller is connected to the global trustzone controller.
The DMA controller can generate a clock request output signal to the RCC, whenever the device is in Run, Sleep or Stop mode.
When the microcontroller enters debug mode with the core halted, any channel can either be individually continued or suspended.

DMA specific implementation & user guidelines

Feature	LPDMA	GPDMA
Number of channels	4	16
Master port(s)	1x (32-bit) AHB	2x (32-bit) AHB ❖ Port#0 should be typically allocated for transfers to/from peripherals ➢ There is a direct hardware datapath to APB peripherals, outside the AHB matrix ❖ Port#1 should be typically allocated for transfers to/from memory ❖ In any case, any GPDMA target can be addressed from any port
DMA transfers	Single only	Single and bursts
DMA scheduler	Direct transfers (read followed by write)	FIFO-based bursts (dual issue)
Channel FIFO size	NA	Ch0-11: 8 bytes (2 words) ❖ These channels should be typically allocated for transfers from/to an APB/AHB peripheral and SRAM Ch12-15: 32 bytes (8 words) ❖ These channels may be also used for transfers from/to a data-demanding AHB peripheral and SRAM, or for transfers from/to external memories ❖ 4-word burst should be privileged when applicable for faster performances (faster back-to-back transfers, lower bus utilization)
Channel addressing mode	Linear (fixed or incremented with contiguous data)	Ch0-11: linear Ch12-15: 2D addressing
Maximum request ID	16	123
Maximum trigger ID	31	56

life.augmented

This table describes the features of the LPDMA and GPDMA controllers, highlights their differences and also provides user guidelines.
The number of channels is four for LPDMA, 16 for GPDMA.
The two master ports of the GPDMA should be used as follows:

- Port 0 should be allocated for transfers to and from peripherals, because there is a direct hardware datapath between this port and the APB peripherals, outside the AHB matrix
- Port 1 should be allocated for transfers to and from memory, which are performed through the AHB interconnect.

This a typical usage, the user is free to select the ports used to access the source location and the destination location.
The LPDMA only supports single-data transfer, while the GPDMA supports burst transfers.
These burst transfers are performed through a per channel FIFO for queuing source and destination transfers.
The depth of the FIFO is 8 bytes for channels 0 to 11, and 32 bytes for channels 12 to 15.
Consequently channels 0 to 11 should be allocated for transfers involving peripherals and channels 12 to 15 should be allocated for memory to memory transfers and transfers involving data-demanding AHB peripherals.
In terms of addressing mode, the LPDMA supports linear addresses, either fixed or incremented without stride.
The GPDMA supports more sophisticated addressing modes: linear with possibly non contiguous data on all channels and 2D addressing only on channels 12 to 15.
The number of requests and trigger inputs is not the same for LPDMA and GPDMA. The number of inputs is lower for the LPDMA, because it serves the peripherals belonging to the Smart Run domain.

Thank you

life.augmented

In addition to this presentation, you can refer to the following presentations:

- Power management
- Reset and clock controller
- ULPMark-Peripheral Profile and LPBAM use case consumption
- Autonomous DMA & low power mode
- DMA Circular buffering & double buffering
- DMA Register file
- DMA Error reporting
- DMA Linked list
- DMA Input-output LLI control.