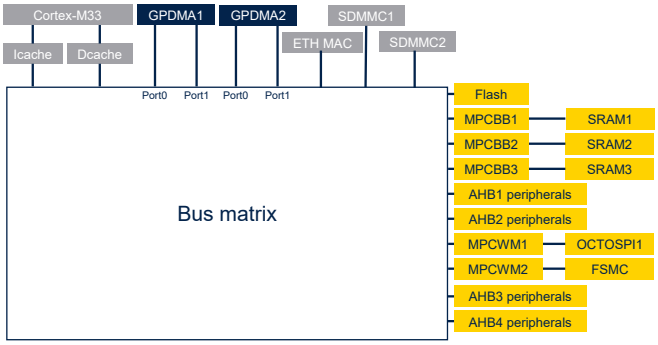
Hello, and welcome to this overview of GPDMA controllers embedded in the STM32H5.

System architecture & DMA overview



Application benefit

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

- Single DMA driver
- Two same hardware instances: GPDMA1 and GPDMA2
  - 8 concurrent channels for each
- 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 STM32H5 embeds two new DMA controller instances in the high-performance segment of the STM32 devices : the General Purpose DMA 1 (or GPDMA1) and the General Purpose DMA 2.
GPDMA1 and GPDMA2 are identical and have exactly the same connections to the system and peripherals of the device.
These DMA controllers are in charge of data movement between memory-mapped locations, memory or peripherals, thus offloading the Cortex-M33 core.
Each of them supports 8 concurrent and independent channels.
GPDMA1 and GPDMA2 are controlled by the same software driver.
The DMA multiplexer, connecting the requests generated by peripherals to the channels is integrated in the GPDMA

instance.
GPDMA supports linked-list base programming to enable buffer chaining.
These DMA controllers are able to temporarily request the clock when a request is received, in order to perform the transfer while the CPU is in Sleep mode.
Also, flexible intra-channel and inter-channel transfer chaining is supported, without software intervention.

GPDMA key features

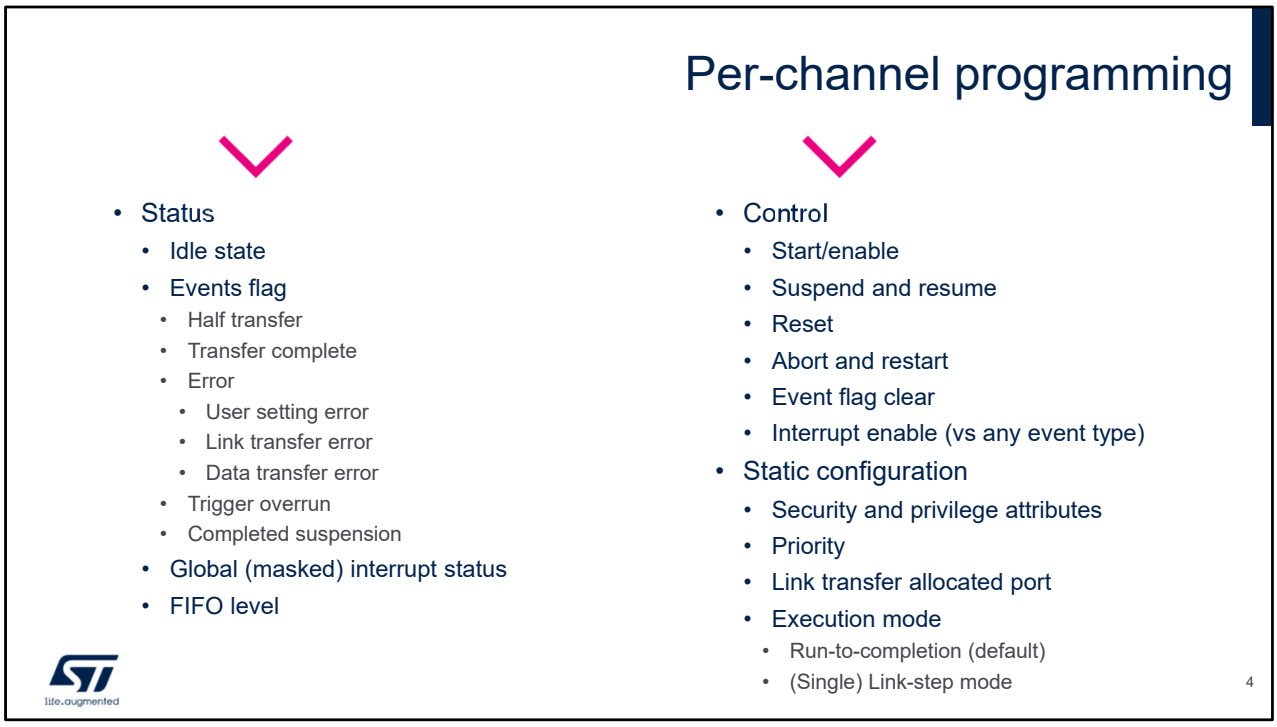
- Two AHB bidirectional master ports
- Memory-mapped data transfers from a source to a destination
  - Peripheral-to-memory
  - Memory-to-peripheral
  - Memory-to-memory
  - Peripheral-to-peripheral
- Autonomous data transfers during Sleep mode
- Concurrent DMA channels
- Transfers arbitration is based 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

Each instance of 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 GPDMA enables autonomous transfers during sleep mode.
Within a same instance, 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.



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.

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
    - The (optional) link transfer: (next) LLIN+1 is loaded by the DMA hardware and updates its (linked-list) register file (aka LLI0)
    - 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.
An LLI n transfer is the sequence of:

1. A data transfer: GPDMA executes the data transfer as described by the GPDMA internal register file (this data transfer can be void/null for LLI0).
2. A conditional link transfer: GPDMA automatically and conditionally updates its internal register file by the data structure of the next LLI n+1.

The programmable number of data bytes to be transferred from the source for a given LLI, defines the block size.

Per-channel programming

- Per-LLI programming
  - Address
    - Source & destination start address
    - Source & destination addressing mode
    - Source & destination address offset between bursts (GPDMA1/2 ch6 & ch7 only)
  - Data
    - Number of data bytes from the source, defining the block level
    - Number of repeated blocks (GPDMA1/2 ch6 & ch7 only)
    - Source & destination data width and data handling
    - Source & destination burst length
  - Source & destination allocated port
  - 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 6 and 7 and burst length for all channels.

The AHB ports used to access the source and destination locations have to be chosen.

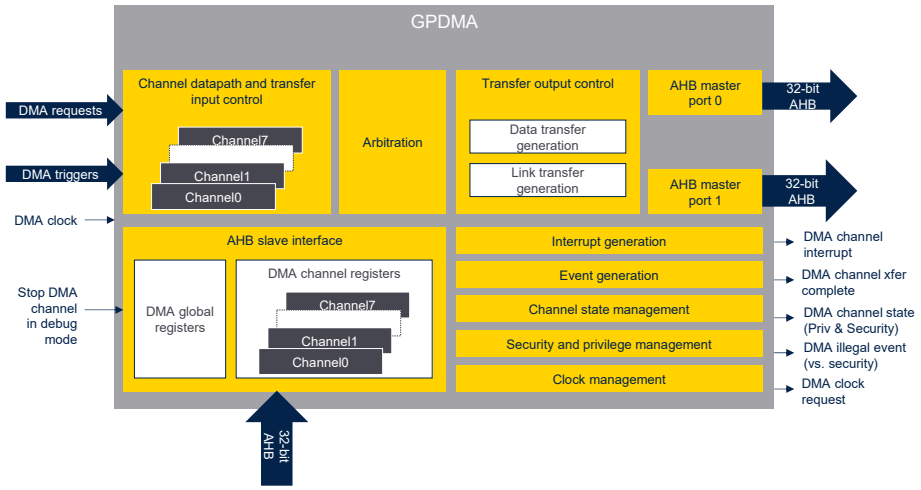
The security attribute is also selectable when the channel itself is programmed as secure.

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.

GPDMA block diagram



The block diagram of GPDMA is depicted.
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 or Sleep.
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	GPDMA (GPDMA1 or GPDMA2)
Number of channels	8
Master port(s)	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 and bursts
DMA scheduler	FIFO-based bursts (dual issue)
Channel FIFO size	Ch0-3: 8 bytes (2 words) ➢ These channels should be typically allocated for transfers from/to an APB/AHB peripheral and SRAM Ch4-7: 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	Ch0-5: linear Ch6-7: 2D addressing
Maximum request ID	135
Maximum trigger ID	43

life.augmented

8

This table describes the features of the GPDMA controller instances. GPDMA1 and GPDMA2 are identical with same features.

Each instance has 8 channels, providing for the DMA driver a total of 16 channel.

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 GPDMA supports burst transfers (and single transfers if the burst length is set to 1).
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 3, and 32 bytes for channels 4 to 7.
Consequently channels 0 to 3 should be allocated for transfers involving peripherals and channels 4 to 7 should be allocated for memory-to-memory transfers and transfers involving data-demanding AHB peripherals.
In terms of addressing mode, the GPDMA channels 0 to 5 support linear addresses, either fixed or incremented without stride.
Each GPDMA instance supports an additional addressing mode on top of the linear mode for the channels 6 and 7: the 2D addressing.
The number of requests and trigger inputs is the same for GPDMA1 and GPDMA2 and they are connected to the same peripheral requests and triggers.

Thank you

life.augmented

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

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