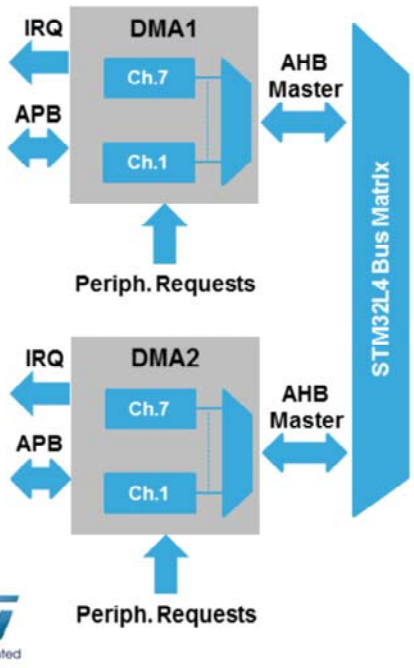
Welcome to the presentation of the STM32L4 direct memory access controller (DMA) . It covers the main features of this module, which is widely used to handle the STM32 peripheral data transfers.

Overview



- STM32L4 DMA features
  - Flexible configuration
  - Hardware and software priority management
  - Configurable data transfer modes
    - Peripheral-to-Memory, Memory-to-Peripheral, Peripheral-to-Peripheral, and Memory-to-Memory modes

Application benefits

- DMA support for timers, ADC, and communication peripherals
- Offload CPU from data transfer management
- Simple integration

The STM32L4 has two Direct Memory Access controllers designed to efficiently support data transfers from peripherals and memories without any loading on the CPU. The DMA controllers are fully configurable and manage hardware and software priorities between channels as well as data transfer modes.

Key features

3

- 14 independent configurable channels over DMA1 & DMA2
  - Hardware request or software trigger on each channel
  - Software programmable priorities with hardware priority in case of equality
- Independent and flexible channel configuration
  - Fully programmable channels (data format, increment type, address)
  - Independent channel interrupt flags (half transfer, transfer complete, transfer error, global flags)
  - Support for circular buffer management.
- Faulty channel is automatically disabled in case of bus access error.

life.augmented

The two DMA controllers (DMA1 and DMA2) have 14 channels in total, each dedicated to managing memory access requests from one or more peripherals. Each channel has flexible hardware requests or software trigger. The channel software priority is programmable and a hardware priority is used in case of equality. Channels are independently configurable. Each channel has its own data format, increment type and data address for both source and destination. Independent channel interrupt flags allow to trigger half transfer, transfer complete, transfer error events. A global flag is also available to facilitate the software efficiency. In case of an error, the faulty channel is automatically disabled without any impact on the other active DMA channels.

Individual channel flexibility

4

- Programmable features
  - Independent source and destination data size (8-bit/16-bit/32-bit)
  - Independent source and destination address
  - Independent source and destination pointer address increment
  - Programmable number of data to be transferred up to 65,535 requests
- Circular mode
  - Handle circular buffers with continuous data flow
  - Source and Destination addresses are automatically reloaded
  - Data transfer size is automatically reloaded

life.augmented

For each channel, the source and destination data size format is independently configurable for 8-, 16- or 32-bit packets. The source and destination address and pointer increment is also independently configurable. The transfer data size can be pre-programmed up to 65535. Circular buffer mode is available to support a continuous flow of data. The source and the destination addresses and the number of data to be transferred are automatically reloaded after the complete transfer.

Channel transfer management

- Memory-to-Memory mode
  - Transfer starts as soon as the channel is enabled (No hardware request)
- Peripheral-to-Memory, Memory-to-Peripheral, Peripheral-to-Peripheral
  - A transfer occurs on each hardware request
  - Once the transfer is completed, the request is acknowledged

Memory-to-memory mode allows transfers from one address location to another without a hardware request. Once the channel is configured and enabled, the transfer starts immediately. When data is transferred from or to a peripheral, the hardware request coming from the selected peripheral is used to trigger the data transfer. Once the transfer is completed, the request is acknowledged.

STM32L4 instances

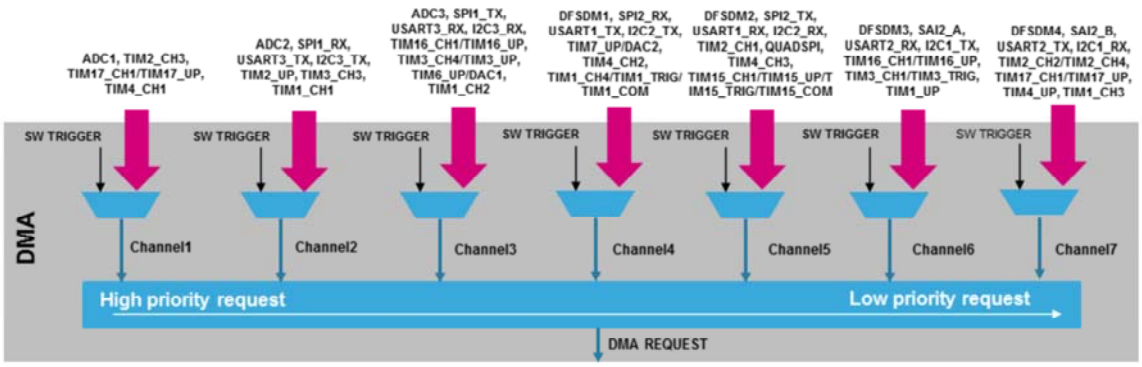
DMA features	DMA1	DMA2
Number of channels	7	7

The STM32L4 implements two instances of the DMA with 7 channels each

DMA1 requests mapping

7

- DMA1 controller provides access to 7 channels
  - New: Peripheral requests are mapped through a multiplexer (Not OR gate)
  - Independent software trigger for each channel

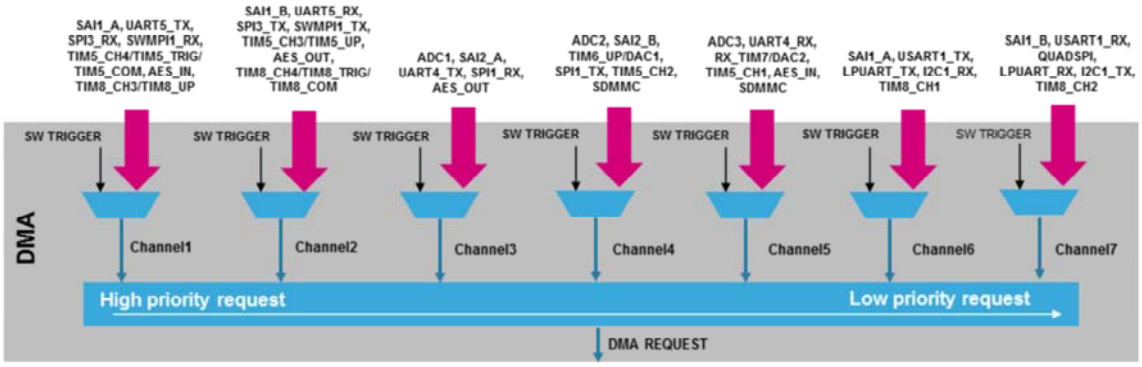


The current figures reflects STM32L47x/48x devices. Other devices have compatible request mapping (either not applicable request when peripheral not available or new request for new peripheral).

DMA1 peripheral requests are mapped through a multiplexer for each corresponding channel. Each channel has a fixed group of peripheral requests, and the multiplexer allows only one peripheral request for each channel.

DMA2 requests mapping

- DMA2 controller provides access to 7 channels
  - New: Peripheral requests are mapped through a multiplexer (Not OR gate)
  - Independent Software trigger for each channel



The current figures reflects STM32L47x/48x devices. Other devices have compatible request mapping (either not applicable request when peripheral not available or new request for new peripheral).

The DMA2 controller also provides access to 7 channels with independent multiplexers for each channel. Some of the peripheral requests are similar to the DMA1, but there are unique peripherals requests as well (for example, TIM5, TIM8, LPUART, SWMPI1, UART4, UART5, and AES).

Interrupts

- Interrupt events for each channel

Interrupt event	Description
Half transfer	Set when half of the data transfer size has completed
Transfer complete	Set when the full data transfer size has completed
Transfer error	Set when an error occurs during the data transfer
Global interrupt	Set whenever a half transfer, a transfer complete or a full transfer event occurs

life.augmented

Each DMA channel is designed with this group of Interrupt events. The Half Transfer interrupt flag is set when the half the data has been transferred; the Transfer Complete flag is set when the transfer is complete; the Transfer Error flag is set when an error occurs during the data transfer; the Global Interrupt flag is set whenever a half transfer, a transfer complete or a full transfer event occurs.

DMA in low-power modes

Mode	Description
Run	Active.
Sleep	Active. DMA interrupts can wake the STM32L4.
Low-power run	Active.
Low-power sleep	Active. DMA interrupts can wake the STM32L4.
Stop 0/Stop 1	Frozen. DMA registers content is retained.
Stop 2	Frozen. DMA registers content is retained.
Standby	Powered-down. DMA must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. DMA must be reinitialized after exiting Shutdown mode.

life.augmented

The DMA is active in Run, Sleep, Low-power run and Low-power sleep mode. DMA interrupts will wake the STM32L4 from Sleep and Low-power sleep modes. In Stop mode, the DMA is stopped and the contents of the DMA registers are retained. The DMA is powered-down in Standby and Shutdown modes, and the DMA registers must be reinitialized after exiting Standby or Shutdown mode.