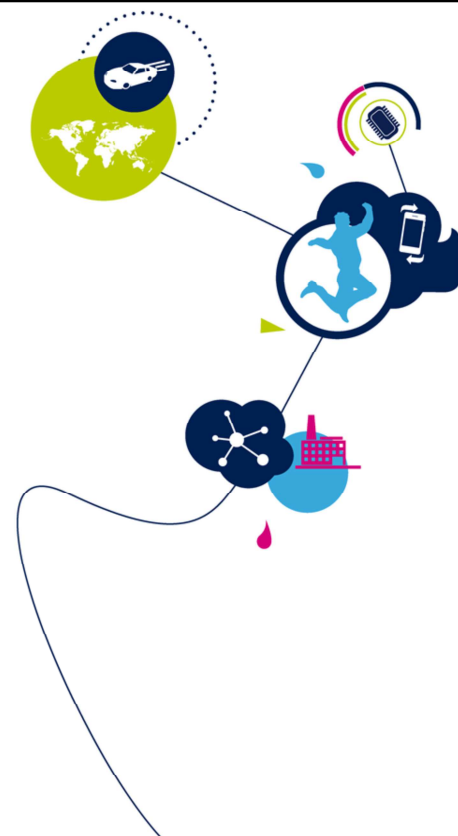
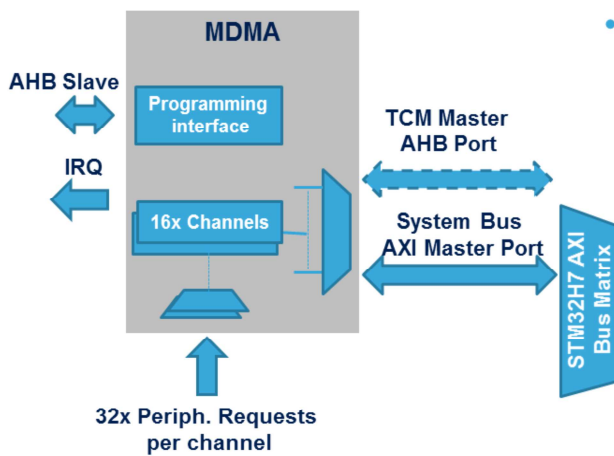


STM32H7 - MDMA

Master direct memory access controller (MDMA)
Revision 1.0



Welcome to the presentation of the STM32H7 master direct memory access controller (MDMA). It covers the main features of this module, which is widely used to handle data transfers.



- STM32H7 MDMA features

- Dual master bus
 - 64-bit AXI and 32-bit AHB master port
- Flexible and independent configuration
- Hardware and software priority management
- Configurable data transfer modes
 - Peripheral-to-Memory, Memory-to-Peripheral, and Memory-to-Memory modes

Application benefits

- DMA support for block transfer and linked list
- Offload CPU from data transfer management
- Simple integration

The master direct memory access (MDMA) is optimized for data transfers between memories since it supports linked list transfers that allow performing a chained list of transfers without the need for CPU intervention. This keeps the CPU resources free for other operations.

The MDMA controller provides a master AXI interface for main memory and peripheral registers access (system access port) and a master AHB interface only for Cortex-M7 TCM memory access (TCM access port).

- Each MDMA channel can perform:
 - **Single block transfer:** one block is transferred (up to 64 Kbytes). At the end of the block, the DMA channel is disabled.
 - **Repeated block transfer:** a number of blocks (up to 4096 blocks) is transferred before disabling the channel.
 - **Linked list transfer:** when the transfer of the current data is completed, a new block control structure is loaded from memory and a new block transfer is started.
- Individual channel flexibility:
 - Independent Incrementing, decrementing or non-incrementing for source and destination
 - Independent transfer size, and increment size for source and destination
 - Endianness eXchange: Byte, Half-word and Word



Each of the DMA controller channels provides a unidirectional transfer link between a source and a destination.

Each channel can perform:

Single block transfer: one block is transferred. At the end of the block, the DMA channel is disabled and an End of Channel Transfer interrupt is generated.

Repeated block transfer: a number of blocks is transferred before disabling the channel.

Linked list transfer: when the transfer of the current data block (or last block in a repeat) is completed, a new block control structure is loaded from memory and a new block transfer is started.

The MDMA also features incrementing, decrementing or non-incrementing (fixed) addressing for source and destination. The size and address increment for both source and destination can be independently selected.

- MDMA is useful to collect data from other DMAs (DMA1/DMA2 and BDMA) and make data available to the CPU in the D1 domain (in DTCM or AXI-SRAM).
- DMA linked lists are used to perform a set of DMA transfers without the need for CPU intervention:
 - Can be used to prepare data for other DMAs and then set the DMA configuration to start transfers.
 - It is used to support scatter/gathering. This means that the source and destination areas do not need to occupy contiguous areas in memory. The source and destination data areas are defined by a series of linked list descriptors that control the transfer of data blocks.
 - It can be used to perform graphics operations like converting JPEG images and drawing them using DMA2D



MDMA is useful to collect data from other DMAs (DMA1/DMA2 and BDMA) and make data available to the CPU in the D1 domain (in DTCM or AXI-SRAM).
DMA linked lists are used to perform a set of DMA transfers without the need for CPU intervention:
Can be used to prepare data for other DMAs and then set the DMA configuration to start transfers.
It is used to support scatter/gathering. This means that the source and destination areas do not need to occupy contiguous areas in memory. The source and destination data areas are defined by a series of linked lists descriptors that control the transfer of data blocks.
It can be used to pass data to a JPEG decoder and DMA2D to accelerate image drawing.

- MDMA supports single or incremental burst transfers
- Software-configurable burst sizes up to 128 bytes
- Maximum burst data size is 128 bytes. For larger data sizes the burst length is limited by the 128-level FIFO that will be used to store temporary the data to be transferred
 - e.g. 16x64-bit or 32x32-bit
- For the TCM memory accesses, the burst access is only allowed when the increment and data size are identical and lower than or equal to 32 bits.



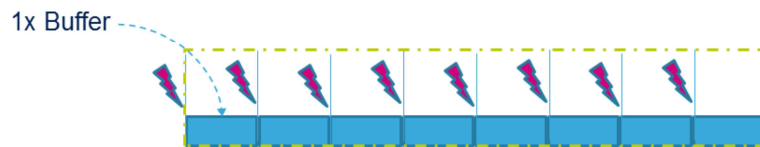
The MDMA supports incremental burst transfers. The size of the burst is software-configurable, up to 128 bytes. For larger data sizes the burst length is limited, as to respect the maximum data burst size of 128 bytes (e.g. 16x64-bit or 32x32-bit).

For the TCM memory accesses, the burst access is only allowed when the increment and data size are identical and lower than or equal to 32 bits.

MDMA channel triggering model

6

- The size of the data array to be transferred for a single request will be one of the following:
 1. The buffer transfer size
 - TRGM = '00'



One buffer is transferred for each request

The size is done selected using the TRGM[1:0] (Trigger Mode) selection field; The size of the data array to be transferred for a single request will be the buffer transfer size when the TRGM (Trigger mode) = '00'.

MDMA channel triggering model

7

- The size of the data array to be transferred for a single request will be one of the following:
 1. The buffer transfer size
 - TRGM = '00'
 2. The block size
 - TRGM = '01'



One block is transferred for each request

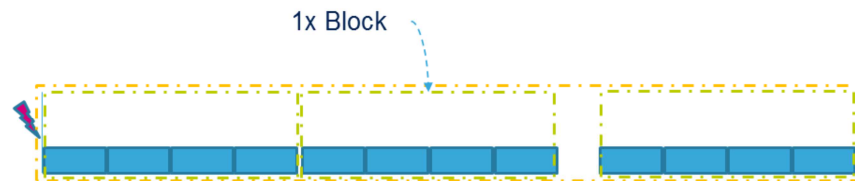


The size of the data array to be transferred for a single request will be the block size when the TRGM (Trigger Mode) = '01'.

MDMA channel triggering model

8

- The size of the data array to be transferred for a single request will be one of the following:
 1. The buffer transfer size
 - TRGM = '00'
 2. The block size
 - TRGM = '01'
 3. Repeated block
 - TRGM = '10'



Several blocks are transferred for each request

The size of the data array to be transferred for a single request will be a repeated block when TRGM = '10'.

MDMA channel triggering model

9

- The size of the data array to be transferred for a single request will be one of the following:
 1. The buffer transfer size
 - TRGM = '00'
 2. The block size
 - TRGM = '01'
 3. Repeated block
 - TRGM = '10'
 4. Complete channel data
 - TRGM = '11'



A request starts the transfer until the linked list pointer for the channel is null



The size of the data array to be transferred for a single request will be a complete channel data (until the linked list pointer for the channel is null) when TRGM = '11'.

Request Arbitration (1/2)

10

- Each channel has programmable SW priority (4 priority levels)
 - If two channels have same SW priority, the lower channel number has priority
- A buffer transfer is the minimum data size to be transferred by the MDMA without performing a new arbitration between MDMA channel requests
- In case of a block transfer, after transferring an individual buffer, the MDMA will enter in a new arbitration phase between new external requests and internally memorized ones.
 - If no other channel request with a higher priority is active, a new buffer transfer will be started for the same channel
 - Higher level MDMA requests will not be blocked for the more than a buffer transfer period as the channel arbitration is performed after each buffer transfer



Each channel has programmable priority. If two channels have same programmable priority, the lower channel number has higher priority.

An arbiter manages the MDMA channel requests based on their priority. When MDMA is idle and after the end of each buffer transfer, all MDMA requests (hardware or software) are checked for all enabled channels.

A buffer transfer is the minimum data size to be transferred by the MDMA without starting a new arbitration between MDMA channel requests.

In case of a block transfer, after transferring an individual buffer, the MDMA will enter in a new arbitration phase between new external requests and internally memorized ones.

If no other channel request with a higher priority is active, a new buffer transfer will be started for the same channel.

Higher level MDMA requests will not be blocked for more than a buffer transfer period as the channel arbitration is

performed after each buffer transfer.

Request Arbitration (2/2)

11

- Two data array size parameters have an impact on MDMA and application responsiveness
 1. Buffer transfer size: data transfer lengths which are uninterruptible at the MDMA level from other channels' requests.
 - It is the length of the data to be transferred, on a channel, before checking for MDMA requests on other channels.
 2. Burst size: it defines the maximum data transfer length which may be uninterruptible at the bus arbitration level
 - It is the length of the data transfer which may be transferred in burst mode.
 - It may block other masters from gaining access to the bus
- It is important to select the correct burst and buffer transfer size considering the “real-time” requirements for other MDMA channels and masters.



Two data array size **parameters** have **an** impact on MDMA and application responsiveness:

1. Buffer transfer size: data transfer lengths which **are** uninterruptible at **the** MDMA level from other channels' requests.
 - It is the length of the data to be transferred, on a channel, before checking for MDMA requests on other channels.
2. Burst size: it defines the maximum data transfer length which may be uninterruptible at **the** bus arbitration level.
 - It is the length of the data transfer which may be **transferred** in burst mode.
 - It may block other masters from gaining access to the bus.

It is important to select **the** correct burst and buffer transfer size **considering** the “real-time” requirements for other

MDMA channels and masters.

- A buffer transfer is the minimum logical amount of data which is transferred on an MDMA request event, on one channel.
 - The total amount of data to be transferred, on the current channel, following a MDMA request, is determined by the triggering model.
- The number of data items to be transferred, their width (8-bit, 16-bit, 32-bit or 64-bit) and the length of the burst used for data transfers, are software programmable.
- After servicing a DMA/Peripheral request event, the request is acknowledged by writing the mask data value to the address given in the mask address.
 - Mainly used to clear the request event source flag.



A buffer transfer -is the minimum logical amount of data (up to 128 bytes) which is transferred on an MDMA request event, on one channel.

An MDMA buffer transfer consists of a sequence of a given number of data transfers (done as single or burst data transfers). The number of data items to be transferred and their width (8-bit, 16-bit, 32-bit or 64-bit) are software programmable. The length of the burst used for data transfers is also programmable, independently. After an event requiring a data array to be transferred, the DMA/peripheral sends a request signal to the MDMA controller. The MDMA controller serves the request depending on the channel priorities. The request is acknowledged by writing the mask data value to the address given in the mask address, when these registers are set.

MDMA Block transfer (1/2)

13

- A block is a contiguous array of data, up to 64 Kbytes, which is transferred by successive buffer transfers.
- Each block of data is defined by a “start address” and “block length”.
- Depending on the triggering model configured, a whole block transfer can be triggered by one peripheral/DMA request.
 - A new buffer transfer will be started for the same channel if no other channel request with a higher priority is active



life.augmented

A block is a “contiguous” array of data, up to 64 Kbytes, which is transferred by successive buffer transfers. Each block of data is defined by the start address and the block length.

- When a block transfer is completed, three actions may be executed:
 - If the block is part of a repeated block transfer: the block length is reloaded and a new block start address is computed based on the information in the Block Repeat address Update register.
 - If it is a single block or the last block in a repeated block transfer: the next block information is loaded from the memory.
 - If it is the last block which needs to be transferred for the current MDMA channel : the channel is disabled and no further MDMA requests will be accepted for this channel.



When a block transfer is completed, one of the following three actions can be executed:

- If the block is part of a repeated block transfer: the block length is reloaded and a new block start address is computed (based on the information in the CxBRUR register)
- If it is a single block or the last block in a repeated block transfer: the next block information is loaded from the memory (using the linked list address information, from the MDMA_CxLAR)
- If it is the last block which needs to be transferred for the current MDMA channel (MDMA_CxLAR = 0): the channel is disabled and no further MDMA requests will be accepted for this channel.

MDMA Block repeat mode

15

- The block repeat mode allows to repeat a block transfer, with different “start addresses” for source and destination.
- When the repeat block mode is active (repeat counter is not equal to ‘0’), at the end of the current block transfer, the block parameters will be updated:
 - Reload block number of data bytes to transfer (BNDT) value
 - Update block Source/Destination address values according to block repeat Source/Destination address update mode (BRSUM/BRDUM) configuration
 - Decrement by 1 the repeat counter
- When the repeat block counter reaches 0, this last block will be treated as a single block transfer.



The block repeat mode allows repetition of a block transfer, with different start addresses for source and destination. When the repeat block mode is active (repeat counter is not equal to ‘0’), at the end of the current block transfer, the block parameters will be updated (the BNDDT value reloaded and SAR/DAR values updated according to BRSUM/BRDUM configuration), and the repeat counter is decremented by 1. When the repeat block counter reaches 0, this last block will be treated as a single block transfer.

- The Linked list mode allows to load a new MDMA configuration from the address given in the Channel Link Address register (CLAR)
- Following this operation, the channel is ready to accept new requests, as defined in the block/repeated block modes above, or continue the transfer if the triggering model is set to Complete channel data (TRGM = '11'),
- The trigger source may be automatically changed, when loading the trigger and bus selection register (TBR) value.
- If triggering model is set to TRGM = '11', the triggering model and SW request selected must not be changed.



The Linked list mode allows loading of a new MDMA configuration (CxTCR, CxBNDTR, CxSAR, CxDAR, CxBRUR, CxLAR, CxTBR, CxMAR and CxMDR registers) from the address given in the CxLAR register. This address must address a memory mapped on the AXI system bus. Following this operation, the channel is ready to accept new requests, as defined in the block/repeated block modes above, or continue the transfer if TRGM[1:0] equals '11'. The trigger source can be automatically changed when loading the CxTBR value. The TRGM and SWRM values must not be changed when TRGM[1:0] equals '11'.

MDMA Linked list mode

- The channel configuration (channel link address LAR) must be in the AXI address space.
- LAR value must be aligned at a Double Word address, i.e. $LAR[2:0] = 0x0$

Register (32-bit word)	Offset from Link Address register	Description
CTCR	0x00	Transfer Configuration register
CBNDTR	0x04	Block number of data register
CSAR	0x08	Source address register
CDAR	0x0C	Destination address register
CBRUR	0x10	Block Repeat address Update register
CLAR	0x14	Link Address register: Next descriptor
CTBR	0x18	Trigger and Bus selection Register
CMAR	0x20	Mask address register
CMDR	0x24	Mask Data register



The channel configuration (channel link address LAR) must be in the AXI address space.

LAR value must be aligned at a Double Word address, i.e. $LAR[2:0] = 0x0$.

MDMA request mapping

MDMA request	Request Source	description
mdma_str0 – 7	dma1_tcf0 – 7	DMA1 stream 0 transfer complete flag
mdma_str8 – 15	dma2_tcf0 – 7	DMA2 stream 0 transfer complete flag
mdma_str16	LTDC	LTDC line interrupt
mdma_str17	JPEG	JPEG input FIFO threshold
mdma_str18		JPEG input FIFO not full
mdma_str19		JPEG output FIFO threshold
mdma_str20		JPEG output FIFO not empty
mdma_str21		JPEG end of conversion
mdma_str22	QUADSPI	QUADSPI FIFO threshold
mdma_str23		QUADSPI transfer complete
mdma_str24	DMA2D	DMA2D CLUT transfer complete
mdma_str25		DMA2D transfer complete
mdma_str26		DMA2D transfer watermark
mdma_str29	SDMMC1	SDMMC1 End of data



This table describes the MDMA requests and their mapping to devices. An MDMA channel can be triggered by the end of transfer of the DMA1 stream. In response to this trigger, the MDMA can:

- perform data transfer from SRAM D2 to D1 RAM
- Or reprogram the DMA1 zero flow for a new transfer

To enable the MDMA to efficiently offload the CPU, MDMA channels can be triggered by devices interrupt to automate data exchanges and processing. Examples of peripherals triggers are described in this table.

- Interrupt events for each channel

Interrupt event	Description
CTCIF	MDMA channel transfer complete. It is also set as a result of writing channel enable bit to 0.
BTIF	MDMA block transfer complete
BTRIF	MDMA block repeat transfer complete
TCIF	MDMA buffer transfer complete. It may be used as a debug feature (without interrupt), indicating that (at least) a MDMA buffer transfer had been generated since the last flag reset
TEIF	MDMA transfer error



For each MDMA channel, an interrupt can be produced on the following events:

- Channel transfer completed
- Block transfer completed
- Block transfer repeat completed
- buffer transfer completed
- Transfer error

MDMA in low-power modes

Mode	Description
Run	Active.
Sleep	Active. MDMA interrupts can wake the STM32H7.
Stop	Frozen. MDMA registers content is retained (D1 domain in Stop).
Standby	Powered-down. MDMA must be reinitialized after exiting Standby mode.

The MDMA is active in Run and Sleep modes. DMA interrupts will wake the STM32H7 from Sleep mode. In Stop mode, the DMA is stopped and the contents of the DMA registers are retained. The DMA is powered down in Standby mode, and the DMA registers must be reinitialized after exiting Standby mode.