Chapter 8 Direct Memory Access Control (DMA)

Direct Memory Access Controller (DMA) provides a high-speed data transfer method between peripherals and memory or between memory and memory without CPU intervention, and data can be moved quickly through DMA to save CPU resources for other operations.
Each channel of the DMA controller is dedicated to managing requests for memory access from one or more peripherals. There is also an arbiter to coordinate the priority between the channels.

8.1 Main Features

- Multiple independently configurable channels
- Each channel is directly connected to a dedicated hardware DMA request and supports software triggering
- Buffer management with loop support
- Request priority between multiple channels can be set by software programming (very high, high, medium and low) and priority setting is determined by the channel number when equal (the lower the channel number the higher the priority)
- Supports peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers
- Flash memory, SRAM, peripheral SRAM and HB peripherals can be used as access sources and targets
- Programmable number of data transfer bytes: up to 65535

8.2 Function Description

8.2.1 DMA Channel Processing

1) Arbitration priority

DMA requests generated by multiple independent channels are fed to the DMA controller via a logical or structure, and only one channel request is currently responded to. An arbiter inside the module selects the peripheral/memory access to be initiated based on the priority of the channel request.
In software management, the application can configure the priority level for each channel independently by setting the PL[1:0] bits of the DMA_CFGRx register, including four levels: highest, high, medium and low. When the software setting levels are the same between channels, the module will be selected according to a fixed hardware priority, with the lower channel number having a higher priority than the higher one.

2) DMA configuration

When the DMA controller receives a request signal, it accesses the requested peripheral or memory and establishes a data transfer between the peripheral or memory and the memory. It consists of the following 3 main operation steps.

(1) Fetch data from the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register. The start address for the first transfer is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx registers.

(2) Store data to the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register, and the start address for the first transfer is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx registers.

(3) Performs a decrement operation of the value in the DMA_CNTRx register, which indicates the number of operations currently outstanding for transfer.

Each channel includes 3 types of DMA data transfer methods.

- Peripheral to memory (MEM2MEM=0, DIR=0)
- Memory to peripheral (MEM2MEM=0, DIR=1)
- Memory to memory (MEM2MEM=1)

Note: The memory-to-memory mode does not require a peripheral request signal. After configuring this mode (MEM2MEM=1), the channel is turned on (EN=1) to start data transfer. This mode does not support cyclic *mode.*

The configuration process is as follows.

1) Set the first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register. This address will be the source or destination address for data transfer when a DMA request occurs.
2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transferred data will be read from or written to this address.
3) Set the amount of data to be transferred in the DMA_CNTRx register. This value is decremented after each data transfer.
4) Set the priority of the channel in the PL[1:0] bits of the DMA_CFGRx register.
5) Set the direction of data transfer, cyclic mode, incremental mode for peripheral and memory, data width for peripheral and memory, transfer halfway, transfer complete, and transfer error interrupt enable bits in the DMA_CFGRx register.
6) Set the ENABLE bit of the DMA_CCRx register to start channel x.

Note: The DMA_PADDRx/DMA_MADDRx/DMA_CNTRx registers and the direction of data transfer (DIR), cyclic mode (location), and incremental mode of peripherals and memory (MINC/PINC) control bits in the DMA_CFGRx register can be configured to write only when the DMA channel is turned off.

3) Circular mode

Setting CIRC position 1 of the DMA_CFGRx register enables the cyclic mode function for channel data transfers. In cyclic mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to its initial value, and the internal peripheral and memory address registers are reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers, and DMA operation will continue until the channel is turned off or the DMA mode is turned off.

4) DMA processing status

- Transfer half: It corresponds to the hardware setting of HTIFx bit in DMA_INTFR register. The DMA transfer bytes half flag will be generated when the number of DMA transfers is reduced to less than half of the initial set value, and an interrupt will be generated if HTIE is set in the DMA_CCRx register. The hardware uses this flag to alert the application that it can prepare for a new round of data transfers.
- Transfer completion: corresponds to the hardware setting of the TCIFx bit in the DMA_INTFR register. When the number of DMA transfer bytes decreases to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA_CCRx register, an interrupt will be generated.
- Transfer error: corresponds to a hardware set of the TEIFx bit in the DMA_INTFR register. Reading and writing a reserved address area will generate a DMA transfer error. At the same time the module hardware will automatically clear the EN bit of the DMA_CCRx register corresponding to the channel where the error occurred, and the channel is turned off. If TEIE is set in the DMA_CCRx register, an interrupt will be generated.

When the application queries the DMA channel status, it can first access the GIFx bit of the DMA_INTFR register to determine which channel is currently experiencing a DMA event, and then process the specific DMA event content for that channel.

8.2.2 Programmable Total Data Transfer Size/Data Bit Width/Alignment

The total size of the data to be transferred per DMA channel round is programmable up to 65535 times, and the number of pending transfer bytes is indicated in the DMA_CNTRx register. At EN=0, the set value is written, and at EN=1 when the DMA transfer channel is turned on, this register becomes a read-only attribute with a decreasing value after each transfer.

The transferred data fetch values of peripherals and memories support the address pointer auto-increment function with programmable pointer increments. The first transmitted data address they access is stored in the

DMA_PADDRx and DMA_MADDRx registers.By setting the PINC bit or MINC position 1 of the DMA_CFGRx register, the peripheral address self-increment mode or memory address self-increment mode can be enabled, respectively. PSIZE[1:0] sets the peripheral address fetch data size and address selfincrement size. MSIZE[1:0] sets the memory address to take the data size and address self-increasing small, including three choices: 8-bit, 16-bit, 32-bit. The specific data transfer methods are listed in the following table.

Table 8-1 DMA transfer with different data bit widths (PINC=MINC=1)

Source bit width	Objectives bit width	Transmission number	Source: address/data	Target: address/data	Transfer operations
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● The source address increment is aligned with the data bit width set at the source and takes a value equal to the data bit width at the source ● The target address increment is aligned with the bit width of the target setup data and takes a value equal to the target data bit width ● DMA transfer of data sent to the target based on the principle: the high bit of the data size is not enough to make up 0, the high bit of the data size overflow is removed ● Storage data mode: small-end mode, low address stores low bytes, high address stores high bytes
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/0000B1B0 0x04/0000B3B2 0x08/0000B5B4 0x0C/0000B7B6	
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

8.2.3 DMA Request Mapping

The DMA controller provides seven channels, each corresponding to multiple peripheral requests. By setting the corresponding DMA control bits in the corresponding peripheral registers, the DMA function of each peripheral can be turned on or off independently, and the specific correspondence is as follows.

Figure 8-1 DMA1 request image

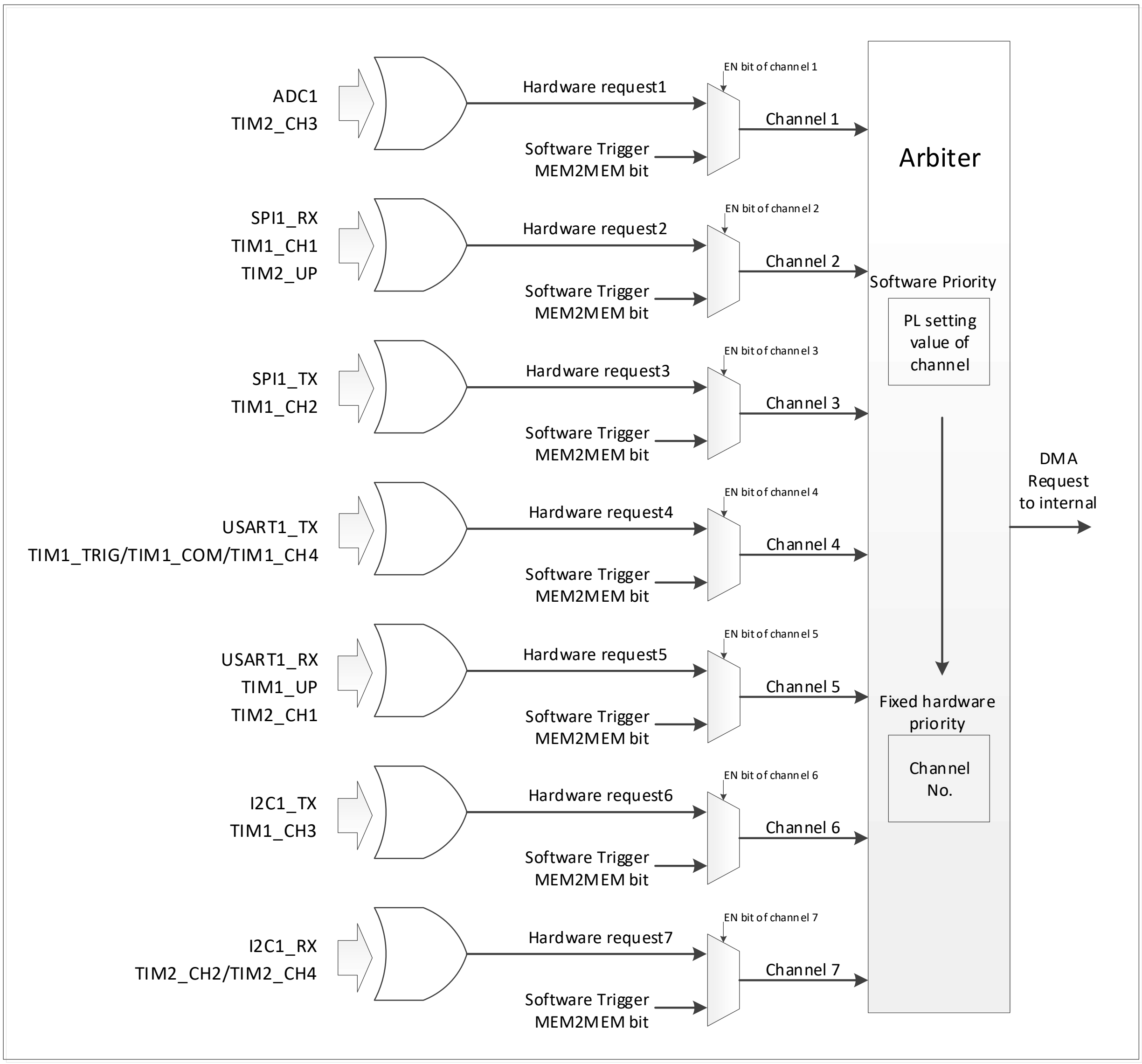

Table 8-2 DMA1 peripheral mapping table for each channel

Peripherals	Channel1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1						
SPI1		SPI1_RX	SPI1_TX				
USART1				USART1_TX	USART1_RX		
I2C1						I2C1_TX	I2C1_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4

8.3 Register Description

Table 8-3 DMA-related registers list

Name	Access address	Description	Reset value
R32_DMA_INTFR	0x40020000	DMA interrupt status register	0x00000000
R32_DMA_INTFCR	0x40020004	DMA interrupt flag clear register	0x00000000
R32_DMA_CFGR1	0x40020008	DMA channel 1 configuration register	0x00000000
R32_DMA_CNTR1	0x4002000C	DMA channel 1 number of data register	0x00000000

R32_DMA_PADDR1	0x40020010	DMA channel 1 peripheral address register	0x00000000
R32_DMA_MADDR1	0x40020014	DMA channel 1 memory address register	0x00000000
R32_DMA_CFGR2	0x4002001C	DMA channel 2 configuration register	0x00000000
R32_DMA_CNTR2	0x40020020	DMA channel 2 number of data register	0x00000000
R32_DMA_PADDR2	0x40020024	DMA channel 2 peripheral address register	0x00000000
R32_DMA_MADDR2	0x40020028	DMA channel 2 memory address register	0x00000000
R32_DMA_CFGR3	0x40020030	DMA channel 3 configuration register	0x00000000
R32_DMA_CNTR3	0x40020034	DMA channel 3 number of data register	0x00000000
R32_DMA_PADDR3	0x40020038	DMA channel 3 peripheral address register	0x00000000
R32_DMA_MADDR3	0x4002003C	DMA channel 3 memory address register	0x00000000
R32_DMA_CFGR4	0x40020044	DMA channel 4 configuration register	0x00000000
R32_DMA_CNTR4	0x40020048	DMA channel 4 number of data register	0x00000000
R32_DMA_PADDR4	0x4002004C	DMA channel 4 peripheral address register	0x00000000
R32_DMA_MADDR4	0x40020050	DMA channel 4 memory address register	0x00000000
R32_DMA_CFGR5	0x40020058	DMA channel 5 configuration register	0x00000000
R32_DMA_CNTR5	0x4002005C	DMA channel 5 number of data register	0x00000000
R32_DMA_PADDR5	0x40020060	DMA channel 5 peripheral address register	0x00000000
R32_DMA_MADDR5	0x40020064	DMA channel 5 memory address register	0x00000000
R32_DMA_CFGR6	0x4002006C	DMA channel 6 configuration register	0x00000000
R32_DMA_CNTR6	0x40020070	DMA channel 6 number of data register	0x00000000
R32_DMA_PADDR6	0x40020074	DMA channel 6 peripheral address register	0x00000000
R32_DMA_MADDR6	0x40020078	DMA channel 6 memory address register	0x00000000
R32_DMA_CFGR7	0x40020080	DMA channel 7 configuration register	0x00000000
R32_DMA_CNTR7	0x40020084	DMA channel 7 number of data register	0x00000000
R32_DMA_PADDR7	0x40020088	DMA channel 7 peripheral address register	0x00000000
R32_DMA_MADDR7	0x4002008C	DMA channel 7 memory address register	0x00000000

8.3.1 DMA Interrupt Status Register (DMA_INTFR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
27/23/19/15/11/7/3	TEIFx	RO	Transmission error flag for channel x (x=1/2/3/4/5/6/7). 1: A transmission error occurred on channel x. 0: No transmission error on channel x. Hardware set, software write CTEIFx bit to clear this flag.	0
26/22/18/14/10/6/2	HTIFx	RO	Transmission halfway flag for channel x (x=1/2/3/4/5/6/7). 1: a transmission over half event is generated on channel x. 0: No transmission over half on channel x. Hardware set, software write CHTIFx bit to clear this flag.	0
25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flag for channel x (x=1/2/3/4/5/6/7). 1: a transmission completion event is generated on channel x. 0: No transmission completion event on channel x. Hardware set, software write CTCIFx bit to clear this flag.	0
24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag for channel x (x=1/2/3/4/5/6/7). 1: TEIFx or HTIFx or TCIFx is generated on channel x. 0: No TEIFx or HTIFx or TCIFx occurred on channel x. Hardware set, software write CGIFx bit to clear this flag.	0

8.3.2 DMA Interrupt Flag Clear Register (DMA_INTFCR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
27/23/19/15/11/7/3	CTEIFx	WO	Clear the transmission error flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx flag in the DMA_INTFR register. 0: No effect.	0
26/22/18/14/10/6/2	CHTIFx	WO	Clear the transmission halfway flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the HTIFx flag in the DMA_INTFR register. 0: No effect.	0
25/21/17/13/9/5/1	CTCIFx	WO	Clear the transmission completion flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TCIFx flag in the DMA_INTFR register. 0: No effect.	0
24/20/16/12/8/4/0	CGIFx	WO	Clear the global interrupt flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx/HTIFx/TCIFx/ GIFx flags in the DMA_INTFR register. 0: No effect.	0

8.3.3 DMA Channel x Configuration Register (DMA_CFGRx) (x=1/2/3/4/5/6/7)

Offset address: 0x08 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	MEM2MEM	RW	Memory-to-memory mode enable. 1: Enable memory-to-memory data transfer mode. 0: Disable memory-to-memory data transfer mode.	0
[13:12]	PL	RW	Channel priority setting. 00: low; 01: medium. 10: High; 11:Very high.	0
[11:10]	MSIZE	RW	Memory address data width setting. 00: 8 bits; 01: 16 bits. 10: 32 bits; 11: Reserved.	0
[9:8]	PSIZE	RW	Peripheral address data width setting. 00: 8 bits; 01: 16 bits.	0

			10: 32 bits; 11: Reserved.	
7	MINC	RW	Memory address incremental incremental mode enable. 1: Enable incremental memory address increment operation. 0: Memory address remains unchanged operation.	0
6	PINC	RW	Peripheral address incremental incremental mode enable. 1: Enable incremental incremental operation of the peripheral address. 0: Peripheral address remains unchanged operation.	0
5	CIRC	RW	DMA channel cyclic mode enable. 1: Enables cyclic operation. 0: Perform a single operation.	0
4	DIR	RW	Data transfer direction. 1: Read from memory. 0: Read from peripheral.	0
3	TEIE	RW	Transmission error interrupt enable control. 1: Enable transmission error interrupt. 0: Disable transmission error interrupt.	0
2	HTIE	RW	Transmission over half interrupt enable control. 1: Enable the transmission over half interrupt. 0: Disable the transmission over half interrupt.	0
1	TCIE	RW	Transmission completion interrupt enable control. 1: Enable the transmission completion interrupt. 0: Disable the transmission completion interrupt.	0
0	EN	RW	Channel enable control. 1: Channel on; 0: Channel off. When a DMA transfer error occurs, the hardware automatically clears this bit to 0 and shuts down the channel.	0

8.3.4 DMA Channel x Number of Data Register (DMA_CNTRx) (x=1/2/3/4/5/6/7)

Offset address: 0x0C + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	NDT[15:0]	RW	Number of data transfers, range 0-65535. This register can only be written when the channel is not operating (EN=0 for DMA_CFGRx). After the channel is turned on this register becomes read-only and indicates the number of remaining pending transfer (the register content is decremented after each DMA transfer). When the channel is in cyclic mode, the contents of the register will be automatically reloaded to the previously configured value.	0

8.3.5 DMA Channel x Peripheral Address Register (DMA_PADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x10 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA[31:0]																															

Bit	Name	Access	Description	Reset value
[31:0]	PA[31:0]	RW	Peripheral base address, which serves as the source or destination address for peripheral data transfer. When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit0 and the operation address is automatically 2-byte aligned; when PSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0] and the operation address is automatically 4-byte aligned.	0

Note: This register can only be changed when EN=0 and cannot be written when EN=1.

8.3.6 DMA Channel x Memory Address Register (DMA_MADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x14 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA[31:0]																															

Bit	Name	Access	Description	Reset value
[31:0]	MA[31:0]	RW	The memory data address, which serves as the source or destination address for data transfers. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2-byte aligned; when MSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4-byte aligned.	0

Note: This register can only be changed when EN=0 and cannot be written when EN=1.