Module SPI

Direct Memory Access (DMA)

Direct Memory Access (DMA)

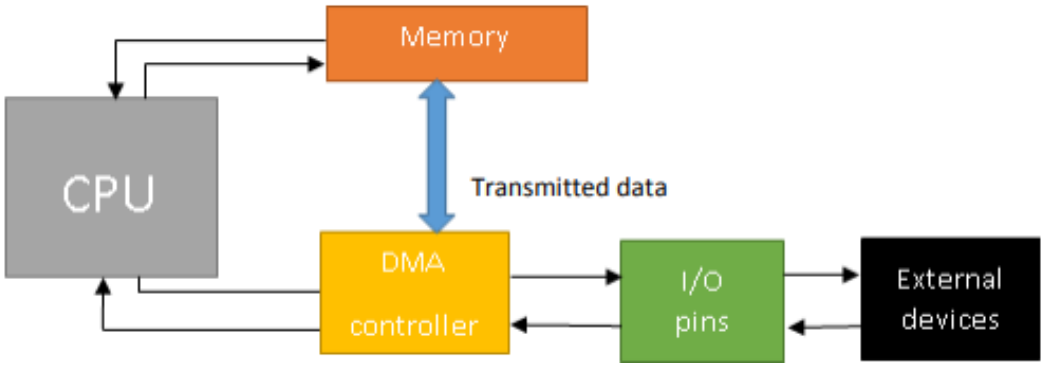
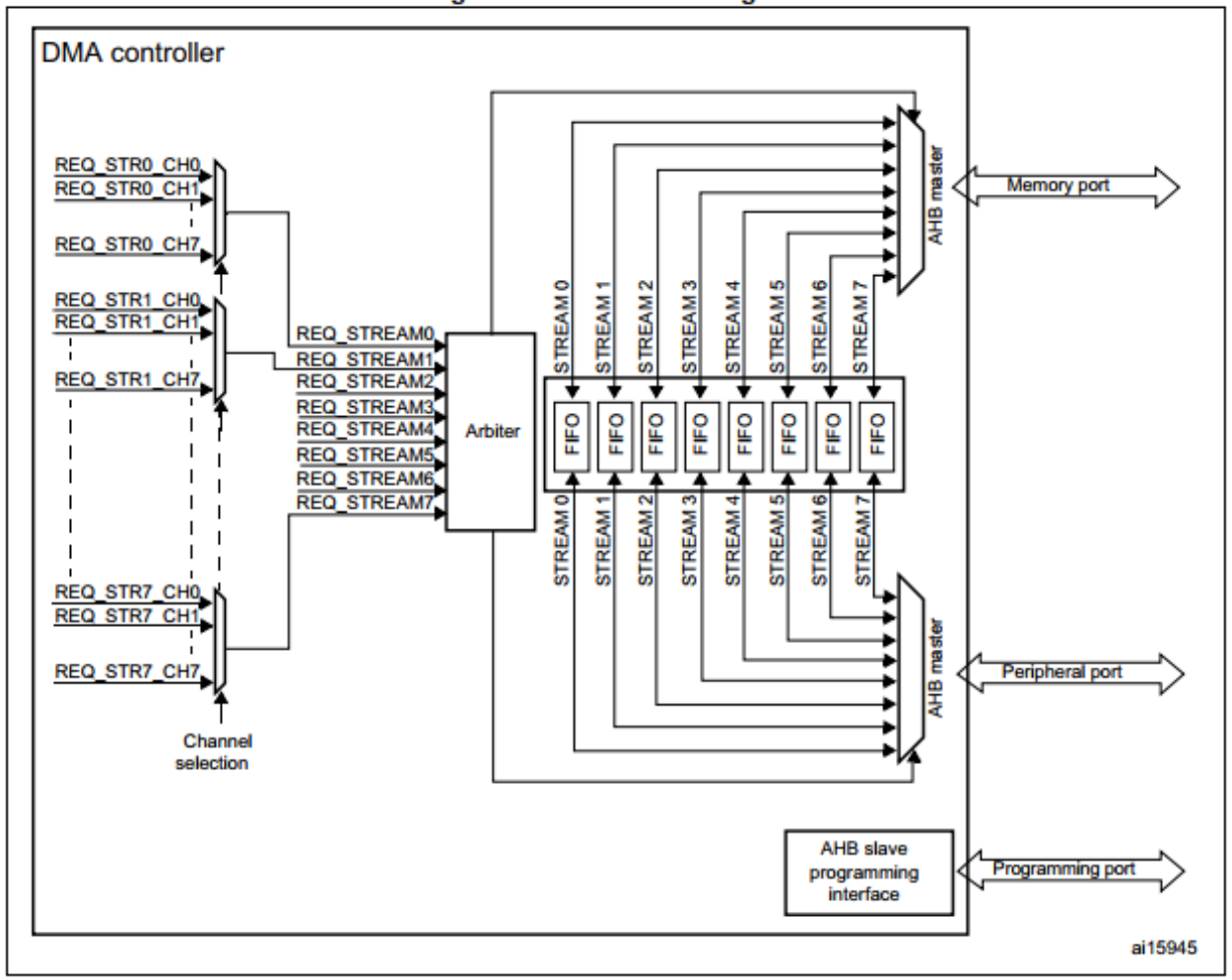


Figure 22. DMA block diagram



Direct Memory Access (DMA)

Thynk Loop

Table 29. DMA2 request mapping (STM32F401xB/C and STM32F401xD/E)

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1	-	-	-	ADC1	-	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
Channel 1	-	-	-	-	-	-	-	-
Channel 2	-	-	-	-	-	-	-	-
Channel 3	SPI1_RX	-	SPI1_RX	SPI1_TX	-	SPI1_TX	-	-
Channel 4	SPI4_RX	SPI4_TX	USART1_RX	SDIO	-	USART1_RX	SDIO	USART1_TX
Channel 5	-	USART6_RX	USART6_RX	SPI4_RX	SPI4_TX	-	USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
Channel 7	-	-	-	-	-	-	-	-

Direct Memory Access (DMA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CHSEL[2:0]			MBURST [1:0]		PBURST[1:0]		Reser-ved	CT	DBM	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

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

Bits 27:25 **CHSEL[2:0]**: Channel selection

These bits are set and cleared by software.

000: channel 0 selected
001: channel 1 selected
010: channel 2 selected
011: channel 3 selected
100: channel 4 selected
101: channel 5 selected
110: channel 6 selected
111: channel 7 selected

Bits 17:16 **PL[1:0]**: Priority level

These bits are set and cleared by software.

00: Low
01: Medium
10: High
11: Very high

These bits are protected and can be written only if EN is '0'.

Bits 14:13 **MSIZE[1:0]**: Memory data size

These bits are set and cleared by software.

00: byte (8-bit)
01: half-word (16-bit)
10: word (32-bit)
11: reserved

These bits are protected and can be written only if EN is '0'.

In direct mode, MSIZE is forced by hardware to the same value as PSIZE as soon as bit EN = '1'.

Bits 7:6 **DIR[1:0]**: Data transfer direction

These bits are set and cleared by software.

00: Peripheral-to-memory
01: Memory-to-peripheral
10: Memory-to-memory
11: reserved

These bits are protected and can be written only if EN is '0'.

Direct Memory Access (DMA)

➢ **Data Transfer Operations**

•Each DMA transfer involves:

- **Loading** from a source (peripheral/memory) via **DMA_SxPAR** or **DMA_SxM0AR**.
- **Storing** to a destination (peripheral/memory) via **DMA_SxPAR** or **DMA_SxM0AR**.
- **Post-decrementing DMA_SxNDTR** (tracks remaining transactions).

➢ **Request and Acknowledge Mechanism**

•Peripheral sends a **DMA request** to start data transfer.

•The DMA controller, based on priority, accesses the peripheral and sends an **Acknowledge** signal.

•The peripheral stops requesting once acknowledged, and the DMA controller releases the signal.

Registers Involved

•**DMA_SxCR** (Control Register): Configures DMA mode, priority, direction, and transfer size.
•**DMA_SxNDTR** (Number of Data Register): Stores the number of transfers remaining.
•**DMA_SxPAR** (Peripheral Address Register): Holds the peripheral address for data transfer.
•**DMA_SxM0AR / DMA_SxM1AR** (Memory Address Registers): Holds the memory address for data transfer.