Disabling a peripheral with the same ID as SPIS will not reset any shared SPIS registers. Configure all SPIS registers to ensure they operate correctly.

See the Instantiation table in Instantiation on page 216 for details on peripherals and their IDs.

8.20.3 EasyDMA

SPIS implements EasyDMA for accessing RAM without CPU involvement.

SPIS implements the EasyDMA channels found in the following table.

Channel	Type	Register Cluster
TXD	READER	TXD
RXD	WRITER	RXD

Table 62: SPIS EasyDMA Channels

For detailed information regarding the use of EasyDMA, see EasyDMA on page 27.

If RXD.MAXCNT is greater than TXD.MAXCNT, the remaining transmitted bytes will contain the value defined in the ORC register.

The END event indicates that EasyDMA is finished accessing the RAM buffer.

8.20.4 SPIS operation

SPIS uses two memory pointers. RXD.PTR points to the RXD buffer (receive buffer) and TXD.PTR points to the TXD buffer (transmit buffer). Because these buffers are located in RAM, which can be accessed by both SPIS and the CPU, a hardware based semaphore mechanism is implemented to enable safe sharing.

The CPU must acquire the SPI semaphore before it can safely update the RXD.PTR and TXD.PTR pointers. The ACQUIRE task must be triggered for the CPU to receive the ACQUIRED event and have access to the semaphore. When the CPU has updated the RXD.PTR and TXD.PTR pointers, the CPU must release the semaphore before SPIS can acquire it.

The CPU releases the semaphore by triggering the RELEASE task, as illustrated in the following figure. Triggering the RELEASE task when the CPU does not have access to the semaphore will have no effect. See Semaphore operation on page 614 for more information.