DMA transfer

Memory Buffer

PLB Central DMA core contains of a 16X64 internal data buffer, which performs the following:

- Supports the PLB burst transfers to speed up the DMA operation
- 32-bit transfers based on the present source and destination address. The mirroring operation (for the 32-bit transfers) is done before the data is put into the internal data buffer, such that the mirrored data is put on the PLB interface during the write cycle

PLB Central DMA Controller Operation

The PLB Central DMA controller operates on the PLB as a master/slave device. It responds as a slave when its registers are being read and written. As a master, it initiates read and write transactions when a DMA operation has been started.

The operation of the PLB Central DMA is initiated by writing values into the following DMA registers. For more information, see the PLB Central DMA Controller Register Descriptions section.

- DMA Control Register (DMACR): The contents of this register set the main parameters for the DMA transfer as follows (refer to Table 7):
  - Source Increment (SINC): SINC should be set to '0', if and only if the Source Address register is written with a *keyhole* address such that a single address is associated with a sequence of data[1]. If the source address should increment for each data transferred, SINC should set to '1'.
  - Destination Increment (DINC): DINC should be set to '0' if and only if the Destination Address is written with a *keyhole* address such that a single address is associated with a sequence of data[2]. If the destination address should increment for each data transferred, DINC should set to '1'.
  - DMA Transfer Data Size (DSIZE): It is set to 4 bytes or 8 bytes, to select the data size to be used for individual bus transfers.
- LENGTH register (LENGTH): The number of bytes to transfer is written into this register. Writing of this register is the event that starts the DMA operation, so it must be done last. (Refer to Table 10 for more information on this register)

The data transfer starts by reading data from the source address into the internal data buffer followed by the transfer of data from the internal buffer to the destination address. This repeats until all the data is moved. The status registers get updated as the DMA operation progresses.

1. An example of a keyhole address is a memory mapped FIFO that maps as an element at a single address but can consume or produce an endless sequence of data. A variation on a keyhole address is a *wide keyhole* where a number of consecutive addresses map to the element. When SINC = '0', PLB Central DMA will perform all read transactions to the same address. It is the responsibility of the corresponding slave to impose a keyhole behavior. For bursts, this implies that the slave will not increment the keyhole address during the burst or will make the keyhole at least as wide as the burst size. It is not feasible to use SINC = '0' with a non-keyhole slave such as a memory. Even though the DMA Controller would deliver a non-incremented address during the address phase of PLB transactions, the memory would increment it during bursts. (See also DINC, which is subject to the same considerations when set to '0'.)
2. See the above note for SINC = '0'. The case DINC = '0' has the same considerations but for the destination address.