Information:

-Refer to the MPMC v4.03a documentation for more information on NPI interfacing and timing diagrams. [Page 20-23 for NPI pin connections] [Page 28 for NPI dependencies] [Page 133-149 for NPI timing diagrams]

-This system was built to use the NPI 8-word, cacheline write [Page 139]

-More information on creating a NPI core can be found at:
http://www.xilinx.com/support/answers/24912.htm

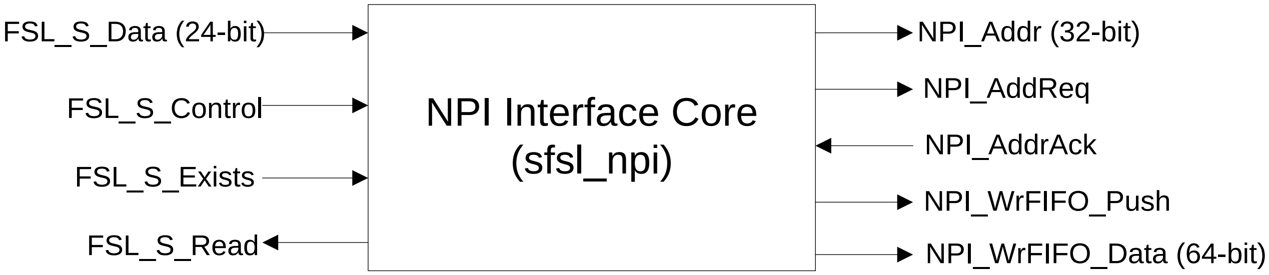
-This core is made available as an example/template of an NPI write system. You are free to edit the files to your needs.

-Refer to the mpd (microprocessor peripheral definition) file to edit the bus connections in the core.

Core Description

This is a custom core that was designed to interface between the FSL and the DDR memory interface. The purpose of this core was to write data coming through the FSL into the desired location in memory. On one port there is an FSL slave bus to receive both the pixel data and the frame control bit, on the other port is the NPI (native port interface) connection which provides a direct means of connection to the MPMC (multi-port memory controller) in order to issue write commands.

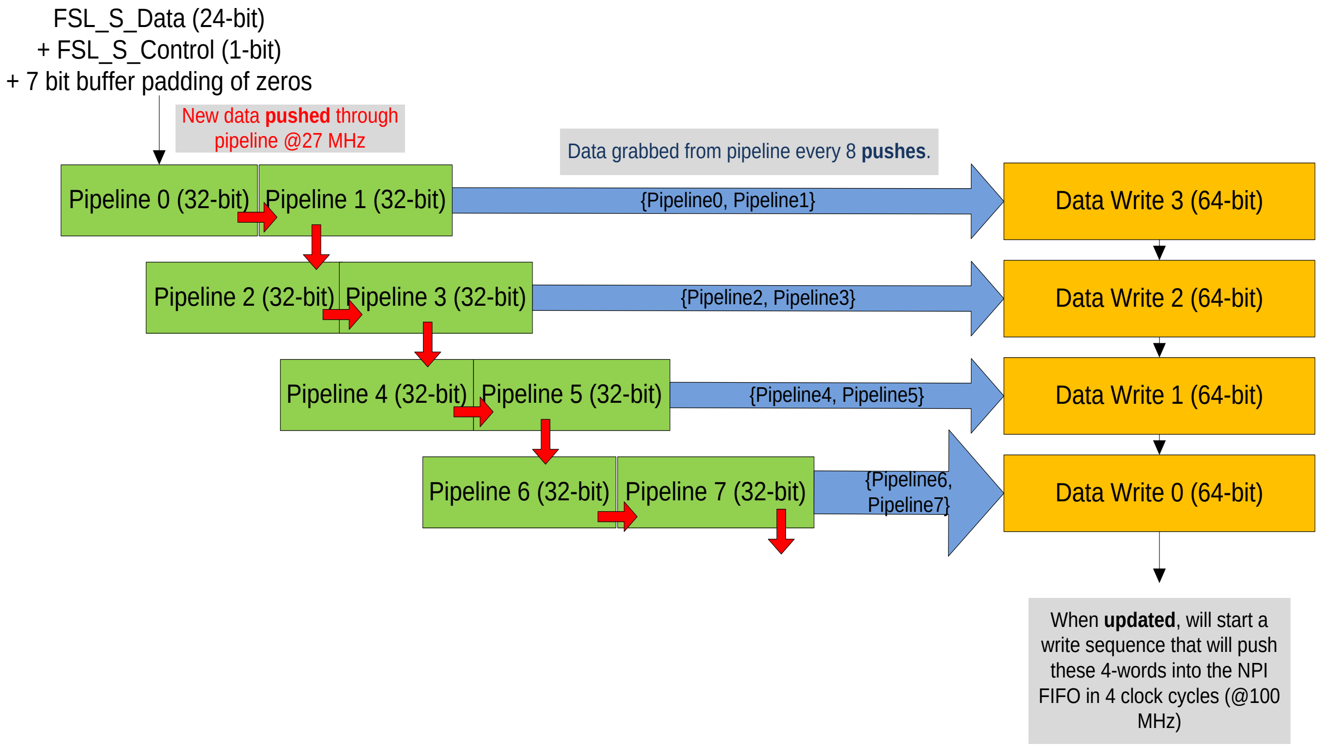
Figure 1: Simplified Port Description of NPI Interface Core



Because this core is operating at the MPMC frequency (100 MHz), it is capable of pulling data out of the FSL FIFO at a faster rate than the data is being pushed into the FSL FIFO (27 MHz). Consequently, there will be times where the FSL FIFO is empty and has no valid data. Valid data in the FSL FIFO can occur within 3 to 4 clock cycles (at 100 MHz) of each other, and is stated by the FSL_S_Exists line going high.

When the FSL_S_Exists signal goes high the custom core will pull the data from the FSL_S_Data bus and the FSL_S_Control bit to store into a 32-bit buffered pipeline. As well, the FSL_S_Read signal will be asserted to tell the FSL to update the data and control bit at the next clock cycle. If there is no new valid data, the FSL will de-assert the FSL_S_Exists signal.

Figure 2: NPI Interface Core Functionality



The custom NPI interface core continuously pulls data from the FSL and pushes it into an 8-word 32-bit pipeline that is built into the core. When this pipeline has eight push events from the FSL, the pipeline data is extracted and written into the NPI FIFO (and thus to DDR memory). The method of writing to the NPI FIFO is an 8-word cacheline, or a 4-word 64-bit writing process. To do this the NPI_WrFIFO_Push signal is asserted for 4 clock cycles, and with every clock cycle the data on the NPI_WrFIFO_Data bus is changed to the desired 64-bit word. On the last clock cycle the NPI_AddrReq (address request) line is asserted and the address that has been placed on the NPI_Addr bus is taken as the base location to be written to. After 1-2 clock cycles the NPI will send its acknowledgement of the address request and the core can de-assert its request line.

The process continues indefinitely, and before every write process the NPI_Addr will be updated with a new address that is 32-bytes higher than the previous address. This ensures that data is being written into the correct location. When the FSL_S_Control bit is high the system has a reset and the address location switches to the base address of either frame buffer 0 or frame buffer 1 (depending on which frame buffer the core was writing to previously). If the core was writing into frame buffer 0, then at the next FSL_S_Control bit, the core would switch to the base address of frame buffer 1, and vice versa. Both of these frame buffer variables are easily changed within XPS.