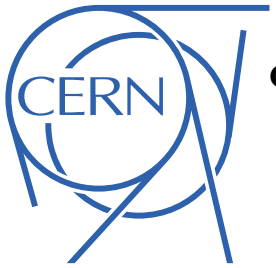

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN BE-CO-HT

Technical Specification
Gennum GN4124 Core For FMC Projects

July 2010

Edited by:
Simon Deprez

Checked by:

Abstract

This technical specification describes the HDL controller for Gennum GN4124 chip. It provides a Wishbone master for control and status registers access and a DMA controller for high speed data transfers.

Revision History

Version	Date	Notes
0.1	10-06-2010	Initial release
0.2	12-07-2010	First draft
0.3	20-07-2010	Add byte swapping

Contents

List of Figures

Introduction

The GN4124 core is a controller for the GN4124 chip on the PCIe FMC Carrier[1] (PFC) and the Simple PCIe FMC carrier[2] (SPEC). The functional specification[3] describes the interfaces of the core and the configuration registers of the DMA controller.

This specification describes how the core works internally. For each internal block, we give a summary description of its function. Figure 1 show the core with all main blocks.

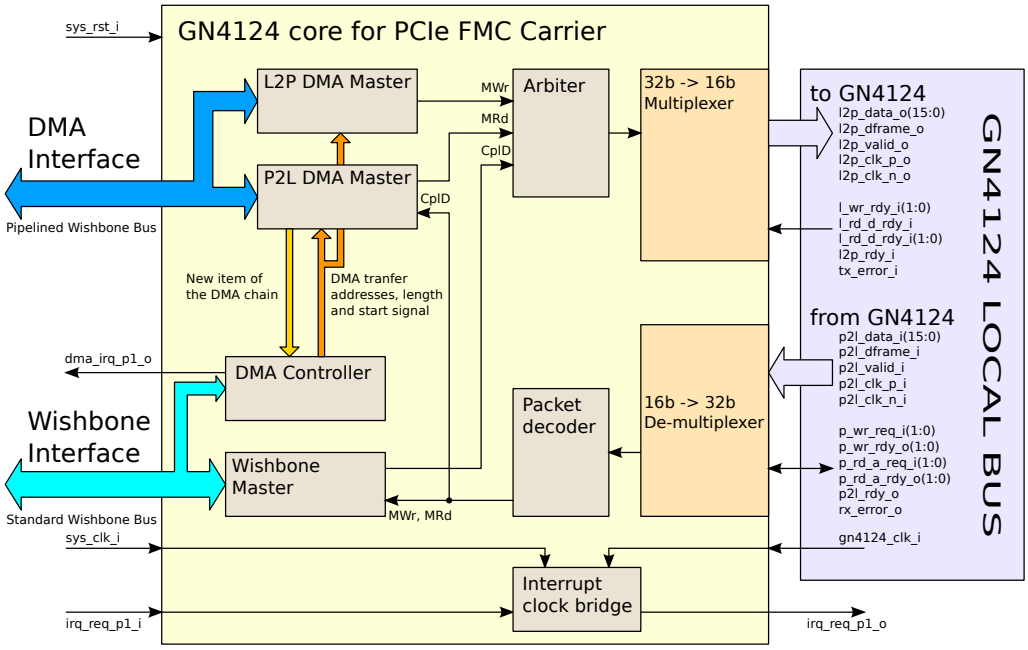

Figure 1: GN4124 core for PCIe FMC carrier

The structure of this cores and some signals names are based on the Gennum cores *Lambo* and *Lotus*.

The VHDL guidelines[4] describes the rules used for writing the HDL code and the signals naming conventions. The one-tick-long pulses are named with a `_p1` suffix.

1 De-multiplexer

This block takes the double data rate 16-bit P2L (PCI Express to Local Bus direction) bus from the GN4124 device and converts it to a single data rate 32-bit bus for use inside the GN4124 core.

2 Multiplexer

It takes the internal single data rate 32 bit data and transmits it as double data rate 16-bit data on the L2P (Local Bus to PCI Express direction) bus.

[1] See http://www.ohwr.org/projects/fmc-pci-carrier.
[2] See http://www.ohwr.org/projects/spec.
[3] See http://svn.ohwr.org/gn4124-core/trunk/documentation/specifications/func_spec_GN4124_core.pdf.
[4] See http://www.ohwr.org/attachments/27/VHDLcoding.pdf.

3 Packet decoder

This block extracts header information, address, data, byte enables, and timing controls of the packets from the GN4124 chip. It provides signals like:

- Type of packet (target read request, target write request, master read completion ...)
- Begin of packet
- Address that will increment with data for data block transfer (e.g. DMA transfers)
- Data
- End of packet

4 Wishbone master

The Wishbone master implements a master for the Wishbone interconnection bus. It transforms a PCIe write into a Wishbone write and a PCIe read into a Wishbone read. Only single word reads and writes are supported on the Wishbone interface.

Data are coming from the packet decoder. Wishbone signals are generated and the master waits for an acknowledge. Write cycles are queued up at full local bus speed into a small FIFO (16 words deep) so that the write cycles can be played out at the Wishbone bus speed. In order to prevent the FIFO from overflowing due to too many writes coming from the host, the `p_wr_rdy_o` signal is de-asserted to the GN4124 whenever the FIFO is not empty.

The `p_wr_rdy_o` signal prevents write requests from the PCI Express bus to the Wishbone configuration interface. The read completions of the P2L DMA transfers are not slowed down.

5 DMA Engine

5.1 DMA controller

NAME	OFFSET	MODE	RESET	DESCRIPTION
DMACTRLR	0x00	R/W	0x00000000	DMA engine control
DMASTATR	0x04	RO	0x00000000	DMA engine status
DMACSTARTR	0x08	R/W	0x00000000	DMA start address in the carrier
DMAHSTARTLR	0x0C	R/W	0x00000000	DMA start address (low) in the PCIe host
DMAHSTARTHR	0x10	R/W	0x00000000	DMA start address (high) in the PCIe host
DMALENR	0x14	R/W	0x00000000	DMA read length in bytes
DMANEXTLR	0x18	R/W	0x00000000	Pointer (low) to next item in list
DMANEXTHR	0x1C	R/W	0x00000000	Pointer (high) to next item in list
DMAATTRIBR	0x20	R/W	0x00000000	DMA chain control

Table 1: Register set for the DMA controller block

The DMA controller is a Wishbone slave controlled from the PCI Express host with the Wishbone master (see the functional specification). Figure 2 shows the internals of the DMA controller.

The transfer starts when the first bit (LSB) of the DMACTRLR (See table 1) register is asserted. One of the two signals, `dma_ctrl_start_L2P_o` and `dma_ctrl_start_P2L_o`, is set to '1' for one clock cycle. The `dma_ctrl_start_L2P_o` signal controls the DMA L2P master that performs transfers from

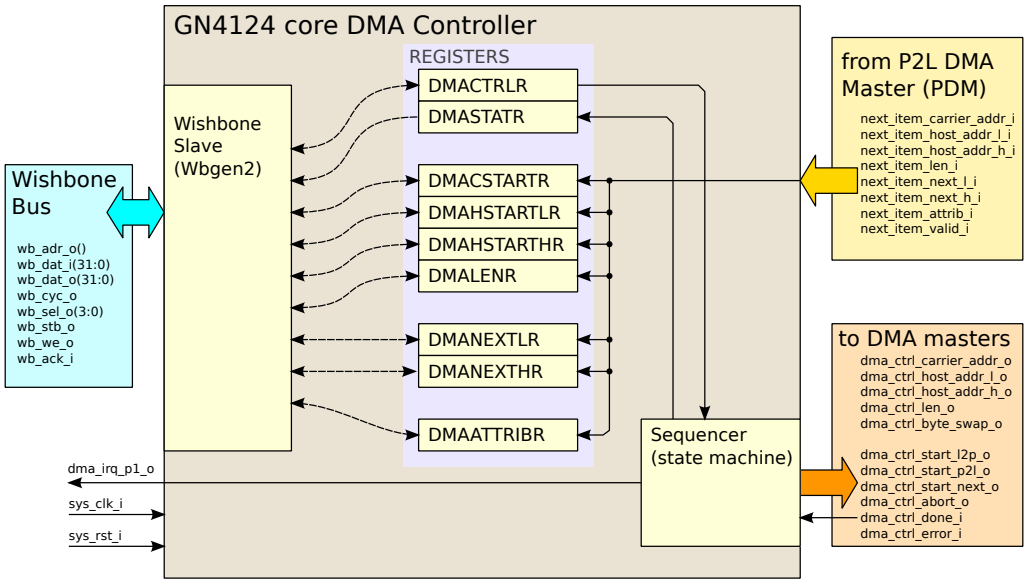

Figure 2: DMA controller

the carrier to the PCI Express host and the `dma_ctrl_start_P2L_o` signal controls the DMA P2L master that performs transfers from the PCI Express host to the carrier.

The DMA controller waits for the `dma_ctrl_done_i` signal from one of the two DMA masters that indicates the end of the transfer. The `dma_ctrl_error_i` signal indicates the end of the current transfer if an error occurs.

After the end of a transfer, if this transfer is not the last, the controller ask to the P2L DMA master the new item of the DMA chain. The `next_item_valid_i` signal indicates that the DMA master can read the new DMA chain item from the P2L DMA master and starts the new cycle.

5.2 L2P DMA Master

The L2P DMA master (See figure 4) performs data transfers from the FMC carrier to the memory of the PCI Express host. The transfer is split in blocks of data with a maximal length of 4096 bytes. This is the maximal length of transfers allowed by the GN4124 chip.

The L2P DMA master sends a master read request on the DMA interface and waits for answer. This transfer is clocked by the `sys_clk_i` signal. The received data are stocked in a FIFO that allows switching between clock domains.

When the block of data is in the FIFO, the DMA master sends a master write request to the PCI Express host. This operation is clocked by the GN4124 Local Bus Clock. The transfer is paused if the `l_wr_rdy_i` signal from GN4124 chip is not asserted.

5.3 P2L DMA Master

The P2L DMA master (See figure 5) performs data transfers from the PCI Express host to the FMC carrier. The transfer is split in blocks of data with a maximal length of 4096 bytes.

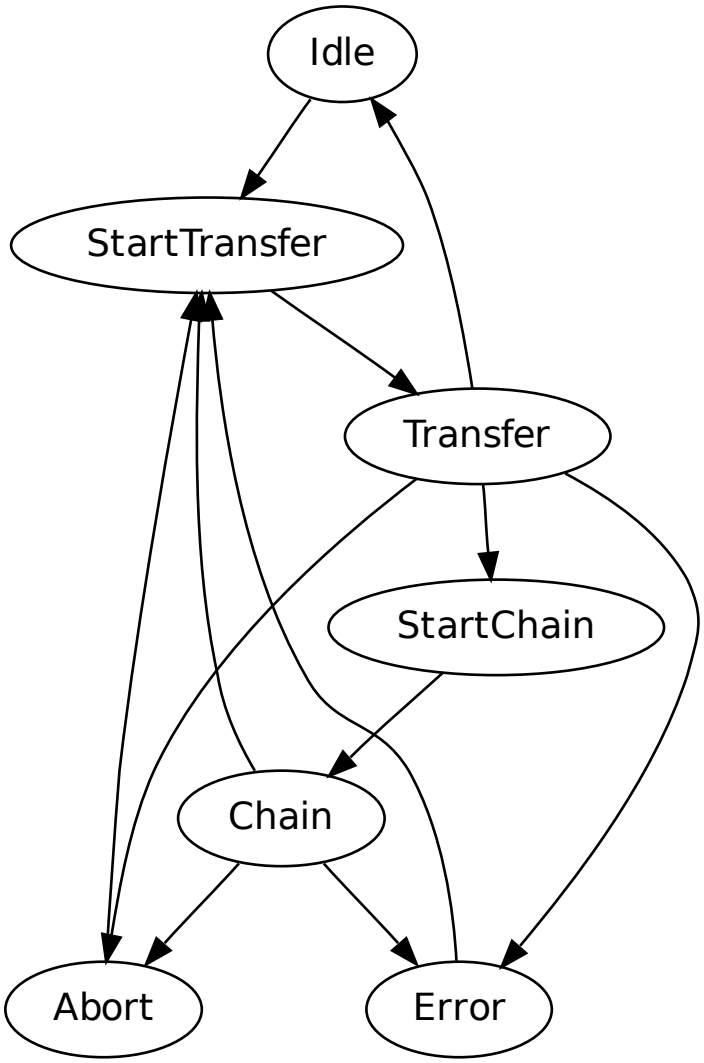

Figure 3: DMA controller state machine

The P2L DMA master sends a memory read request toward the GN4124 chip and waits for the answer. This operation is clocked by the GN4124 Local Bus Clock. The received data are stocked in a FIFO.

When the block of data is in the FIFO, the DMA master starts a master write cycle on the DMA interface of this block. This transfer is clocked by the `sys_clk_i` signal.

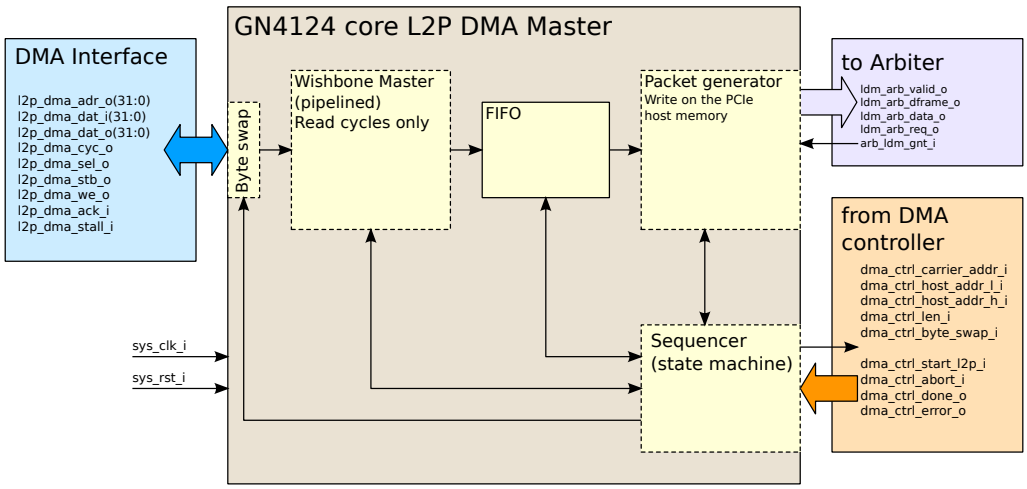

Figure 4: L2P DMA master

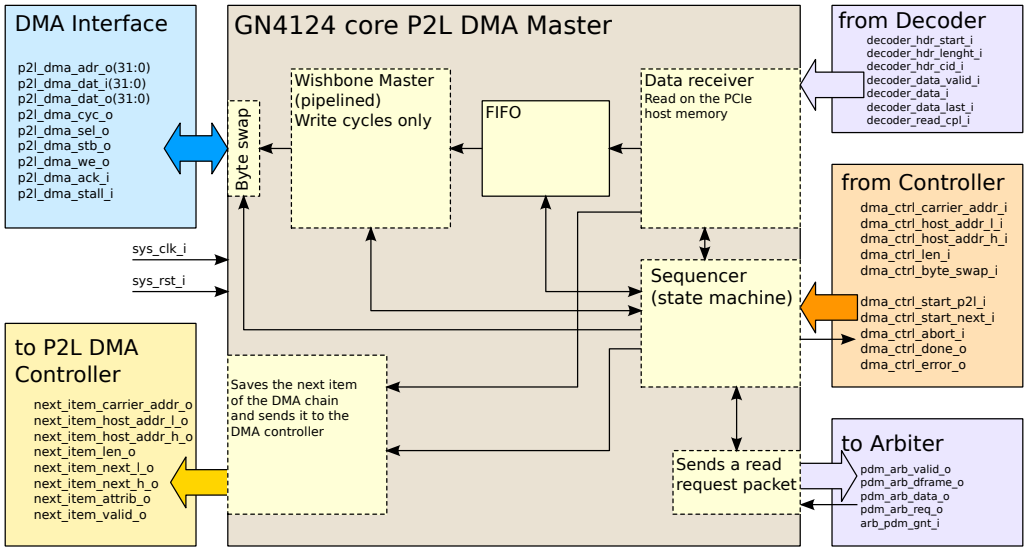

Figure 5: P2L DMA master

6 Arbiter

Arbitrate between Wishbone master and the two DMA masters. The arbiter is waiting for a request signal from one of this blocks and it grants the bus to the first requester until the end of the packet.

The highest priority is for the Wishbone master, then for the P2L DMA master, and the lower priority is for the L2P DMA master.

7 Interrupt clock bridge

It transforms input interrupt one-tick-long pulse clocked by `sys_clk_i` in a one-tick-long pulse clocked by the GN4124 local bus clock.