POWERLINK IP-Core

Xilinx FPGA Documentation

Model No:	OAT113110.11
Version:	1.0
Date:	2013-02-12

We reserve the right to change the content of this manual without prior notice. The information contained herein is believed to be accurate as of the date of publication, however, B&R makes no warranty, expressed or implied, with regards to the products or the documentation contained within this document. B&R shall not be liable in the event if incidental or consequential damages in connection with or arising from the furnishing, performance or use of these products. The software names, hardware names and trademarks used in this document are registered by the respective companies.

I General information

a. Manual history

Version	Date	Comment
1.0	2012-12-13	First Edition

Table 1 Versions

b. Safety notices

Safety notices in this document are organized as follows:

Safety notice	Description
Danger!	Disregarding the safety regulations and guidelines can be life-threatening.
Warning!	Disregarding the safety regulations and guidelines can result in severe injury or heavy damage to material.
Caution!	Disregarding the safety regulations and guidelines can result in injury or damage to material.
Information:	Important information used to prevent errors.

Table 2 Safety notices

c. License

This product and all of its parts are released under the terms of the following license:

Copyright (c) 2013, Bernecker+Rainer Industrie-Elektronik Ges.m.b.H. (B&R)
Copyright (c) 2013, SYSTEC electronik GmbH
Copyright (c) 2013, Kalycito Infotech Private Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

II Table of Contents

a. Manual history 2
b. Safety notices 2
c. License 2

1 Introduction 4

2 Xilinx® Platform Studio (XPS) Integration 5

3 POWERLINK IP-Core GUI 7
3.1 General IP-Core settings 7
3.2 openMAC IP-Core settings 8
3.3 PDI settings 10
3.4 Parallel interface settings 11
3.5 SPI interface settings 12
3.6 Direct IO interface settings 12
3.7 IP-Core Misc tab 12

4 Software Interface 13

5 FPGA Resource Utilization 14

6 Files 15

A Abbreviations 17

A References 18

B Figure index 19

C Table index 20

Index 21

1 Introduction

The POWERLINK IP-Core is provided as ready-to-use in Xilinx® Platform Studio (XPS) version 13.2. This documentation presents the integration of the IP-Core into the XPS, as well as timing considerations and interfacing to the software development environment.

ⓘ If detailed information about the POWERLINK IP-Core itself is required, please refer to the "*POWERLINK IP-Core Generic Documentation*" (02_POWERLINK-IP-Core_Generic.pdf).

The POWERLINK IP-Core is implemented generically, i.e. it is possible to use the same VHDL-sources in any FPGA environment. For instance memory components are given in extra VHDL-files ready to use in Spartan 6 devices. Experienced users can modify these files easily to port the IP-Core to other devices.

The IP-Core configuration is supported by an easy-to-use graphical user interface (GUI) in XPS. This enables the user to define several parameters for the POWERLINK node, like the type of PDI (e.g. SPI or parallel interface), the number of process data objects (PDO) or enabling low-jitter synchronization feature.

ⓘ VHDL or FPGA knowledge is not necessarily required to setup a POWERLINK node on an FPGA. The entire necessary configuration is done depending on the settings defined in the XPS GUI.

2 Xilinx® Platform Studio (XPS) Integration

The Xilinx® POWERLINK IP-Core is available in two versions which are called axi_powerlink and plb_powerlink. One provides an interface for PLB and the second one provides AXI. (In order to stay compatible with future versions of the XPS development environment it is advised to use AXI if possible)

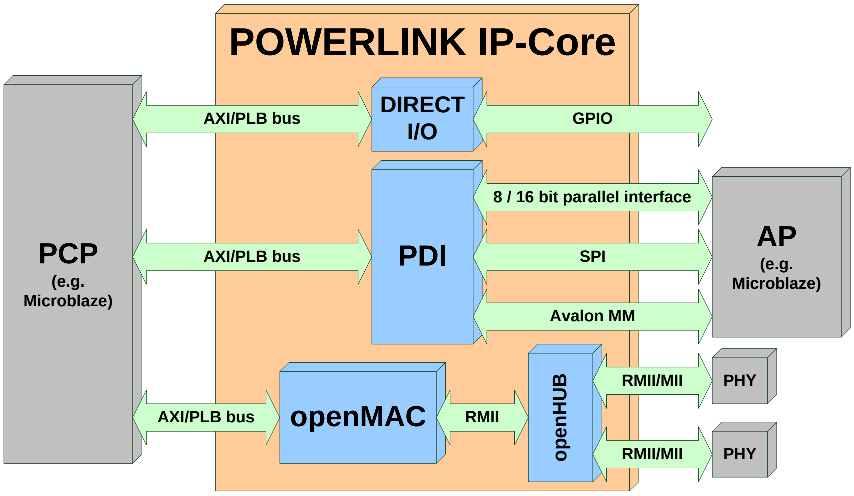
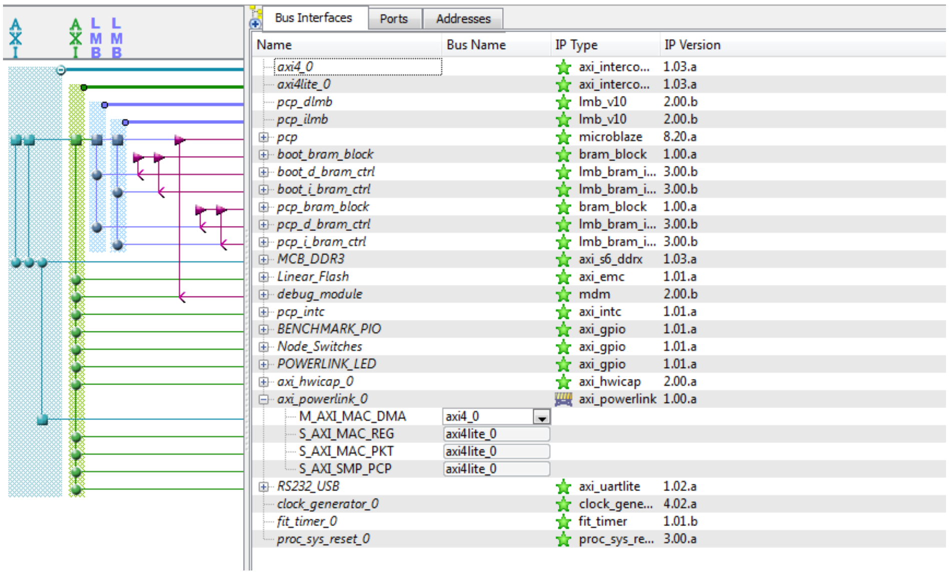


Figure 1 POWERLINK IP-Core Block Diagram

The XPS allows easy configuration of the IP-Core, thus the user is able to define parameters depending on the application of the POWERLINK Node. Figure 1 visualizes the block diagram of the IP-core, emphasizing the interfaces to the POWERLINK Communication Processor (PCP) and to the application with the optional Application Processor (AP). Please note that the PDI and DIRECT I/O blocks are mutual exclusive, depending on the IP-Core configuration. However, the MAC-layer components openMAC and openHUB[1] are mandatory components of a POWERLINK node.



Name	Bus Name	IP Type	IP Version
axi4_0		axi_interco...	1.03.a
axi4lite_0		axi_interco...	1.03.a
pcp_dlmb		lmb_v10	2.00.b
pcp_ilmb		lmb_v10	2.00.b
⊞ pcp		microblaze	8.20.a
⊞ boot_bram_block		bram_block	1.00.a
⊞ boot_d_bram_ctrl		lmb_bram_i...	3.00.b
⊞ boot_i_bram_ctrl		lmb_bram_i...	3.00.b
⊞ pcp_bram_block		bram_block	1.00.a
⊞ pcp_d_bram_ctrl		lmb_bram_i...	3.00.b
⊞ pcp_i_bram_ctrl		lmb_bram_i...	3.00.b
⊞ MCB_DDR3		axi_s6_ddrx	1.03.a
⊞ Linear_Flash		axi_emc	1.01.a
⊞ debug_module		mdm	2.00.b
⊞ pcp_intc		axi_intc	1.01.a
⊞ BENCHMARK_PIO		axi_gpio	1.01.a
⊞ Node_Switches		axi_gpio	1.01.a
⊞ POWERLINK_LED		axi_gpio	1.01.a
⊞ axi_hwicap_0		axi_hwicap	2.00.a
⊟ axi_powerlink_0		axi_powerlink	1.00.a
M_AXI_MAC_DMA	axi4_0		
S_AXI_MAC_REG	axi4lite_0		
S_AXI_MAC_PKT	axi4lite_0		
S_AXI_SMP_PCP	axi4lite_0		
⊞ RS232_USB		axi_uartlite	1.02.a
clock_generator_0		clock_gene...	4.02.a
fit_timer_0		fit_timer	1.01.b
proc_sys_reset_0		proc_sys_re...	3.00.a

Figure 2 Example XPS design with the AXI POWERLINK IP-Core

In Figure 2 an XPS example is shown, which includes the PCP as a Xilinx® Microblaze processor, DDR3 memory controller and other components available in the design. Depending on the configurations of the

[1] openHUB is only required if a second Ethernet interface is available on your hardware platform.

POWERLINK IP-Core the AXI/PLB slave and master interfaces may vary. The following bus interfaces may be visible in the XPS bus interfaces tab at the POWERLINK IP-Core:

- **MAC_DMA**: Master interface for transferring RX/TX packets to external memory.
- **MAC_REG**: Slave interface for accessing the MAC and timer registers. (always mandatory)
- **MAC_PKT**: Slave interface for accessing the RX/TX packets in internal BRAM.
- **PDI_PCP**: Slave interface for the PCP to access the Process Data Interface (PDI).
- **PDI_AP**: Slave interface for the AP to access the Process Data Interface (PDI).
- **SMP_PCP**: Slave interface for accessing the Direct IO module in the IP-Core.

Figure 4 shows the GUI for the POWERLINK IP-Core with the openMAC settings. The parameters are separated into sub-groups (e.g. openMAC, PDI or parallel interface). The user has to set the intended POWERLINK Slave Design Configuration, which affects the subsequent parameters' accessibility e.g. in case of "openMAC only" PDI settings are not usable because they are not meaningful.

The configuration inputs set in the GUI by the user are tested on feasibility and the user is informed with possible errors or warnings in the bottom of the window.

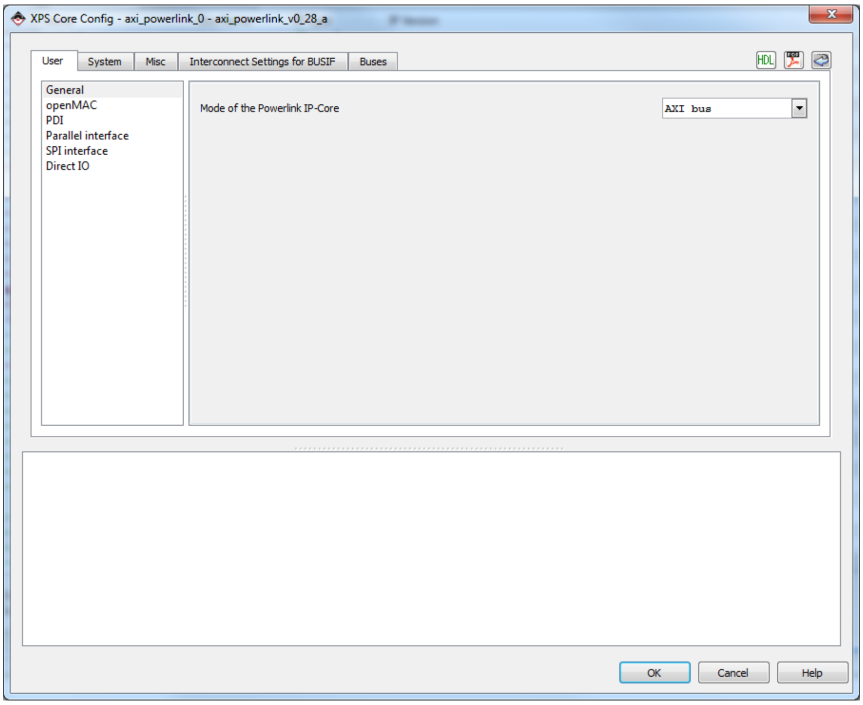
3 POWERLINK IP-Core GUI

This chapter provides details about the IP-Core GUI and its options. The GUI is responsible for all basic settings which can be done in the IP-Core. Some of those settings will be automatically used for the software generation and configuration (cf. chapter 4).

The GUI consists of several tabs where the “User” tab consists of the most important settings. The different sections inside this tab are explained in detail in the following sub sections.

3.1 General IP-Core settings

With the “General” IP-core settings (see Figure 3) it is possible to change the IP-core mode. This mode determines which other parameters are available inside the IP-Core GUI. In addition, also the bus interfaces and IO ports can change and require to be connected differently inside the XPS.



Mode of the Powerlink IP-Core: AXI bus

Figure 3 General settings tab of the axi_powerlink IP-Core

Table 3 provides a detailed description of the available POWERLINK IP-Core design configuration modes. All modes are available for both the AXI and PLB interconnect.

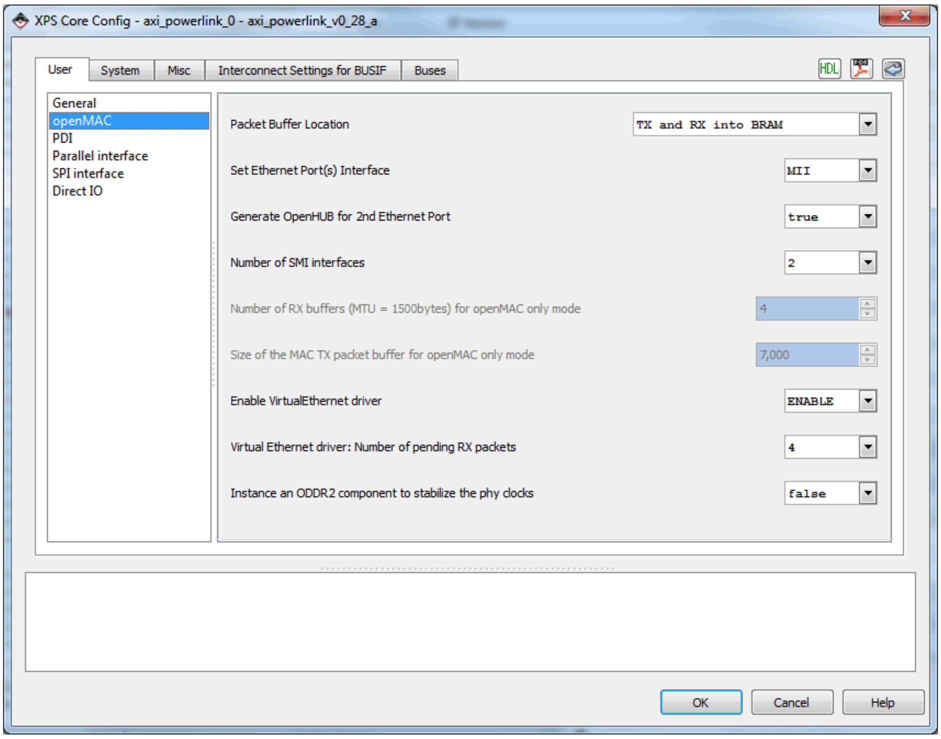
Mode	Description
DirectIO	Design with openMAC and additional 32 IO lines which can be configured as input and output by using the provided configuration port. (No interface for a second processor available)
Parallel Interface (8/16bit)	Design with openMAC and an additional interface for a second processor. This interface provides a parallel bus with the choice of two different data bus widths: 8 or 16bits.
SPI Interface	Design with openMAC and an additional interface for a second processor. This interface consists of a serial bus with four wires. (Clk, MISO, MOSI, SS)
AXI/PLB Interconnect	Design with openMAC and an additional interface for a second processor. This interface provides an FPGA internal connection, to the second processor, by using the common AXI or PLB intercon-

	nects.
openMAC only	The openMAC mode only instantiates the MAC layer and provides no additional user interface.

Table 3 Modes of the POWERLINK IP-Core

3.2 openMAC IP-Core settings

By using the POWERLINK IP-Core the openMAC component, is always instantiated. This component handles the receiving and transmitting of POWERLINK frames. Figure 4 pictures all parameters which can be applied to the POWERLINK MAC layer.



Packet Buffer Location	TX and RX into BRAM
Set Ethernet Port(s) Interface	MII
Generate OpenHUB for 2nd Ethernet Port	true
Number of SMI interfaces	2
Number of RX buffers (MTU = 1500bytes) for openMAC only mode	4
Size of the MAC TX packet buffer for openMAC only mode	7,000
Enable VirtualEthernet driver	ENABLE
Virtual Ethernet driver: Number of pending RX packets	4
Instance an ODDR2 component to stabilize the phy clocks	false

Figure 4 GUI of AXI POWERLINK IP-Core with example configuration of the openMAC

Table 4 provides detailed information of all openMAC settings. Care is advised when changing some of these settings as the bus interfaces and IO ports of the IP-Core can change in some configurations. Connect these additional visible ports by using the XPS bus interfaces and ports tab.

Parameter	Description
Packet buffer location (RX/TX)	All packets received or transmitted by the MAC need to be stored somewhere in a memory. This option provides the possibility to either store the packets in internal BRAM or to forward them by the local interconnect (AXI/PLB) to an external memory. (SRAM or DRAM) The following Table 5 provides more information about this parameter.
Ethernet ports interface	MII and RMII are the two common interfaces for accessing the PHY internal configuration registers. The POWERLINK IP-Core can handle both interfaces but the use of **RMII** is preferred. If this parameter is set to MII an additional *rmii2mii* converter is added which needs more resources. (Logic and BRAM)
Generate openHUB	If this parameter is set to **true** the openHUB component is instantiated which enables additional ports for connecting a second PHY.
Number of SMI interfaces	In case of a second HUB it is possible to enable ports for a second

	SMI interface. (It is also possible to use one SMI interface for two PHY's)
Number of RX buffers (openMAC only mode)	If the IP-Core is in openMAC only mode no packet buffer size calculation is done by the GUI. Therefore the user needs to enter the number of RX buffers manually. For each buffer the maximum frame size of 1518 bytes is allocated. For each receive frame of one POWERLINK cycle and RX buffer needs to be reserved. For example a POWERLINK slave always needs RX buffers for the following frames: SoC, RPDO0, Soa, 7x Asnd (RPDO1 and RPDO2 are optionally used for cross traffic). Therefore a minimum number of **ten** RX buffers are required. (Seven Asnd buffers are needed if the master has multi Asnd enabled. Take into consideration that the limitation to seven Asnd frames is a B&R internal limit and is maybe increased in future versions.) **Note**: In order to guarantee full reception of all (possibly wrong) RX frames it is advised to store the RX packets externally and use all 16 available RX buffers.
Size of TX buffers (openMAC only mode)	If the IP-Core is in openMAC only mode no packet buffer size calculation is done by the GUI. Therefore the user needs to enter the size of the TX buffer manually. For example a POWERLINK slave always needs two TX buffers for the following frames: TPDO (1500bytes), Asnd (1500bytes), IdentResponse (180bytes), StatusResponse (76bytes), NMTRequest (180bytes) and SyncResponse (64bytes). ➔ This results in minimum TX buffer size of about: (1500bytes + 1500bytes + 64bytes + 180bytes + 76bytes + 64bytes) * 2 ~= 7000 bytes
Instance ODDR2 component	If the POWERLINK IP-Core is used in RMII mode and the RX/TX clock from the PHY's are connected to non-dedicated clock pins this option can instantiate a clock buffer manually.

Table 4 openMAC settings description

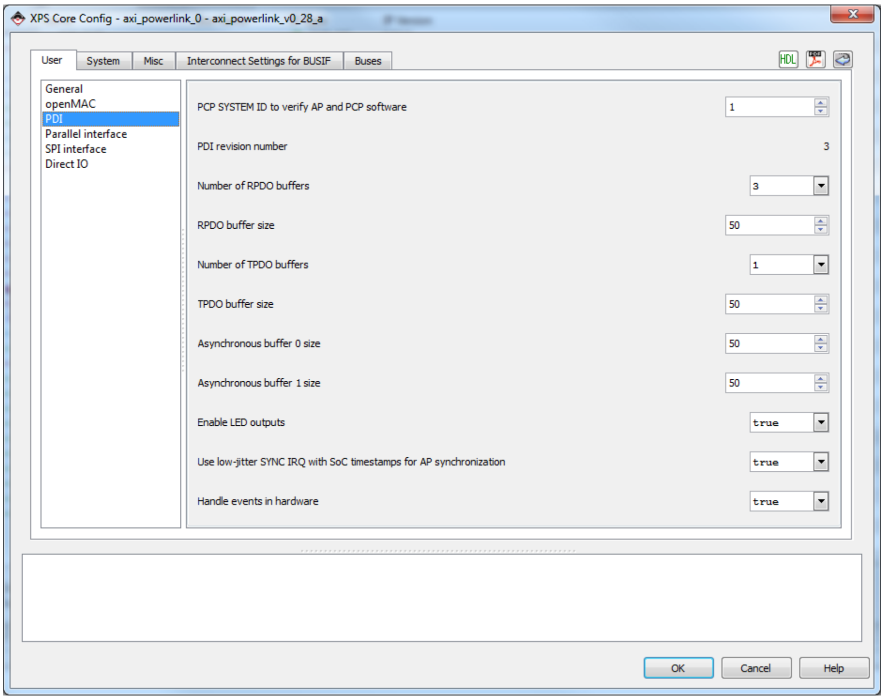
Packet buffer location setting	Description
TX and RX into BRAM	If this option is active RX and TX packets are stored in the internal BRAM. The packet buffer can be accessed by using the bus interface MAC_PKT. This option should be preferred if enough BRAM is available.
TX into BRAM and RX over AXI/PLB	This option stores the TX packets in internal BRAM and forwards the RX packets over the local bus interconnect to the external memory where the heap of the program is located. (The MAC_DMA master interface is used in this case!) The forwarding of the RX packets uses a RX FIFO to buffer the access to the external memory. The FIFO and burst size can be adjusted by using the option in the "*System"* tab in section "[AXI/PLB] MAC DMA".
TX and RX over AXI/PLB	If the packet buffer location is set to this option the TX and RX packets are forwarded over the local bus to the external memory. In this case no internal BRAM is used for the packet buffer. Only BRAM resources for the RX and TX DMA FIFOs are used to buffer the access to the external memory.

	WARNING: Storing the TX packets in the external memory can be very error prone. As several bus masters are connected to the external memory the access time to the memory can be very unpredictable. If the TX packet data is loaded to late from the memory, a corrupted frame has to be transmitted (since the deadline is not met). Nevertheless an error is reported to the Ethernet driver. Because of this, it is advised to not use this option until it is guaranteed that the external memory can provide the data in time. (MAC auto response sends after 960ns inter frame gap) On AXI systems the bus arbitration can be changed from round robin to fixed which enables a more stable response time of the external memory!

Table 5 Location of the MAC packet buffer

3.3 PDI settings

The options in this section are only available when the IP-Core is either in parallel- or SPI interface mode. Otherwise the PDI is deactivated and all options are disabled. Figure 5 pictures all available options in this category. Some of them can be deactivated in order to save resources.



PCP SYSTEM ID to verify AP and PCP software: 1

PDI revision number: 3

Number of RPDO buffers: 3

RPDO buffer size: 50

Number of TPDO buffers: 1

TPDO buffer size: 50

Asynchronous buffer 0 size: 50

Asynchronous buffer 1 size: 50

Enable LED outputs: true

Use low-jitter SYNC IRQ with SoC timestamps for AP synchronization: true

Handle events in hardware: true

Figure 5 Process data interface (PDI) configuration section of the axi_powerlink IP-Core

Detailed description of all parameters is given in Table 6. By changing these parameters the internal memory layout of the PDI can change and more internal BRAM is used. Care is advised when changing any of these parameters.

Parameter	Description
PCP System ID	This ID can be used to verify the software version of the PCP with the version of the AP. It enables the user to check during system boot-up if both software versions are compatible. This ID will be visible for the AP in the status and control register (cf. [1]).
PDI revision number	This number indicates the version of the layout of the PDI in the IP-

	Core. For example if a register is added or removed the revision number is incremented. This ID will be visible for the AP in the status and control register (cf. [1]).
Number of RPDO buffers	One RPDO is always needed to receive data from the MN. The other two RPDO's can be used to read cross traffic from other CN's in the network. More RPDO's are currently not supported by the system.
RPDO buffer size	The size of all RPDO buffers. The packet buffer size calculation uses this value to compute the MAC internal RX buffer size. (16 bytes of header are added automatically)
Number of TPDO buffers	Only one TPDO buffer is available on the CN to transmit data to the MN.
TPDO buffer size	The size of the TPDO buffer can be entered here. The packet buffer size calculation uses this value to compute the MAC internal TX buffer size.
Asynchronous buffer 0 size	The size of the PDI asynchronous buffer 0. This buffer is used to transfer internal parameters between the PCP and the AP processor. (12 bytes of asynchronous message header are added automatically)
Asynchronous buffer 1 size	The size of the PDI asynchronous buffer 1. This buffer is used to transfer external messages like SDO's or Virtual Ethernet frames between the PCP and AP processor. (12 bytes of asynchronous message header are added automatically)
Enable LED output	This option enables the LED gadget in the PDI. This makes it possible to steer the POWERLINK LED's connected at the PCP from the AP side.
Use low-jitter sync IRQ with Soc timestamp forwarding	When the low jitter synchronization interrupt is enabled the Soc internal timestamp is used to generate the sync interrupt for the AP. This enables the synchronization of AP tasks to the global POWERLINK time. When this option is enabled the time fields in the PDI are filled with meaningful data. **WARNING:** If this option is turned off the real-time capability of POWERLINK is disabled.
Handle events in hardware	This option enables hardware support for event handling which provides a fast reaction on events at the AP side.

Table 6 Detailed description of all PDI IP-Core settings

3.4 Parallel interface settings

The parallel interface section of the POWERLINK IP-Core GUI provides parameters when the IP-Core mode is set to PDI with 8/16 bit parallel interface. Otherwise all options are disabled in this category. Table 7 provides a detailed description of all parameters in this section.

Parameter	Description
Parallel interface polarity	This parameter changes the polarity of the parallel interface control ports. It can either be high- or low active.
Data width of the parallel interface	The data width of the parallel interface can either be 8 or 16bits. The byte enable port is removed in case of 8 bits data width. Address pin zero can be omitted when using 16 bits data width.
Parallel interface endianness	This option gives the endianness of the AP processor. (In case of a Microblaze with PLB bus this option needs to be big endian. When

	the AP is a Microblaze with AXI little endian needs to be chosen.)

Table 7 Parallel interface configuration parameters

3.5 SPI interface settings

The SPI interface section of the POWERLINK IP-Core GUI provides parameters when the IP-Core mode is set to PDI with SPI interface. Otherwise all options are disabled in this category. Table 8 provides detailed information of all parameters.

Parameter	Description
SPI interface endianness	This option gives the endianness of the AP processor. (In case of a Microblaze with PLB bus this option needs to be big endian. When the AP is a Microblaze with AXI little endian needs to be chosen.)
CPHA mode	Enable the SPI CPHA mode
CPOL mode	Enable the SPI CPOL mode

Table 8 SPI interface parameters of the POWERLINK IP-Core

3.6 Direct IO interface settings

The Direct IO section of the POWERLINK IP-Core GUI provides parameters when the IP-Core mode is set to Direct IO only. Otherwise all options are disabled in this category. Table 9 provides detailed information of all parameters of this mode.

Parameter	Description
Number of RPDO buffers	One RPDO is always needed to receive data from the MN. The other two RPDO’s can be used to read cross traffic from other CN’s in the network.
Number of TPDO buffers	Only one TPDO buffer is available on the CN to transmit data to the MN.
Direct IO value length	Provides the number of cycles how long the output is valid and the output data is activated.

Table 9 Direct IO interface configuration parameters

3.7 IP-Core Misc tab

In the IP-Core miscellaneous (Misc) tab it is possible to enable or disable the MAC Direct Memory Access (DMA) observer. This observer can be used to check if the data fetch or store from the external memory has been successful. Note that the observer is not needed if the packets are stored internally.

The location of the MAC packet buffers can be changed in the openMAC IP-Core settings in chapter 3.2.

4 Software Interface

The POWERLINK IP-core is included into the Xilinx® Platform Studio (XPS). In order to provide configuration information to the PCP, a header file is generated which is called xparameters.h. In case of a POWERLINK IP-Core configuration with PDI the IP-Core also generates a library configuration file which is called cnApiCfg.h and consists of configuration data for the application processor (AP). The POWERLINK IP-core provides parameters depending on the settings done via the POWERLINK IP-Core GUI as listed in Table 10. Parameters for the AP configuration file are given in Table 11.

Name	Macro	Value
MAC buffer size	MAC_PKT_SIZE	Any integer (0 if PKTLOC = 2)
MAC RX buffers	MAC_RX_BUFFERS	1 … 16
Number of phys	PHY_COUNT	1 or 2
MAC packet buffer location	PACKET_LOCATION	0, 1 or 2
Time synchronization feature enable	PDI_GEN_TIME_SYNC	0 or 1 (false or true)
MAC DMA observer feature enable	OBSERVER_ENABLE	0 or 1 (false or true)
Number of RPDO buffers in PDI	NUM_RPDO	1, 2 or 3 (0 if CONFIG = 5)
Number of TPDO buffers in PDI	NUM_TPDO	1 (0 if CONFIG = 5)
Size of the RPDO in the PDI	RPDO_BUF_SIZE	1 … 1490 (only of C_IP_CORE_MODE is 4)
Size of the TPDO in the PDI	TPDO_BUF_SIZE	1 … 1490 (only of C_IP_CORE_MODE is 4)
The system ID of the PCP	PCP_SYS_ID	Constant value
The revision number of the PDI	PDI_REV	Constant value
Virtual Ethernet enable	VETH_ENABLE	0 = Disable; 1= Enable
Virtual Ethernet number of pending RX buffers.	VETH_RX_PENDING	1 … 6

Table 10 System description parameters

Name	Macro	Value
Mode of the IP-Core (C_IP_CORE_MODE)	CN_API_INT_[AXI ,PLB] CN_API_USING_SPI CN_API_USING_8BIT CN_API_USING_16BIT	None
AP Endianess	AP_IS_BIG_ENDIAN AP_IS_LITTLE_ENDIAN	none
Number of RPDO buffers in PDI	PCP_PDI_RPDO_CHANNELS	1, 2 or 3
Number of TPDO buffers in PDI	PCP_PDI_TPDO_CHANNELS	1
The revision number of the PDI	PCP_PDI_REVISION	Constant value
The system ID of the PCP	PCP_SYSTEM_ID	Constant value
Time synchronization feature enable	PCP_FPGA_TIMESYNCHW	0 = Disable; 1 = Enable
Maximum number of asynchronous buffers	PCP_PDI_ASYNC_BUF_MAX	1 or 2
Virtual Ethernet enable	VETH_DRV_ENABLE	0 = Disable; 1= Enable

Table 11 Auto generated AP configuration file (cnApiCfg.h)

5 FPGA Resource Utilization

The POWERLINK IP-core is used in combination with other components in an FPGA, hence the resource utilization and the circuit's performance (f_{max}) given in this section is an estimation only. The BRAM utilization is not given since it depends on several parameters defined in the GUI. Note that the parameters not noted in the table are set to default values (not changed in GUI).

PARAMETERS								FITTER Results				
IP CORE MODE	PACKET BUFFER LOCATION	USE RMII	USE 2ND PHY	NUMBER OF RPDO	LOW-JITTER SYNC	EVENT HW SUPPORT	PAP DATA WIDTH	FFs (POWERLINK[2])	LUTs (POWERLINK[2])	FFs (SYSTEM [3])	LUTs (SYSTEM [3])	fmax [MHz] [4]
Direct I/O	0	✓	✓	3	-	-	-	1102	1262	3860	4057	105
Parallel Interface	0	✓	✓	3	×	×	8	1372	1728	4400	4665	107
Parallel Interface	0	✓	✓	1	×	×	16	1348	1744	4376	4693	106
Parallel Interface	0	✓	✓	2	×	×	16	1360	1753	4388	4702	105
Parallel Interface	0	✓	✓	3	×	×	16	1382	1722	4410	4691	104
Parallel Interface	0	✓	✓	3	✓	×	16	1442	1817	4470	4773	105
Parallel Interface	0	✓	✓	3	✓	✓	16	1478	1864	4506	4860	106
SPI	0	✓	✓	3	×	×	-	1409	1782	4437	4739	106
PLB [5]	0	✓	✓	3	×	×	-	1475	1748	5520	6165	103
openMAC only	0	✓	✓	-	-	-	-	1058	1311	3815	4100	106
openMAC only	0	✓	×	-	-	-	-	995	1191	3752	3987	102
openMAC only	1	✓	✓	-	-	-	-	1374	1519	4193	4518	102

Table 12 Resource utilization on Spartan 6 FPGA (XC6SL16CSG324-2C) with the PLB IP-Core

Table note:

1. The Packet Buffer Location determines the location of the openMAC's packet buffers (0 = "TX and RX into BRAM" or 1 = "TX into BRAM and RX over AXI/PLB).
2. The fitter results are given for the POWERLINK IP-Core only.
3. The fitter results are given for the whole system design including the recommended component setup in XPS (e.g. Microblaze, AXI/PLB and memory controller).
4. The f_{max} values give the circuit's performance of the 100 MHz clock in slow corner.
5. The "AXI/PLB" mode refers to a second Microblaze CPU within the FPGA, which executes the application. Note that the application-specific logic is considered in the system's area utilization, as well as in the circuit's performance (f_{max}).
6. Unused parameters are identified with a hyphen ("-").

Copyright © B&R - Subject to change without notice

6 Files

The POWERLINK IP-Core for Xilinx FPGA is delivered with the necessary VHDL files and the XPS component description files (e.g: *_v2_1_0.mpd, *_v2_1_0.mdd, *_v2_1_0.mui and *_v2_1_0.tcl). In Table 13 the package directories structure of the POWERLINK IP-Core is described.

ⓘ It is essential that the global repository search path of the XPS is set to the IP-Core directory of your provided package or that the IP-Core is located in the **pcores** directory of the current project. The global repository search path can be set in the XPS under "**Edit -> Preferences -> Application -> Global Repository Search Path**"

Directory name	Description
doc	The doc-directory includes all IP-Core documentation files.
data	The data directory consists of the configuration files of the IP-Core. They are needed by the XPS to identify all IP-Core parameters.
hdl/vhdl hdl/vhdl/lib hdl/vhdl/openMAC_DMAmaster hdl/vhdl/ openMAC_DMAFifo_Xilinx	The VHDL-files are stored in the hdl/vhdl-directory.

Table 13 Directory description

All folders which are given in Table 13 are further described in detail in the following Table 14 to Table 15.

File name	Description
OpenMAC.pdf	The "openMAC & Components Documentation" introduces the IP-Cores openMAC, openFILTER and openHUB, as well as the software drivers (omethlib).
POWERLINK-IP-Core_Xilinx.pdf	This document.
POWERLINK-IP-Core_Generic.pdf	The "POWERLINK IP-Core Generic Documentation" provides a detailed description of the IP-Core.

Table 14 Documentation file description (doc)

File name	Description
lib/addr_decoder.vhd	Address decoder used in several components of the POWERLINK IP-Core
lib/ edgedet.vhd	Component to detect level changes of a signal
lib/memMap.vhd	Package file used in pdi.vhd
lib/req_ack.vhd	Used to generate wait- or acknowledge signals
lib/sync.vhd	Two-stage FF synchronizer
lib/slow2fastSync.vhd	Synchronizer to transfer pulse signals
openMAC_DMAFifo_Xilinx/ async_fifo_ctrl.vhd	Xilinx fifo control
openMAC_DMAFifo_Xilinx/fifo_read.vhd	Xilinx fifo read part
openMAC_DMAFifo_Xilinx/fifo_write.vhd	Xilinx fifo write part
openMAC_DMAFifo_Xilinx/n_synchronizer.vhd	Xilinx fifo two FF synchronizer
openMAC_DMAmaster/dma_handler.vhd	DMA handler sub-component
openMAC_DMAmaster/ipif_master_handler.vhd	Xilinx ipif for the DMA master
openMAC_DMAmaster/master_handler.vhd	DMA master handler sub-component
openMAC_Ethernet.vhd	MAC-layer top-level file
openFILTER.vhd	RMII filter that prevents distortions to propagate
openHUB.vhd	Ethernet hub

openMAC.vhd	Media Access Controller with dedicated hardware acceleration for POWERLINK
openMAC_16to32conv.vhd	Converter to access with a 32 bit CPU the openMAC register interface
openMAC_cmp.vhd	OpenMAC high-resolution timer component
openMAC_DMAFifo_Xilinx.vhd	Dual-clocked FIFO provided to the DMA master implementation
openMAC_DMAmaster.vhd	DMA master top-level file
openMAC_DPR_Xilinx.vhd	Dual Ported RAM description used in OpenMAC_*.vhdl files
openMAC_phyAct.vhd	Component to generate phy-activity LED signal
openMAC_PHYMI.vhd	Component to configure the phys via SMI
openMAC_rmii2mii.vhd	RMII to MII converter
pdi.vhd	Process Data Interface top-level file
pdi_apIrqGen.vhd	Sub-component to generate sync-interrupts to the AP
pdi_controlStatusReg.vhd	Sub-component implementing the Status/Control register
pdi_dpr_Xilinx.vhd	Dual Ported RAM description used in pdi.vhd
pdi_event.vhd	Sub-component providing hardware supported event handling
pdi_led.vhd	Sub-component providing LED interface
pdi_par.vhd	Asynchronous 8/16 bit parallel interface to AP
pdi_simpleReg.vhd	Sub-component implementing e.g. asynchronous buffer in PDI
pdi_spi.vhd	SPI interface to AP
spi.vhd	SPI slave implementation used by pdi_spi.vhd
spi_sreg.vhd	SPI shift register implementation used by spi.vhd
pdi_tripleVBufLogic.vhd	Sub-component used to generate triple-buffer management of PDOs
portio.vhd	Direct I/O top-level file
portio_cnt.vhd	Counter for direct I/O data valid signal generator
powerlink.vhd	POWERLINK IP-Core top-level file
plb_powerlink.vhd	PLB wrapper for file powerlink.vhd
axi_powerlink.vhd	AXI wrapper for file powerlink.vhd

Table 15 VHDL source files (src)

ⓘ The highlighted rows in the table refer to top-level files.

A Abbreviations

Abbreviation	Definition
AP	Application Processor
CN	Controlled Node
CPU	Central Processing Unit
DDR2	Double Data Rate type 2 SDRAM
DDR3	Double Data Rate type 3 SDRAM
DPRAM	Dual Ported RAM
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
IEEE	Institute of Electrical and Electronics Engineers
IP	Intellectual Property
IRQ	Interrupt Request
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LSB	Least Significant Bit
MCU	Microcontroller Unit
MII	Media Independent Interface
MSB	Most Significant Bit
PCB	Printed Circuit Board
PCP	POWERLINK Communication Processor
PDI	Process Data Interface
PDO	Process Data Object
PHY	Physical Transceiver
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RJ45	Registered Jack 45: Ethernet connector according to IEC 60603-7 (8P8C)
RMII	Reduced Media Independent Interface
SDRAM	Synchronous Dynamic RAM
SPI	Serial Peripheral Interface
SRAM	Static RAM

A References

[1] Bernecker+ Rainer Industrie-Elektronik GmbH, POWERLINK Slave for FPGAs - User's Manual Part II - DPRAM-based Host Interface Description, 2012.

[2] Ethernet POWERLINK Standardization Group, *Ethernet POWERLINK: Communication Profile Specification, DS301, V1.1.0.*

[3] Ethernet POWERLINK Standardization Group, „Ethernet POWERLINK XML Device Description DS311 V1.0.0," 2009. [Online]. Available: http://www.ethernet-powerlink.org/.

[4] Bernecker+ Rainer Industrie-Elektronik GmbH, POWERLINK IP-Core Generic Documentation, Eggelsberg, 2012.

[5] Bernecker+ Rainer Industrie-Elektronik GmbH, POWERLINK Slave for FPGAs - User's Manual Part I - Hardware Design and IP-Core Settings, 2012.

[6] Bernecker+ Rainer Industrie-Elektronik GmbH, POWERLINK Slave for FPGAs - User's Manual Part III - POWERLINK Application Programming, 2012.

[7] SYSTEC Electronic GmbH, openPOWERLINK: Ethernet POWERLINK Protocol Stack, Greiz, 2010.

[8] B&R, POWERLINK Slave for FPGAs - User's Manual Part II - DPRAM-based Host Interface Description, 2012.

B Figure index

Figure 1 POWERLINK IP-Core Block Diagram 5
Figure 2 Example XPS design with the AXI POWERLINK IP-Core 5
Figure 3 General settings tab of the axi_powerlink IP-Core 7
Figure 4 GUI of AXI POWERLINK IP-Core with example configuration of the openMAC 8
Figure 5 Process data interface (PDI) configuration section of the axi_powerlink IP-Core 10

C Table index

Table 1 Versions 2
Table 2 Safety notices 2
Table 3 Modes of the POWERLINK IP-Core....................8
Table 4 openMAC settings description....................9
Table 5 Location of the MAC packet buffer 10
Table 6 Detailed description of all PDI IP-Core settings 11
Table 7 Parallel interface configuration parameters....................12
Table 8 SPI interface parameters of the POWERLINK IP-Core 12
Table 9 Direct IO interface configuration parameters 12
Table 10 System description parameters....................13
Table 11 Auto generated AP configuration file (cnApiCfg.h)....................13
Table 12 Resource utilization on Spartan 6 FPGA (XC6SL16CSG324-2C) with the PLB IP-Core 14
Table 13 Directory description....................15
Table 14 Documentation file description (doc) 15
Table 15 VHDL source files (src)....................16

Index

A

Abbreviations 17

F

Figure index 19

G

General information 2

I

Index 21
Introduction 4

L

License..2

M

Manual history... 2

R

References.. 18

S

Safety notices 2

T

Table index 20
Table of Contents.. 3