Perfection in Automation
www.br-automation.com

POWERLINK IP-Core
Altera FPGA Documentation

Model No: OAT113110.10
Version: 1.0

Date: 2013-02-12

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:

II Table of Contents

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

1 Introduction 4

2 System-On-a-Programmable-Chip (SOPC) Integration 5

3 POWERLINK IP-Core GUI 7
3.1 General IP-Core settings 7
3.2 openMAC IP-Core settings 8
3.2.1 openMAC DMA observer 9
3.3 PDI settings 9
3.4 Parallel interface settings 11
3.5 SPI interface settings 11
3.6 Direct IO interface settings 12
3.7 openMAC only settings 12

4 Timing Consideration 14

5 Software Interfacing (system.h) 15

6 FPGA Resource Utilization 16

7 Files 17

A Abbreviations 19

B References 20

C Figure index 21

D Table index 22

Index 23

1 Introduction

The POWERLINK IP-Core is provided as ready-to-use in Altera Quartus II environment with SOPC version 10.1 SP1. This documentation presents the integration of the IP-Core into the SOPC environment, 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*" (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 Cyclone III or IV 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), which can be started by using the Altera SOPC builder. This enables the user to define several parameters for the POWERLINK node, like the type of API (e.g. SPI or parallel interface), the number of process data objects (PDO) or enabling the 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 SOPC GUI.

2 System-On-a-Programmable-Chip (SOPC) Integration

The POWERLINK IP-Core's "Internal Bus Interface" complies with the Altera® Avalon Interface Specification [1], hence there is no extra VHDL wrapper file necessary for conversion purpose. The IP-Core is fully compatible to the Avalon Memory Mapped interface, hence it can be used in combination with e.g. the Altera Nios II® processor.

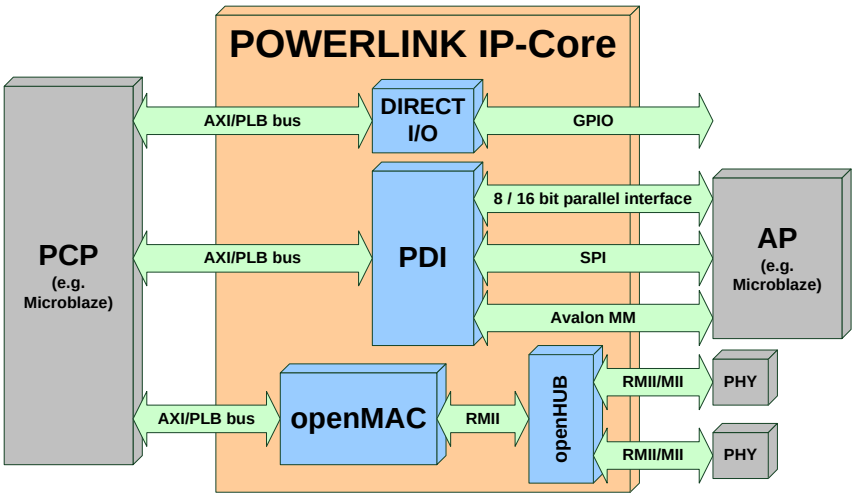

Figure 1 POWERLINK IP-Core Block Diagram

The SOPC 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 SOPC configuration. However, the MAC-layer components openMAC and openHUB[1] are mandatory components of a POWERLINK node.

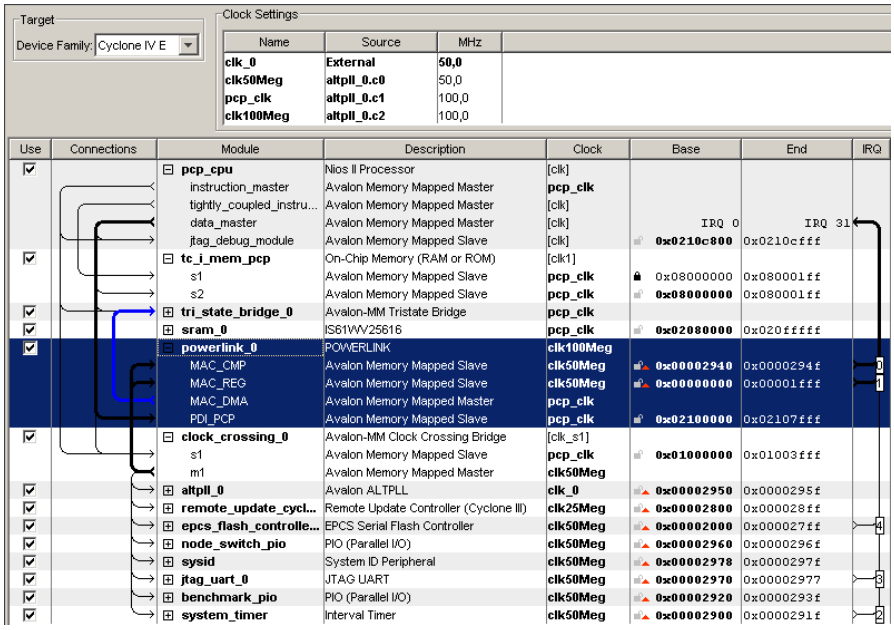

Figure 2 Example SOPC with POWERLINK IP-Core

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

In Figure 2 an SOPC design is shown, which includes the PCP as a Nios II/s, SRAM memory controller and other components available in the SOPC. Depending on the configurations of the POWERLINK IP-Core the Avalon Memory Mapped Slave/Master interfaces may vary, however, MAC_REG and MAC_CMP are mandatory in any case. The following interfaces may be visible when the POWERLINK IP-Core is inserted in the SOPC builder:

- **MAC_DMA**: Master interface for transferring RX/TX packets to external memory.
- **MAC_REG**: Slave interface for accessing the MAC internal registers. (always mandatory)
- **MAC_CMP**: Slave interface for accessing the MAC internal timer registers. (always mandatory)
- **MAC_PKT**: Slave interface for accessing the RX/TX packets in internal M9K blocks.
- **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 3 shows the GUI for the POWERLINK IP-Core configuration. The parameters are separated into sub-groups e.g. “General Settings” or “Process Data Interface Settings”. The user has to set the intended POWERLINK Slave Design Configuration, which affects the subsequent parameters’ visibility e.g. in case of “openMAC only” PDI settings are not visible because they are not meaningful.

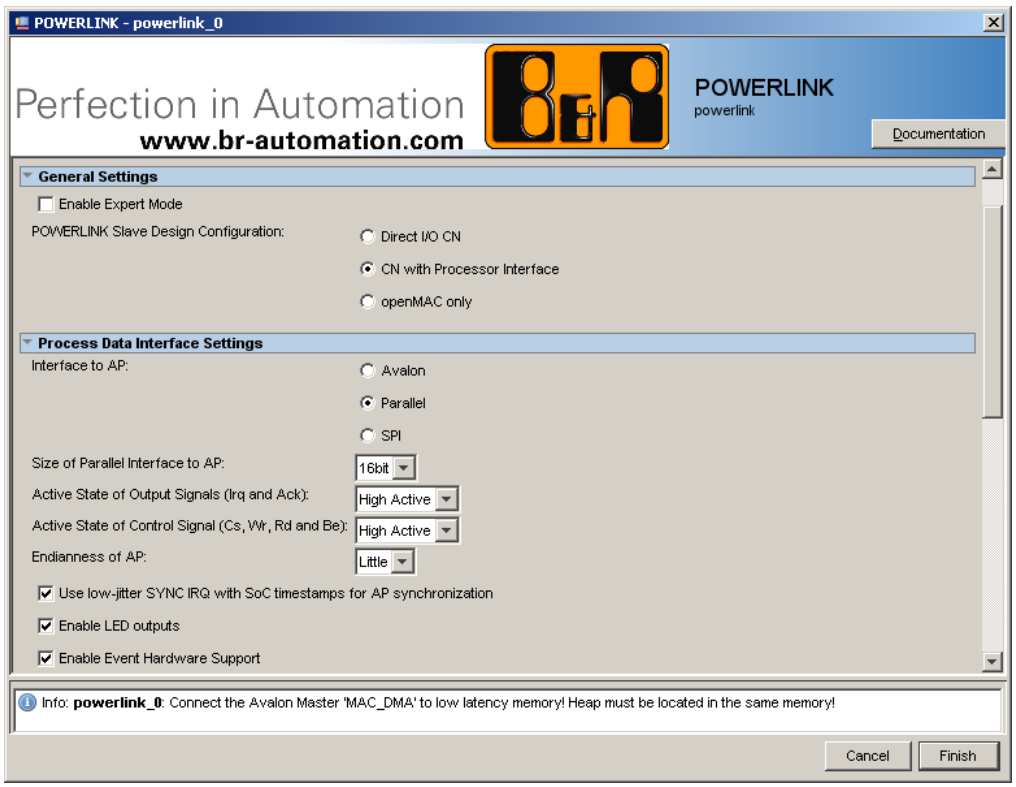

Figure 3 GUI of POWERLINK IP-Core with example configuration

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 e.g. “Info: Powerlink_0: Consider to use RMII to reduce resource usage!”.

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 5).

3.1 General IP-Core settings

With the “General Settings” IP-core GUI section (see Figure 4) 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 SOPC builder.

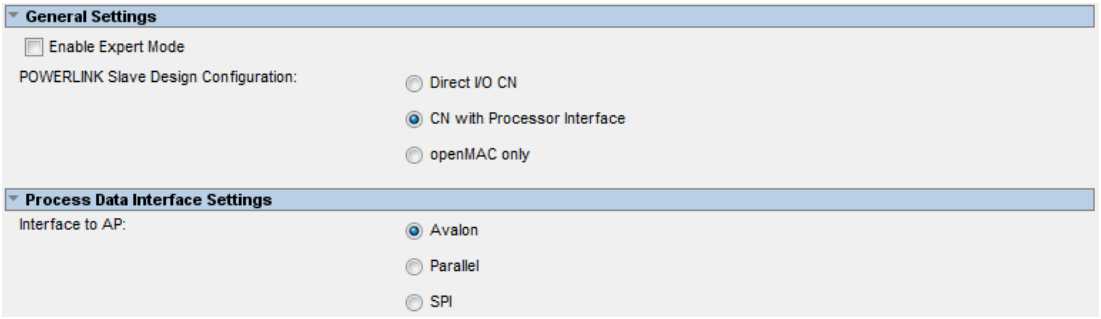

Figure 4 General settings fields of the POWERLINK IP-Core

The general settings of the GUI provide an option to enable the expert mode in the IP-Core GUI. When this mode is activated additional configuration settings are provided which should only be used very carefully. The IP-core version will be displayed in “General Settings” when activating the “expert mode”.

Table 3 provides a detailed description of the available POWERLINK IP-Core design configuration modes.

Mode	Description
Direct IO CN	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 is available)
CN with Processor Interface	This mode provides an additional interface for a second processor. The Process Data Interface (PDI) can have different interfaces to the second application processor (AP) and is described further in Table 4.
openMAC only	The openMAC mode only instantiates the MAC layer and provides no additional user interface.

Table 3 Modes of the POWERLINK IP-Core

Process Data Interface	Description
Parallel Interface (8/16bit)	This interface provides a parallel bus with the choice of two different data bus widths: 8 or 16bits.
SPI Interface	This interface consists of a serial bus with four wires. (Clk, MISO, MOSI, SS)
Avalon Interface	This interface provides an FPGA internal connection to the second processor by using the Avalon Memory Mapped Slave interface.

Table 4 Process Data Interface (PDI) description

3.2 openMAC IP-Core settings

By using the POWERLINK IP-Core the openMAC component, is always instantiated. This component handles the reception and transmission of POWERLINK frames. Figure 5 pictures all parameters which can be applied to the POWERLINK MAC layer.

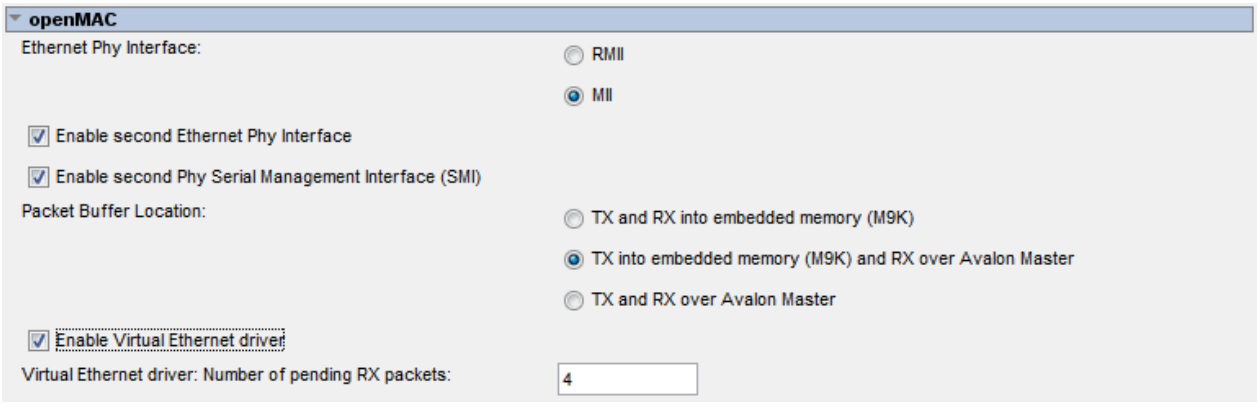

Figure 5 openMAC configuration settings of the POWERLINK IP-Core

Table 5 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 SOPC builder.

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 M9K blocks or to forward them by the local Avalon interconnect to an external memory. (Probably SRAM) The following Table 6 provides more information about this parameter.
Ethernet Phy 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 M9K blocks). If RMII is selected, it an additional 100Mhz clock needs to be assigned to the IP-core (see Figure 6).
Enable second Ethernet Phy Interface	If this parameter is enabled the openHUB component is instantiated which enables additional ports for connecting a second PHY.
Enable second Phy Serial Management Interface (SMI)	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)

Table 5 openMAC settings description

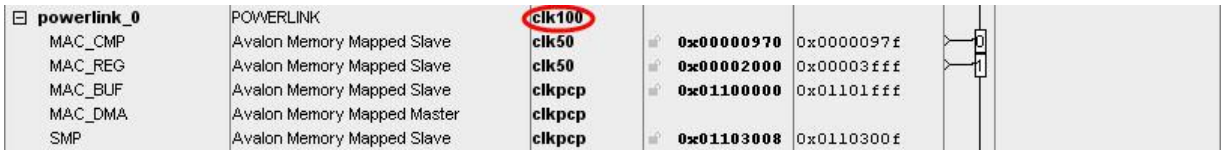

Figure 6 Additional clock setting for RMII

Packet buffer location setting	Description
TX and RX into embedded memory (M9K)	If this option is active RX and TX packets are stored in internal M9K blocks. The packet buffer can be accessed by using the bus inter-

	face MAC_PKT. This option should be preferred if enough M9K blocks are available.
TX into embedded memory (M9K) and RX over Avalon	This option stores the TX packets in internal M9K blocks 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 burst size of the MAC_DMA can be adjusted by using the "*Number of words per DMA write transfer*" option when the "*Expert mode*" is enabled.
TX and RX over Avalon	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 M9k block is used for the packet buffer. Only block RAM 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 is 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)

Table 6 Location of the MAC packet buffer

3.2.1 openMAC DMA observer

If the "Expert Mode" of the IP-Core is enabled the MAC Direct Memory Access (DMA) observer is visible. 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.

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 hidden. Figure 7 pictures all available options in this category. Some of them can be deactivated in order to save resources.

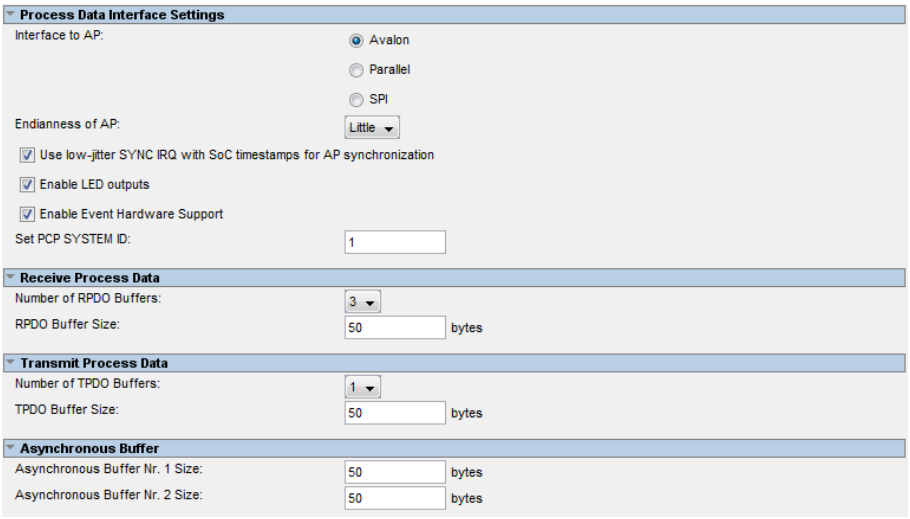

Figure 7 Process data interface (PDI) configuration section of the POWERLINK IP-Core

Detailed description of all parameters is given in Table 7. By changing these parameters the internal memory layout of the PDI can change and more internal M9K blocks are 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. [2]).
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 (Enable the expert mode to see this option!). This ID will be visible for the AP in the status and control register (cf. [2]).
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 a 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 1 size	The size of the PDI asynchronous buffer 1. 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 2 size	The size of the PDI asynchronous buffer 2. 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 control the POWERLINK LED's connected to 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. (Enable the expert mode to see this option! Per default it is activated.)

Table 7 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 8 provides a detailed description of all parameters in this section.

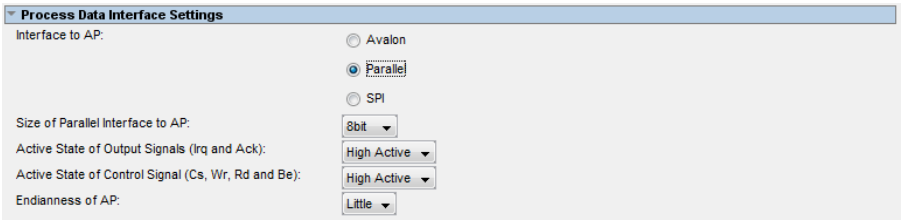

Figure 8 Parallel interface POWERLINK IP-Core GUI settings

Parameter	Description
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, however the Be (Byte enable) signals have to be used.
Active state of output signals	This option sets the interrupt- and the acknowledge signal polarity. It can either be high- or low active.
Active state of control signals	This option sets the polarity of the CS, Wr, Rd and Be signals of the parallel interface. It can either be high- or low active.
Parallel interface endianness	This option gives the endianness of the AP processor. (In case of a Nios® II processor little endian needs to be chosen.)

Table 8 Parallel interface configuration parameters

3.5 SPI interface settings

The SPI interface section of the POWERLINK IP-Core GUI provides parameters if the IP-Core mode is set to PDI with SPI interface. Otherwise all options are disabled in this category. Table 9 provides detailed information of all parameters.

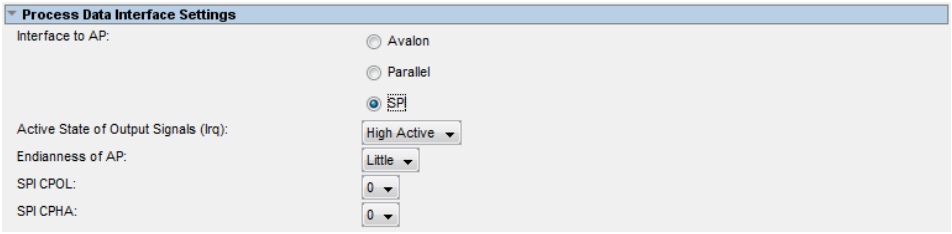

Figure 9 SPI interface POWERLINK IP-Core GUI settings

Parameter	Description
Active state of output signals	This option sets the interrupt- and the acknowledge signal polarity. It can either be high- or low active.
SPI interface endianness	This option gives the endianness of the AP processor. (In case of a Nios® II processor this option needs to be little endian)
CPHA mode	Enable the SPI CPHA mode
CPOL mode	Enable the SPI CPOL mode

Table 9 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 10 provides detailed information of all parameters of this mode.

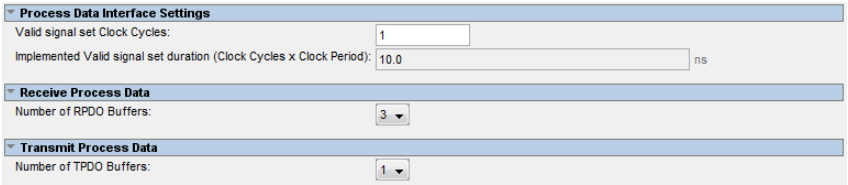

Figure 10 Direct IO interface POWERLINK IP-Core GUI settings

Parameter	Description
Valid signal set clock cycles	Provides the number of cycles how long the output is valid and the output data is activated.
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.

Table 10 Direct IO interface configuration parameters

3.7 openMAC only settings

If the IP-Core GUI is set to openMAC only mode no PDI or Direct IO module is present in the POWERLINK IP-Core. Therefore the internal packet buffer size can't be determined by the GUI and needs to be entered manually. The additional options available in this mode are given in the following Table 11.

Parameter	Description
Number of RX buffers	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	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

Table 11 openMAC only mode settings

4 Timing Consideration

The POWERLINK IP-Core and the mandatory components (e.g. PCP and memory controller) require further configuration if implemented in an FPGA. In order to work properly timing constraints have to be applied to the design, which is done by SDC-files directing the Altera TimeQuest timing analyzer tool. The files are delivered with the POWERLINK IP-Core (refer to Table 17) and have to be forwarded to TimeQuest (in Quartus II: Assignments – Settings – TimeQuest – “SDC files to include in project”). The timing constraints are considered by Quartus II during place & route, which ensures an optimized fitter result within short compilation times.

The delivered SDC-files include the configuration of the POWERLINK IP-Core (e.g. PHY interface and parallel interface) and the SRAM interface. If the user adds additional components to the SOPC (e.g. SDRAM or GPIO), timing constraints have to be added manually.

It is important that the design’s top level pin- (and signal-) names match those defined in the SDC files (e.g. **EXT_CLK** or **PHY0_TXEN**). Otherwise *TimeQuest* is not able to apply constraints properly!

Further information about TimeQuest and basics on timing constraints are provided by Altera (www.altera.com).

5 Software Interfacing (system.h)

The POWERLINK IP-core is included into an SOPC (Altera) or any other processor environment. In order to provide configuration information to the PCP, a header file is generated (system.h). The POWERLINK IP-core provides parameters depending on the settings done via the SOPC GUI as listed in Table 12. Note that several parameters are described more detailed in Table 13.

Name	Macro	Value
MAC buffer size	MACBUFSIZE	Any integer (0 if PKTLOC = 2)
MAC RX buffer size	MACRXBUFSIZE	Any integer (0 if PKTLOC = 1 or 2)
MAC RX buffers	MACRXBUFFERS	1 … 16
MAC TX buffer size	MACTXBUFSIZE	Any integer (0 if PKTLOC = 2)
MAC TX buffers	MACTXBUFFERS	Any integer (0 if CONFIG = 5)
Number of MAC CMP timer	CMPTIMERCNT	1 or 2 (1 if CONFIG = 0 or 5)
Number of PDI RPDOs	PDIRPDOS	1, 2 or 3 (0 if CONFIG = 5)
Number of PDI TPDOs	PDITPDOS	1 (0 if CONFIG = 5)
POWERLINK IP-core configuration	CONFIG	0, 1, … , 5
AP interface endianness	CONFIGAPENDIAN	0 or 1
Number of phys	PHYCNT	1 or 2
MAC packet buffer location	PKTLOC	0, 1 or 2
Time synchronization feature enable	TIMESYNC	0 or 1 (false or true)
LED feature enable	LEDGADGET	0 or 1 (false or true)
MAC DMA observer feature enable	DMAOBSERVER	0 or 1 (false or true)

Table 12 System description parameters

Macro	Value	Description
CONFIG	0	Direct I/O (no AP intended)
	1	8bit parallel interface
	2	16bit parallel interface
	3	SPI
	4	Avalon bus
	5	openMAC only (no PDI available)
CONFIGAPENDIAN	0	Little endian
	1	Big endian
PKTLOC	0	TX and RX packets are stored in MAC-internal DPRAM
	1	TX packets are stored in MAC-internal DPRAM, RX buffers are located in system's heap
	2	TX and RX packets are stored in system's heap

Table 13 System description parameters, details

6 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 M9K 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	LEs (POWERLINK[2])	LEs (SYSTEM [3])	fmax [MHz] [4]
Direct I/O	2	×	✓	3	-	-	-	2006	4988	105
Parallel Interface	2	×	✓	3	×	×	8	2351	5388	105
Parallel Interface	2	×	✓	1	×	×	16	2313	5306	105
Parallel Interface	2	×	✓	2	×	×	16	2392	5399	105
Parallel Interface	2	×	✓	3	×	×	16	2448	5464	105
Parallel Interface	2	×	✓	3	✓	×	16	2547	5574	105
Parallel Interface	2	×	✓	3	✓	✓	16	2572	5604	104
SPI	2	×	✓	3	×	×	-	2432	5460	106
Avalon	2	×	✓	3	×	×	-	2307	5429 [5]	105
openMAC only	0	×	✓	-	-	-	-			
openMAC only	1	×	✓	-	-	-	-			
openMAC only	2	×	✓	-	-	-	-			
openMAC only	2	✓	✓	-	-	-	-			

Table 14 Resource utilization on Cyclone 4 FPGA (EP4CE115F29C7)

Table note:

1. The Packet Buffer Location determines the location of the openMAC's packet buffers (0 = "TX and RX into embedded memory (M9K)", 1 = "TX into embedded memory (M9K) and RX over Avalon" or 2 = "TX and RX over Avalon").
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 SOPC (e.g. Nios II/s, clock-crossing bridge and memory controller).
4. The f_{max} values give the circuit's performance of the 100 MHz clock in slow corner (85°C).
5. The resource utilization does not include any application-specific components (e.g. second Nios II).
6. Unused parameters are identified with a hyphen ("-")

7 Files

The POWERLINK IP-Core for Altera FPGA is delivered with the necessary VHDL files and the SOPC component description (*_hw.tcl). In Table 15 to Table 18 the package directories are described.

It is essential that the package is stored in a subdirectory of your Quartus project. Otherwise SOPC cannot recognize the package, and in addition the mif-directory is created incorrectly! Consider to apply the following path example: **$QUARTUS_PRJ/POWERLINK**
Alternatively, *.IPX files can be used to reference the POWERLINK IP-core directory relatively from the Quartus project path (tutorials on www.altera.com).

Directory name	Description
doc	The doc-directory includes all IP-Core documentation files.
img	In the img-directory images for the GUI in SOPC are stored.
sdc	The sdc-directory provides different timing-constraints file usable for TimeQuest.
src src/lib src/openMAC_DMAmaster	The VHDL-files are stored in the src-directory.
../mif	The mif-directory is created when inserting the POWERLINK IP-core successfully into SOPC. It provides memory initialization files necessary for the IP-core during compilation of Quartus II.

Table 15 Directory description

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_Altera.pdf	This document.
POWERLINK-IP-Core_Generic.pdf	The “POWERLINK IP-Core Generic Documentation” provides a detailed description of the IP-Core.

Table 16 Documentation file description (doc)

File name	Description
PLK_MII_base.sdc	Defines the timing constraints for MII phys.
PLK_RMII_base.sdc	Defines the timing constraints for RMII phys.
PLK_RMII_base_PARPDI.sdc	Defines the timing constraints for RMII phys and a parallel asynchronous 8/16 bit interface (to the AP).

Table 17 Timing Constraints file description (sdc)

File name	Description
lib/addr_decoder.vhd	Address decoder used in several components of the POWERLINK IP-Core
lib/edge_det.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_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_Altera.vhd	Dual-clocked FIFO provided to the DMA master implementation
openMAC_DMAmaster.vhd	DMA master top-level file
openMAC_DMAmaster/dma_handler.vhd	DMA handler sub-component
openMAC_DMAmaster/master_handler.vhd	DMA master handler sub-component
openMAC_DPR_Altera.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_Altera.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

Table 18 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

B References

[1] Altera, „Avalon Interface Specification,“ Version 11.0 May 2011. [Online]. Available: http://www.altera.com/literature/manual/mnl_avalon_spec.pdf.

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

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

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

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

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

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

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

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

C Figure index

Figure 1 POWERLINK IP-Core Block Diagram 5
Figure 2 Example SOPC with POWERLINK IP-Core 5
Figure 3 GUI of POWERLINK IP-Core with example configuration 6
Figure 4 General settings fields of the POWERLINK IP-Core 7
Figure 5 openMAC configuration settings of the POWERLINK IP-Core 8
Figure 6 Additional clock setting for RMII 8
Figure 7 Process data interface (PDI) configuration section of the POWERLINK IP-Core 10
Figure 8 Parallel interface POWERLINK IP-Core GUI settings 11
Figure 9 SPI interface POWERLINK IP-Core GUI settings 12
Figure 10 Direct IO interface POWERLINK IP-Core GUI settings 12

D Table index

Table 1 Versions....2
Table 2 Safety notices....2
Table 3 Modes of the POWERLINK IP-Core....7
Table 4 Process Data Interface (PDI) description....7
Table 5 openMAC settings description....8
Table 6 Location of the MAC packet buffer....9
Table 7 Detailed description of all PDI IP-Core settings....11
Table 8 Parallel interface configuration parameters....11
Table 9 SPI interface parameters of the POWERLINK IP-Core....12
Table 10 Direct IO interface configuration parameters....12
Table 11 openMAC only mode settings....13
Table 12 System description parameters....15
Table 13 System description parameters, details....15
Table 14 Resource utilization on Cyclone 4 FPGA (EP4CE115F29C7)....16
Table 15 Directory description....17
Table 16 Documentation file description (doc)....17
Table 17 Timing Constraints file description (sdc)....17
Table 18 VHDL source files (src)....18

Index

A

Abbreviations 19

C

C Macro 15

F

Figure index 21

G

General information 2

I

Index 23
Introduction 4

L

License 2

M

Manual history 2

R

References 20

S

Safety notices 2
System description 15

T

Table index 22
Table of Contents 3