Standard Products

UT100SpW02 SpaceWire Protocol Handler IP for RadTol Eclipse FPGA

Preliminary Data Sheet

October 3, 2008
www.aeroflex.com/SpaceWire

FEATURES

- Designed for use with the RadTol Eclipse FPGA (view datasheet at www.aeroflex.com/RadTolFPGA)
- Dual ECSS-E-50-12A compliant links
- Data rates from 2 to 100 Mbits/sec
- 9 bit transmit and receive FIFO user interface

INTRODUCTION

Aeroflex Colorado Springs' UT100SpW02 SpaceWire Dual Link Protocol Handler is designed to manage the SpaceWire protocol as defined in ECSS-E-50-12A. (For a copy of the standard, please visit www.estec.esa.nl. The primary user interfaces to the device are 9 by 128 receive and transmit FIFO's. The FIFO data is organized into 8-bits of width while the other bit is used to indicate an End of Packet (EOP) or an End of Error Packet (EEP). The device is designed to interface with the UT100SpWPHY01 Physical Layer Chip.A time code feature is included in the standard and the bits are defined as six for the time code and two bits for control flags. Two more signals are included as part of the time code and they are TICK IN and TICK OUT. These signals will either trigger the sending of a time code or indicate a time code has been received.

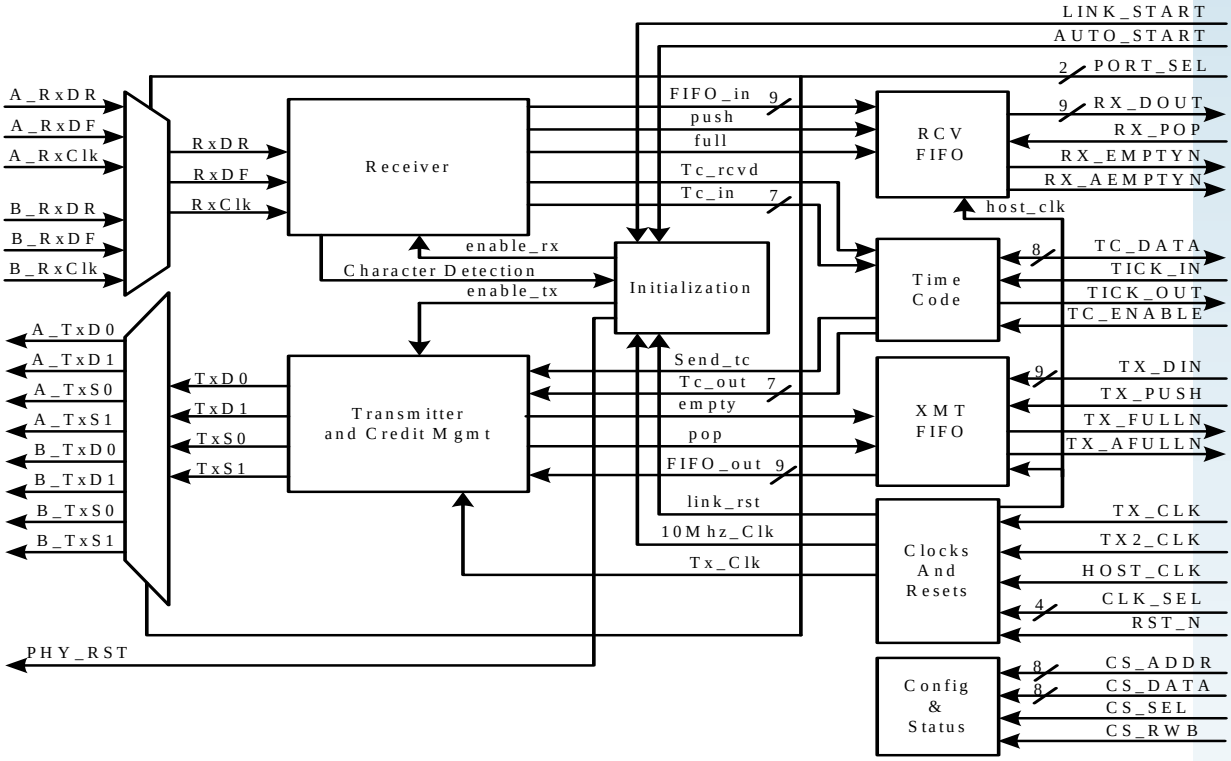


Figure 1. UT100SpW02 SpaceWire Protocol Block Diagram

Application Information

Utilization

Device	Cells	% Utilization
UT6325	435 of 1536	35

UT6325 Clock Utilization

Dedicated Clk	Global Clks	Quad Clks
0	3	5

FIFO's

The UT100SpW02 uses two synchronous FIFO's for transmitting and receiving data. The FIFO bus width is 9 bits with the 9th bit used for EOP and EEP handling. Table 2 below defines EOP and EEP handling in the FIFO's. The FIFo's are implemented using two of the UT6325 RadTol memory blocks on the UT6325 RadTol Eclipse FPGA.

Table 1. FIFO EOP and EEP Definitions

Bit 8	Bits 7 to 0	Character
1	0000_0000	EOP
1	0000_0001	EEP

RX_FIFO

The RX_FIFO provides an interface between the captured SpaceWire data and the local host. The FIFO size is 9 bits wide by 128 bytes deep. Data is written into the receive FIFO using the recovered clock (Rx_Clk). The FIFO is completely empty when RX_EMPTYN is low and has 1/4 entries left when RX_AEMPTYN is low.

TX_FIFO

The TX_FIFO is identical in depth and width as the RX_FIFO. Data is written into the FIFO using the HST_CLK. TX_EMPTYN and TX_AFULLN are updated for use in the host_clk domain. The FIFO is 3/4 full when TX_AFULLN is low. The TX_EMPTYN Flag can be used to determine when a complete packet has been read by the protocol handler. Refer to Figure

Time Codes

Time codes are handled as defined in ECSS-E-50-12A. When EN_TME_CODE is set high, the 8 bits of time code data are inputs and TICK_IN is enabled. This condition is referred to in the SpaceWire standard as Time Master. When TC_ENABLE is low then TICK_OUT will toggle when a Time Code is received and the Time Code data will be placed on the TC_Data bus.

Enabling the Link

SpaceWire is a self managing protocol that uses a series of control characters as a handshake mechanism to manage the flow of data between two nodes. Enabling the protocol handler requires por_n be set high and since the interface is to the UT200SpW01 SpaceWire Transceiver, the signal phy_rstn must be set high as well. The signal link_disable must also be set low and the protocol handler will then attempt to establish a link using the protocol defined the SpaceWire Standard.

Clocks

The Protocol Handler has three clock inputs. The first is the Transmit Clock (Tx_Clk). The second is the Transmit divided by 2 clock (Tx2_Clk). The Tx2_Clk must be 1/2 of the Tx_Clk. The third clock is the Host Clock (Hst_Clk). The Hst_Clk is used to read and write to the FIFO's. For convenience the Tx2_Clk and the Hst_Clk can be the same frequency.

Port Select

Two bits are used to select which port is active. There are options for forcing Port A, forcing Port B and an autodetect feature that will select the first active port detected. Table 3 below shows the options for these input signals.

Table 2. FIFO EOP and EEP Definitions

Port Sel [1:0]	Function
00	Disable Both Links
01	Force Port A
10	Force Port B
11	Autodetect

SpaceWire Physical Interface

The UT100SpW02 is designed to interface with the Aeroflex UT200SpWPHY01 physical layer device. Since each device constitutes a single link, two of the UT100SpWPHY01 chips are required to implement Port A and Port B on the protocol handler. Transmit data and strobe are written to the Transceiver Chip in bit pairs using the Wr_Clk signal. Conversely, received data is written into the Protocol Handler in bit pairs on the rising edge of the recovered Rx_Clk. The requirements for the Wr_Clk and the Tx_Clk are they must be phase stable and the Wr_Clk must be 1/2 the frequency of the Tx_Clk. Refer to Figure 7below.

Status Registers

The UT100SpW02 has an 8 bit output port for and status registers. There are 4 Status registers on the device. The following tables define all of these registers. Status registers are readable and can be cleared by writing FF into CS_ADDR = 100

Status Register 0 CS_ADDR = 000

Table 3. Status Register 0 Bit Descriptions

R/W	Bit Position	Description	Default Binary
Read/ Clear	7	Link reached Run State	0
Read/ Clear	6	An error occurred in link Run State	0
Read/ Clear	5	The Transmit FIFO Overflowed	0
Read/ Clear	4	The Receive FIFO Overflowed	0
Read/ Clear	3	A parity error was detected	0
Read/ Clear	2	A transmit Credit Error was detected	0
Read/ Clear	1	A receive Credit Error was detected	0
Read/ Clear	0	A Disconnect Error occurred	0

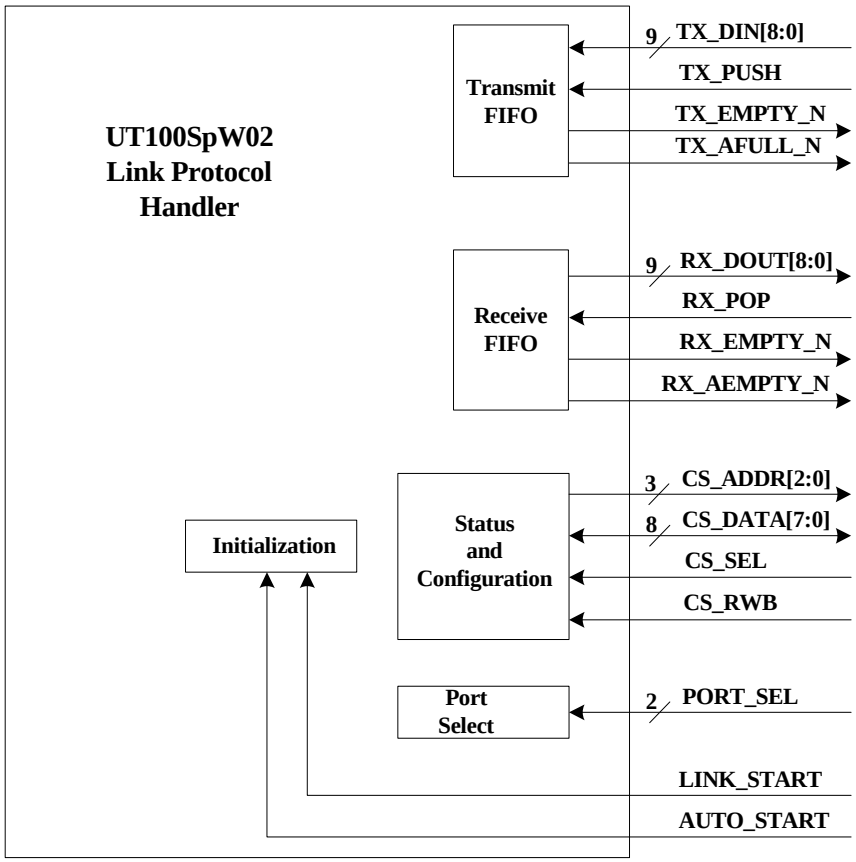


Figure 2. User Interface

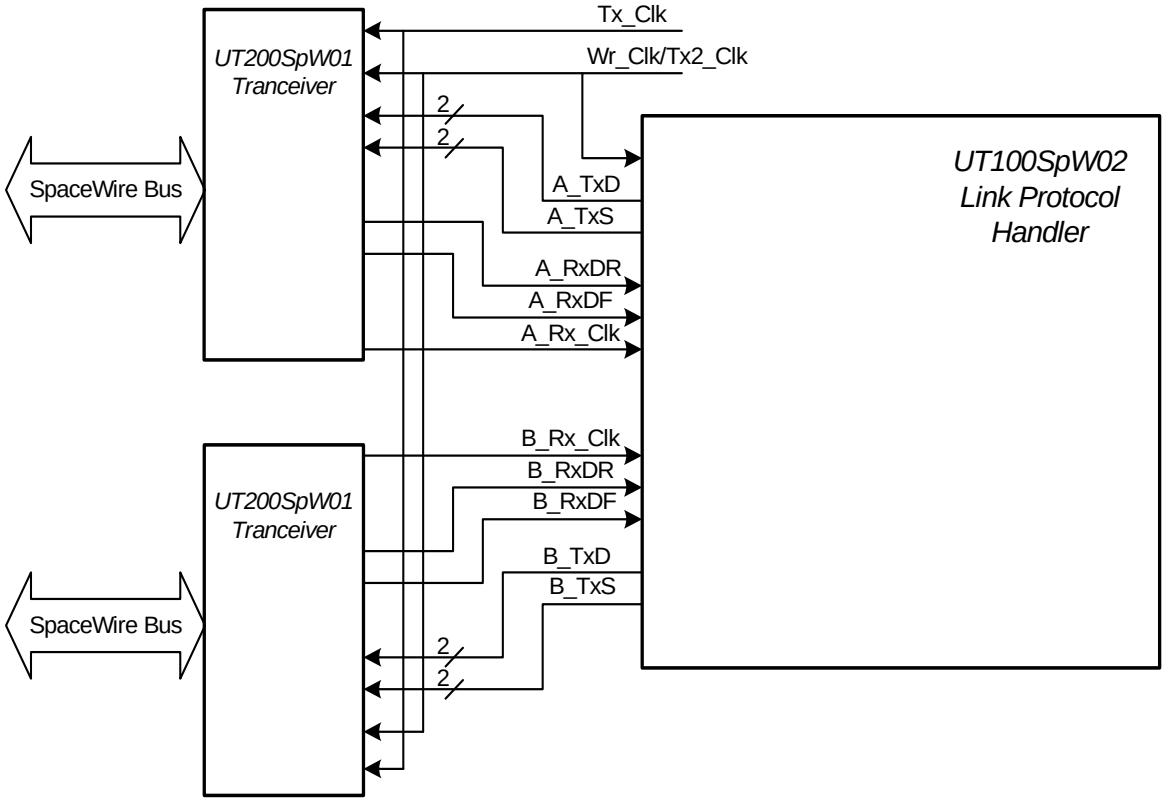


Figure 3. Physical Layer Transceiver Interface

Status Register 1 CS_ADDR = 001

Table 4. Status Register 1 Bit Descriptions

R/W	Bit Position	Description	Default Binary
Read/ Clear	7	An error occurred during link initialization	0
Read/ Clear	6	A disconnect error occurred during link initialization	0
Read/ Clear	5	An unexpected Time Code value was received	0
Read/ Clear	4	An EEP was received on the SpaceWire bus	0
Read/ Clear	3	Time Code Received	0
Read/ Clear	2	Sent First FCT	0
Read/ Clear	1	Transmit Credit Available	0
Read/ Clear	0	Null Received	0

Status Register 2 CS_ADDR = 002

Table 5. Status Register 2 Bit Description

R/W	Bit Position	Description	Default Binary
Read/ Clear	7 to 0	Current Time Code Value	0000_0000

Part Number CS_ADDR = 011
The user can read from this address to get the part number of the SpaceWire LPH.

Clear Status Bits CS_ADDR = 100
The user can toggle the CS_RWB signal while driving the above address (Do not drive data since data is an output) to clear all of the status bits in all status registers.

Clock Select Pins
The 5bit register CLK_SEL[4:0] is used to set the 10MHz Clock based on the Tx_Clk input. Below is a table showing some example Tx_Clk inputs and the correct CLK_SEL setting.

Tx_CLK (MHz)	Clock Divider	CLK_SEL (BIN)
100	10	1010
90	9	1001
80	8	1000
70	7	0111

Autostart
Can be set to request the link to start automatically on receipt of a NULL. Autostart actually means the link waits for activity from the other one before starting. Therefore, a link will never start if both ends of the link are in Autostart.

Link Start
Can be set to start a link, i.e. to cause the transition from the Ready state to the Started state.

Table 6. UT100SpW02 Core Signal Description

Name	I/O	Description
a_ser21ph_data[1:0]	I	Port A rising edge data in (bit 0)
arx_clk	I	Port A recovered clock
b_ser21ph_data[1:0]	I	Port B rising edge data in (bit 0)
brx_clk	I	Port B recovered clock
a_lph2ser_data[1:0]	O	Port A transmit bit 0 data
a_lph2ser_strb[1:0]	O	Port A transmit bit 0 strobe
b_lph2ser_data[1:0]	O	Port B transmit bit 0 data
a_lph2ser_strb[1:0]	O	Port B transmit bit 0 strobe
rx_dout[8:0]	O	Receive FIFO Data out bus (9 bits)
rx_emptyn	O	Receive FIFO empty flag
rx_aemptyn	O	Receive FIFO almost empty flag
rx_pop	I	Receive FIFO pop
tx_din[8:0]	I	Transmit data bus (9 bits)
tx_push	I	Transmit push
tx_empty_n	O	Transmit empty flag
tx_afull_n	O	Transmit almost full flag
host_clk	I	Host clock (internally used as the FIFO clock)
tx_clk	I	Transmit Clock
tx2_clk	I	Transmit Clock divided by 2
tx_clk_sel[4:0]	I	Clock Select (used to set the 10 Mhz clock for initialization)
spar_err	O	Parity Error flag
linka	O	Link Run Port A
linkb	O	Link Run Port B
tick_out	O	Tick Out
tick_in	I	Tick In
port_sel[1:0]	I	Port Select
tcc_extern[1:0]	I	External Time Code Flags
tc_extern[5:0]	I	External Time Code Data
tc_load	I	Load External Time Data
tcc_load	I	Load External Code Flags
cs_addr[2:0]	I	Configuration and Status Address
cs_data[7:0]	O	Configuration and Status Data

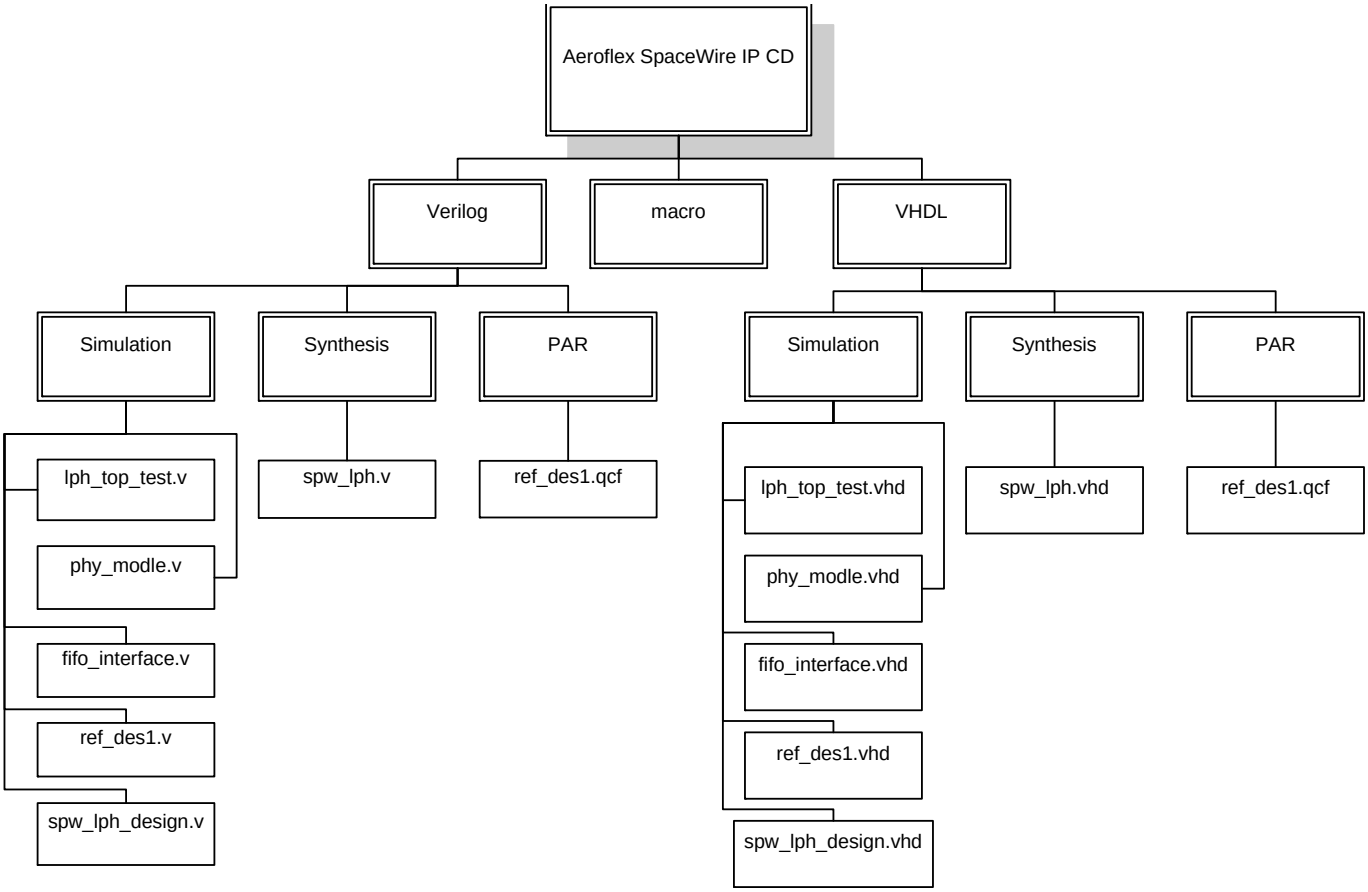
Table 6. UT100SpW02 Core Signal Description

Name	I/O	Description
cs_sel	I	Configuration and Status Select
cs_rwb	I	Configuration and Status Read/Write
host_rst_n	I	Reset status registers and if enabled, resets the FIFO's
host_rst_fifo	I	Enables the FIFO reset controled by Host_Rstn
autostart	I	Enables the Autostart function as defined by ECSS-E-50-12A
link_start	I	Enables the Link Start function as defined by ECSS-E-50-12A
fifo_lpbk	I	Enables the FIFO loopback for debug
slrun_err	O	Error Flag indicating a SpaceWire error on the serial bus
phy_rstn	O	Reset the SpaceWire Transceiver

DESIGN FLOW for Implementation of the Aeroflex SpaceWire IP.

General Requirements

A general understanding of the tool flow for FPGA design is required to implement the Aeroflex SpaceWire IP. In addition, a basic understanding of the QuickLogic design tools including SpDE is also needed.



Software Requirements

Before beginning the design flow, the user should install the QuickLogic QuickWorks software. Then you must locate the spde.ini file in the directory pasic\spde\data. Open this file up in Wordpad and locate the section called [LOF]. At the end of the [LOF] section, add the line LOFGEN=TRUE.

Design File Mangement

Locate the directory where the QuickWorks tools have been installed. Find the directory on the CD called macro and place it in the pasic\spde\data directory. A summary of all the required files and their respective directory locations is contained in Table 7 for Verilog and Table 8 for VHDL. Table 9 contains files that are common to either language.

Verilog Example[1]

Simulation:

Use the ref_des1.v file to instantiate the SpaceWire LPH Model (spw_lph_design.v) and any user defined modules. In addition to the IP instantiation, the ref_des1.v file also instantiates the necessary I/O's and clock buffers needed by the IP. There are a number of High Drive input instantiations as well in the ref_de1s.v file. These use the hdpad_25um macro and if the signals associated with these instantiations are to be driven by internal signals in the user design, the instantiations will need to be commented out. The user can start by using the testbench files lph_top_test.v, fifo_interface.v and phy_model.v. Additional testbench files can be created to test any user designed functions and simulations run until functionality is satisfactory. One additional note on simulation is these files and flows have been tested using Aldec Active HDL version 7.2 which has the ability to identify the particular vendor the user is targeting. In this case, we target Quicklogic for the vendor and Eclipse as the device family. Other simulation tools may not have this capability and may require the user to compile the macros-25.v or macros-2.5.vhd files. Also, since the phy_model.v/vhd is a gate level netlist, you may also need to include the qlprims.v for Verilog designs or for VHDL designs the qlvtl95-25.vhd file. These are located in the pasic\spde\data directory of the Quicklogic install.

[1]For VHDL Flow substitute .vhd for .v wherever .v occurs

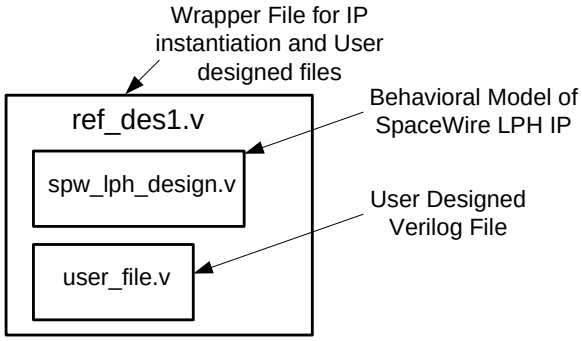


Figure 4. Simulation Example

Synthesis

Include the files ref_des1.v and the spw_lph.v when performing synthesis. Do not use the spw_lph_design.v file. The spw_lph_design.v file is only used for simulation and will not be treated as a "Black Box" by the synthesis tool. The macros-25.v file located in the pasic\spde\data directory if you are not using Synplicity for synthesis. If you are using Synplicity, refer to the note below. Most synthesis tools will give a warning when a Black Box has been defined (You may ignore the warning.). Of course, also include any user designed files. Enter in any constraints such as clock frequencies, etc., and perform synthesis. The results will produce either an .edf netlist or a .qdf netlist. It is important to note that the Mentor Precision Synthesis tool will only produce an edf netlist while Synplicity has the ability to produce either an edf or qdf netlist. Refer to Figure 5.

Note 1: When using Synplify Pro for Verilog designs, do not add the file macros-25.v. Instead, once you have created a project file, close the Synplicity application and open the Synplicity project file in a text editor. Now look for the section near the top called #add_file and add the following line under the #add_file section. This line does not need to be added for VHDL designs.

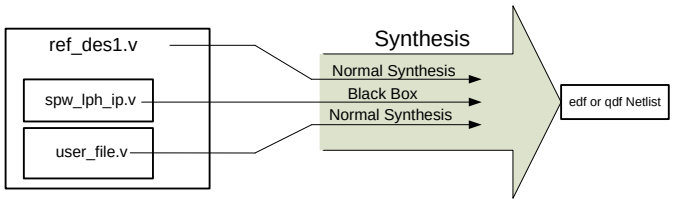


Figure 5. Synthesis Example

Place & Route

The user should create a directory for place and route and place the .edf or .qdf file in that directory. Also place the ref_des1.qcf and the file ending in .vh in this directory. The .qcf file contains timing constraints, as well as some fixed placed I/O's and the orientation for the macro. The .vh file is needed to maintain buses when generating gate level simulation files. Do not modify the existing fixed placed I/O's that are associated with the Phy chip interface in the .qcf file. The user can follow the format for the pin placement and place their own defined I/O's accordingly or remove the pins related to the FIFO interface if they are not being ported out of the device. Folllow the Place & Route procedure in the QuickLogic documentation to complete place and route of the device and generate a fuse (LOF) file. Timing can be analyzed and reported using the QuickLogic tools as well. Refer to Figure 6.

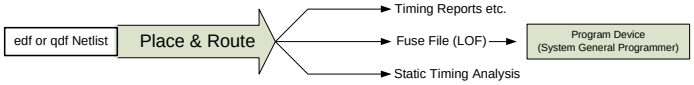


Figure 6. Place & Route Example

VERILOG			
FILE	**LOCATION**	**USAGE**	**DESCRIPTION**
qlprim-25.v	pasic\spde\data	Simulation and Synthesis	QuikLogic Primitives and is included with the SpDE install
spw_lph_design.v	Simulation Directory	Simulation	Behavioral Model of the SpaceWire Protocol Handler
lph_top_test.v	Simulation Directory	Simulation	Testbench
phy_model.v	Simulation Directory	Simulation	A testbench file. This file is the model of the UT200SpWPHY01 SpaceWire Transceiver
fifo_interface.v	Simulation directory	Simulation	Another testbench file that reads and writes to the Receive and Transmit FIFO's.
spw_lph.v	Synthesis Directory	Synthesis	Black Box Instantiation used for synthesis only
ref_des1.v	Simulation Directory	Simulation	Instantiates the macro for simulation. It also contains the required I/O instantiations. User defined modules can also be instantiated in this file as well.

Table 7. Required SpaceWire IP Files for Verilog Design

VHDL			
FILE	**LOCATION**	**USAGE**	**DESCRIPTION**
qlvtl95-25.vhd	pasic\spde\data		QuickLogic Primitives and is included in the SpDE install
spw_lph_design.vhd	Simulation Directory	Simulation	Behavioral Model of the SpaceWire Link
lph_top_test.vhd	Simulation Directory	Simulation	Testbench
phy_model.vhd	Simulation Directory	Simulation	A testbench file. This file is the model of the UT200SpWPHY01 SpaceWire Transceiver
fifo_interface.vhd	Simulation directory	Simulation	Another testbench file that reads and writes to the Receive and Transmit FIFO's.
spw_lph.vhd	Synthesis Directory	Synthesis	Black Box Instantiation used for synthesis only
ref_des1.vhd	Simulation Directory	Simulation	Instantiates the macro for simulation. It also contains the required I/O instantiations. User defined modules can also be instantiated in this file as well.

Table 8. Required SpaceWire IP Files for VDHL Design

FILE	LOCATION	USAGE	DESCRIPTION
spw_lph.mcr	pasic\spde\data\macro	Place & Route	SpDE File required for place and route of the macro
spw_lph.ini	pasic\spde\data\macro	Place & Route	SpDE File required for place and route of the macro
macro.ini	pasic\spde\data\macro	Place & Route	SpDE File required for place and route of the macro
ref_des1.qcf	Place and Route Directory	Place & Route	This is the constraint file needed to complete place and route. It sets the clock constraints as well as pin placements for fixed I/O's on the macro.
ref_des1.vh	Place and Route Directory	Place & Route	This file is needed for gate level simulations. It will cause the SpDE tool to maintain busses when generating gate level .v or .vhd netlists

Table 9. HDL Independent Files

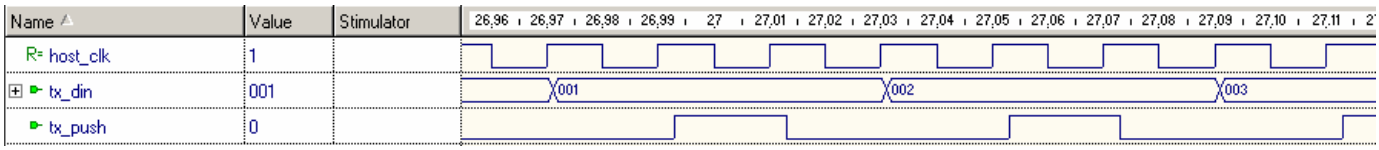


Figure 7. Transmit FIFO Write Timing

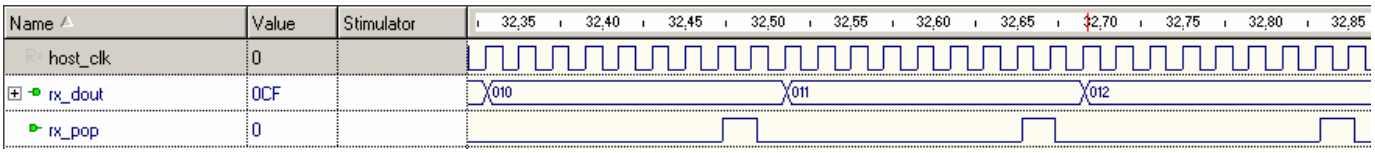


Figure 8. Receive FIFO Read Timing

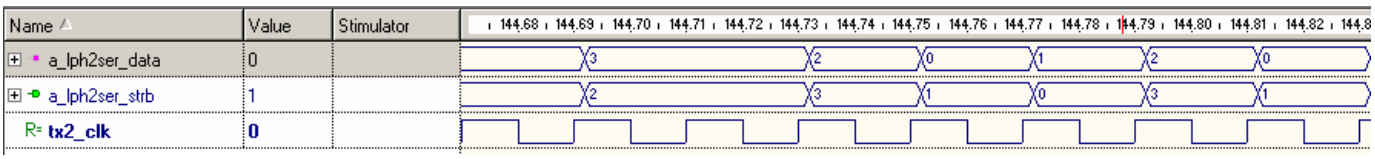


Figure 9. Transmit Data and Strobe Timing

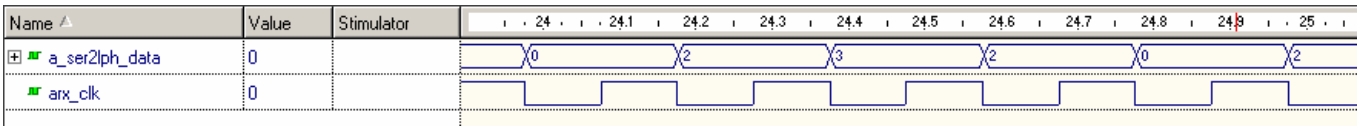
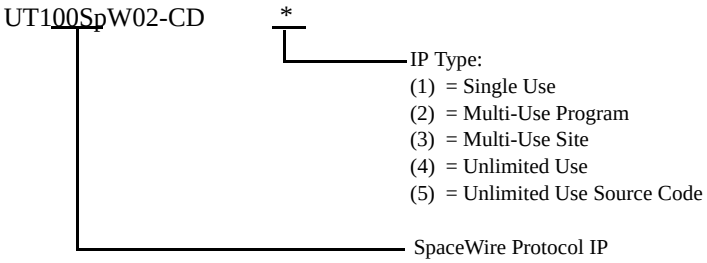


Figure 10. Receive Data Timing

ORDERING INFORMATION

UT100SpW02 SpaceWire Protocol IP:



Contact Aeroflex at 800-645-8862 for pricing.

SpaceWire Protocol IP License Descriptions

(1) Single Use: Allows the licensee to instantiate the SpW IP into a single design at a single site. "Instantiate" means the process of implementing a representation of the IP Core within a UT6325 RadTol Eclipse FPGA.

(2) Multi-Use Program: Allows the licensee to instantiate the SpW IP into a multiple designs for a single location/site and single program. "Instantiate" means the process of implementing a representation of the IP Core within a UT6325 RadTol Eclipse FPGA.

(3) Multi-Use Site: Allows the licensee to instantiate the SpW IP into multiple designs and programs for a single location. "Instantiate" means the process of implementing a representation of the IP Core within a UT6325 RadTol Eclipse FPGA.

(4) Unlimited Use: Allows the licensee to "Instantiate" the SpW IP into multiple designs, multiple locations, and multiple programs. "Instantiate" means the process of implementing a representation of the IP Core within a UT6325 RadTol Eclipse FPGA.

(5) Unlimited Use Source Code: Allows the licensee to "Instantiate" the SpW source code into multiple designs, multiple locations, and multiple programs. "Instantiate" means the process of implementing a representation of the IP Core within an integrated circuit or part of an integrated circuit. The type of integrated circuit in which the models can be "instantiated" is open and not restricted to any particular technology.

All licenses receive:

- Maintenance for 1 year
- Support for 1 year
- Warranty for 1 year
- Royalty 0%
- Deliverables: Encrypted Behavioral Model, Simple Testbench, User’s Guide, Macro Files, Constraint Files and Reference Design

Aeroflex Colordo Springs - Datasheet Definition

Advanced Datasheet - Product In Development

Preliminary Datasheet - Shipping Prototype

Datasheet - Shipping QML & Reduced Hi-Rel

COLORADO
Toll Free: 800-645-8862
Fax: 719-594-8468

INTERNATIONAL
Tel: 805-778-9229
Fax: 805-778-1980

NORTHEAST
Tel: 603-888-3975
Fax: 603-888-4585

SE AND MID-ATLANTIC
Tel: 321-951-4164
Fax: 321-951-4254

WEST COAST
Tel: 949-362-2260
Fax: 949-362-2266

CENTRAL
Tel: 719-594-8017
Fax: 719-594-8468

www.aeroflex.com info-ams@aeroflex.com

AEROFLEX
A passion for performance.

Our passion for performance is defined by three attributes represented by these three icons: solution-minded, performance-driven and customer-focused