CompactRIO White Rabbit Module

Hardware Short Test

Document Version 1.01
19.01.2015
Introduction

This is a brief description of the procedure to test the hardware of a CompactRIO White Rabbit (CRIO-WR) module. Required firmware and software used for the testing can be downloaded here: http://www.ohwr.org/projects/crio-wr/files

1 Hardware, firmware and software requirements

- 1 x CRIO-WR reference board, mode master, incl. SFP FO transceiver (e.g. Axcen AXGE-3454-0531)
- 1 x SFP FO transceiver (e.g. Axcen AXGE-1254-0531), used for the DUT
- 1 x Fibre cable (G652 type with LC connectors)
- 1 x Power supply +5 V (+4.75 V ... +5.25 V), 2 A
- 1 x JTAG adapter (e.g. Xilinx Platform Cable USB II), incl. USB cable
- 1 x Ampère-Meter (>= 2 A DC)
- 1 x Volt-Meter (>= 5 V DC)
- 1 x Oscilloscope (>= 2 channels, >= 500 MHz bandwidth)
- 1 x USB cable (CRIO-WR)
- 2 x Adapter cable “CRIO-Interface” (see appendix)
- 2 x Adapter cable “User-Interface” (see appendix)
- 1 x Adapter cable “JTAG” (see appendix)
- FPGA PROM file, LabVIEW project files
- PC with installed software:
  - CP210x USB to UART bridge VCP driver (Silicon Labs)
  - Terminal program (e.g. PuTTY)
  - Xilinx ISE / iMPACT
  - NI LabVIEW 2013 SP1 (with FPGA and Real-Time package)
  - NI CompactRIO MDK 2.0

2 More information

- White Rabbit PTP Core: http://www.ohwr.org/projects/wr-cores/wiki/wrpc_core
- NI CompactRIO MDK 2.0: http://sine.ni.com/nips/cds/view/p/lang/de/nid/14548
3 Test procedure

3.1 Power Supply

3.1.1 Connect the adapter cable “CRIO-Interface” to CON800 of the DUT and to the power supply (GND, CRIO_VCC, +5 V, ±5 %).

3.1.2 Check supply current @ +5 V: < 1 mA.
No supply current flowing because on-board FET Q900 (SI7101DN) is turned off.

3.1.3 Connect signal CRIO_SLEEP of the adapter cable “CRIO-Interface” to GND.
This turns on FET Q900 and starts on-board DC-DC regulator (U950, LTC3417AEDHC-2) which generates local +3.3 V and +1.2 V.

3.1.4 Check P3V3: +3.3 V, ±3 %.
Use the marked pads on-board to connect the Volt-Meter (P3V3, GND).

3.1.5 Check P1V2: +1.2 V, ±3 %.
Use the marked pads on-board to connect the Volt-Meter (P1V2, GND).

3.1.6 Check supply current @ +5 V: < 180 mA.

3.1.7 Check LED995 (PON) and LED998 (PGD): both are on.

3.2 Program FPGA Configuration FLASH Memory

3.2.1 Connect the adapter cable “JTAG” to CON660 of the DUT and to the JTAG adapter.

3.2.2 Use Xilinx iMPACT to write the FPGA configuration file (crio_wr_top.mcs) to the FLASH memory (U600, M25P32).

3.2.3 Disconnect / connect signal CRIO_SLEEP of the adapter cable “CRIO-Interface” from / to GND.
This turns off / on the on-board power supply and causes FPGA configuration from FLASH memory.

3.3 Main Functionality

3.3.1 Check supply current @ +5 V: < 380 mA.

3.3.2 Check LED995 (PON) and LED998 (PGD): both are on.

3.3.3 Check LED760 (LDB) and LED761 (LDA): both are on.

3.3.4 Check LED760 (LDB) and LED761 (LDA) whilst pressing button SW770 (SWA): both are off.

3.3.5 Connect the adapter cable “User-Interface” to CON730 of the DUT.

3.3.6 Check LVDS signal pair USERIO_2 (100R Z_D) at adapter cable “User-Interface”:
V_OD >= 247 mV, V_OCM 1.125 V ... 1.375 V, f = 62.5 MHz typ. (pllout_clk_sys).
3.3.7 Check LVDS signal pair USERIO_3 (100R Z_D) at adapter cable “User-Interface”:
\( V_{OD} \geq 247 \text{ mV}, V_{OCM} 1.125 \text{ V} \ldots 1.375 \text{ V}, f = 125 \text{ MHz typ. (pllout_clk_ref)}. \)

3.3.8 Check LVDS signal pair USERIO_4 (100R Z_D) at adapter cable “User-Interface”:
\( V_{OD} \geq 247 \text{ mV}, V_{OCM} 1.125 \text{ V} \ldots 1.375 \text{ V}, f = 31.25 \text{ MHz typ. (dummy_cntr_sys)}. \)

3.3.9 Check LVDS signal pair USERIO_5 (100R Z_D) at adapter cable “User-Interface”:
\( V_{OD} \geq 247 \text{ mV}, V_{OCM} 1.125 \text{ V} \ldots 1.375 \text{ V}, f = 31.25 \text{ MHz typ. (dummy_cntr_dmt)}. \)

3.4 USB to UART Bridge

3.4.1 Connect the USB cable to CON700 of the DUT and to a free USB port of the PC.

3.4.2 Check availability of additional (virtual) COM port on the PC:
  Silicon Labs CP210x USB to UART Bridge (COMx).

3.4.3 Start a terminal software (COMx, 115200 bps, 8 data bits, 1 stop bit, no parity, no flow control) to interact with the WRPC shell of the DUT.

3.4.4 Press [ENTER] in the terminal and check the WRPC shell prompt:
  wrc#

3.5 EEPROM (WR)

3.5.1 Before writing to the EEPROM, it is good to stop the ptp daemon:
  wrc# ptp stop

3.5.2 Create an empty sfp database and add two Axcen transceivers:
  wrc# sfp erase
  wrc# sfp add AXGE-1254-0531 180707 148323 72169888
  wrc# sfp add AXGE-3454-0531 180707 148323 -73685416

3.5.3 Check the created sfp database:
  wrc# sfp show
  1: PN:AXGE-1254-0531  dTx: 180707,  dRx: 148323,  alpha: 72169888
  2: PN:AXGE-3454-0531  dTx: 180707,  dRx: 148323,  alpha: -73685416

3.5.4 Write an init script to the EEPROM:
  wrc# init erase
  wrc# init add ptp stop
  wrc# init add sfp detect
  wrc# init add sfp match
  wrc# init add mode slave
  wrc# init add calibration
  wrc# init add ptp start
3.6 Testing WR Link

3.6.1 DUT: Disconnect signal CRIO_SLEEP of the adapter cable “CRIO-Interface” from GND. This turns off DUT’s on-board power supply.

3.6.2 DUT: Insert SFP FO transceiver (AXGE-1254-0531) into CON300.

3.6.3 Reference board: Connect the adapter cable “CRIO-Interface” to CON800 and to the power supply (GND, CRIO_VCC, +5 V, ±5 %).

3.6.4 Reference board: Connect the adapter cable “User-Interface” to CON730.

3.6.5 Connect the fibre cable between the reference board and the DUT.

3.6.6 Reference board: Connect signal CRIO_SLEEP of the adapter cable “CRIO-Interface” to GND. This turns on the on-board power supply of the reference board.

3.6.7 DUT: Connect signal CRIO_SLEEP of the adapter cable “CRIO-Interface” to GND. This turns on the on-board power supply of the DUT.

3.6.8 DUT: Check supply current @ +5 V: < 480 mA.

3.6.9 DUT: Shortly press button SW770 (SWA) and check the log messages that confirm the init script execution like this:

```plaintext
wrc# WR Core: starting up...
get_persistent_mac: Using W1 serial number
ID: cafebabe
PPSi for WRPC. Commit ppsi-v2013.11.1-l-g76d6d83, built on Dec 20 2013
t24p read from EEPROM: 6800 ps
Loops per jiffy: 20817
Locking PLL
Slave Only, clock class set to 255
executing: ptp stop
executing: sfp detect
AXGE-1254-0531
executing: sfp match
SFP matched, dTx=180707, dRx=148323, alpha=72169888
executing: mode slave
Locking PLL
executing: calibration
Found phase transition in EEPROM: 6800ps
executing: ptp start
Slave Only, clock class set to 255
wrc# t24p value is 6800 ps, storing to EEPROM
```

3.6.10 After a while, the DUT (WR slave) should be synchronized to the reference board (WR master). Check the quad LEDs (LED750) of the DUT:

- LED750_4 (top): Sporadically flashing (link activity)
- LED750_3: On (link up)
- LED750_2: On (time valid)
- LED750_1 (bottom): Periodically flashing (PPS)
3.6.11 The PPS signal of the DUT should be phase aligned to the PPS signal of the reference board. Check LVDS signal pairs USERIO_1 (100R Z_D) at adapter cables “User-Interface”:
\[ V_{OD} \geq 247 \text{ mV}, \ V_{OCM} = 1.125 \text{ V} \ldots 1.375 \text{ V}, \ f = 1 \text{ Hz typ.}, \ t_{OFFS\_REF\_DUT} < 1 \text{ ns} \]

4 Testing CRIO Interface

4.1.1 First copy the two Labview folders “MDK-MFG” and “UZH-0001” to the following path:
\[ C:\Program\Files\(x86\)\National\Instruments\LabVIEW\2013\Targets\NI\FPGA\cRIO\other \]

4.2 SPI EEPROM

4.2.1 Put the CRIO-WR module into the slot 8 of the cRIO chassis. Please notice that the FPGA configuration memory of the CRIO-WR module must contain the correct firmware (see 3.2).

4.2.2 Open the “MDK-MFG Development Mode.lvproj” Labview Project and configure the IP address of your cRIO.
4.2.3 Open “MDK-MFG Development Mode (FPGA).vi” and start the VI. Now the VI is compiled. This step may take about ten minutes.

4.2.4 Deploy the project to the cRIO.

4.2.5 Open the “MDK-MFG Read EEPROM Contents from Module (Host).vi” and configure the resource name of the cRIO and the path of the Read.txt file.

4.2.6 Start the VI and then the EEPROM data will be written into the txt file. If the EEPROM is new, every byte has the value 0xFF.

4.2.7 Open the “MDK-MFG Write EEPROM Contents to Module (Host).vi” and configure the resource name of the cRIO and the path of the Write.txt file.

4.2.8 Start the VI and now the configuration file will be written into the EEPROM. For a short check you can read back the data from the EEPROM.

4.2.9 Now you know that the communication with the EEPROM works properly.
4.3 SPI FPGA

4.3.1 Put the CRIO-WR module into the slot 8 of the cRIO chassis.

4.3.2 Open the “UZH-0001 Development Mode.lvproj” Labview project and configure the IP address of your cRIO.

4.3.3 Open “UZH-0001_ApiTest.vi” and start the VI. Now the VI is compiled. This step may takes about ten minutes.

4.3.4 Press the “Send Command” button and then ten bytes are read and written by the SPI interface. If the first byte from the write data is 0x80, then you read the actual timestamp from the cRIO-WR module.
4.3.5 The communication between the cRIO cassis und the CRIO-WR module works properly if you can read correct data over the internal SPI interface.
5 Appendix

5.1 Adapter cable “CRIO-Interface”

Table below shows only minimum required connections

<table>
<thead>
<tr>
<th>Pin # HDSUB-15 male</th>
<th>Signal</th>
<th>Banana plug color</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>CRIO_VCC</td>
<td>Red</td>
</tr>
<tr>
<td>8</td>
<td>CRIO_SLEEP</td>
<td>Orange</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Black</td>
</tr>
</tbody>
</table>
5.2 Adapter cable “User-Interface”

<table>
<thead>
<tr>
<th>Pin # HDSUB-15 male</th>
<th>Signal</th>
<th>Pin # test connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>USERIO_1_P</td>
<td>1_1</td>
</tr>
<tr>
<td>1</td>
<td>USERIO_1_N</td>
<td>1_2</td>
</tr>
<tr>
<td>7</td>
<td>USERIO_2_P</td>
<td>2_1</td>
</tr>
<tr>
<td>2</td>
<td>USERIO_2_N</td>
<td>2_2</td>
</tr>
<tr>
<td>8</td>
<td>USERIO_3_P</td>
<td>3_1</td>
</tr>
<tr>
<td>3</td>
<td>USERIO_3_N</td>
<td>3_2</td>
</tr>
<tr>
<td>9</td>
<td>USERIO_4_P</td>
<td>4_1</td>
</tr>
<tr>
<td>4</td>
<td>USERIO_4_N</td>
<td>4_2</td>
</tr>
<tr>
<td>10</td>
<td>USERIO_5_P</td>
<td>5_1</td>
</tr>
<tr>
<td>5</td>
<td>USERIO_5_N</td>
<td>5_2</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
<td>6_1</td>
</tr>
</tbody>
</table>
5.3 Adapter cable “JTAG” (Xilinx Platform Cable USB II to JTAG interface CRIO-WR)

<table>
<thead>
<tr>
<th>Pin # Xilinx Adapter</th>
<th>Signal</th>
<th>Cable color</th>
<th>Pin # CRIO-WR JTAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VREF</td>
<td>Red</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Black</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>N.C. (key)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>TCK</td>
<td>Yellow</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>HALT</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>TDO</td>
<td>Purple</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>TDI</td>
<td>White</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>N.C. (key)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>TMS</td>
<td>Green</td>
<td>4</td>
</tr>
</tbody>
</table>
5.4 **CRIO-WR Reference Board (Mode Master)**

A reference board operating as WR master is used for the tests described above. If there is no reference board available, a new CRIO-WR board can be setup as reference board. To do this, follow the description below.

Test two new boards as described above in steps 3.1 ... 3.5.3. Write the init script as described above in step 3.5.4 to the EEPROMs of the boards, whereby “mode” of one board must be set to “master” instead of “slave”. Test the WR link between the boards as described in step 3.6.

WRPC needs to make a calibration of t2/4 phase transition value. It has to be done only once for a new bitstream and is performed automatically when WRPC runs in Slave mode. That is why it is very important, even if WRPC is meant to run in Master mode, to configure it to Slave for a moment and connect it to any WR Master. That has to be repeated every time a new bitstream (gateware) is deployed. Measured value is automatically stored to EEPROM and used later in Master or GrandMaster mode.

To do this, power off both boards and exchange the SFP transceivers of the boards, then setup the init scripts such that the Master is now Slave and vice versa. Test the WR link as described in step 3.6.

Finally, erase the init script of board # 1, insert SFP AXGE-3454-0531 and write the init script such that the board is initialized as master. This board can now be used as reference board.