Hardware Data Sheet Section III Addendum

ET1810 / ET1811 / ET1812 and ET1815 / ET1816

EtherCAT® Slave Controller
IP Core for Intel® and Xilinx® FPGAs

Version 3.0
Date: 2019-05-31

BECKHOFF

DOCUMENT ORGANIZATION

The Beckhoff EtherCAT Slave Controller (ESC) documentation covers the following Beckhoff ESCs:

- ET1200
- ET1100
- EtherCAT IP Core for Intel® FPGAs
- EtherCAT IP Core for Xilinx® FPGAs
- ESC20

The documentation is organized in three sections. Section I and section II are common for all Beckhoff ESCs, Section III is specific for each ESC variant.

The latest documentation is available at the Beckhoff homepage (http://www.beckhoff.com).

Section I – Technology (All ESCs)

Section I deals with the basic EtherCAT technology. Starting with the EtherCAT protocol itself, the frame processing inside EtherCAT slaves is described. The features and interfaces of the physical layer with its two alternatives Ethernet and EBUS are explained afterwards. Finally, the details of the functional units of an ESC like FMMU, SyncManager, Distributed Clocks, Slave Information Interface, Interrupts, Watchdogs, and so on, are described.

Since Section I is common for all Beckhoff ESCs, it might describe features which are not available in a specific ESC. Refer to the feature details overview in Section III of a specific ESC to find out which features are available.

Section II – Register Description (All ESCs)

Section II contains detailed information about all ESC registers. This section is also common for all Beckhoff ESCs, thus registers, register bits, or features are described which might not be available in a specific ESC. Refer to the register overview and to the feature details overview in Section III of a specific ESC to find out which registers and features are available.

Section III – Hardware Description (Specific ESC)

Section III is ESC specific and contains detailed information about the ESC features, implemented registers, configuration, interfaces, pinout, usage, electrical and mechanical specification, and so on. Especially the Process Data Interfaces (PDI) supported by the ESC are part of this section.

Additional Documentation

Application notes and utilities can also be found at the Beckhoff homepage. Pinout configuration tools for ET1100/ET1200 are available. Additional information on EtherCAT IP Cores with latest updates regarding design flow compatibility, FPGA device support and known issues are also available.

DOCUMENT HISTORY

Version	Comment
1.0	Initial release
1.1	Update to EtherCAT IP Core V3.0.2/V3.00c
1.2	Update to EtherCAT IP Core V2.4.3/V2.04d
1.3	Update to Altera Quartus 13.0 SP1 and Xilinx Vivado 2013.2
1.4	Update to Altera Quartus 13.1, Xilinx Vivado 2013.3, and ISE14.7
1.5	• Update to EtherCAT IP Core V3.0.5/V3.00f • Update to Xilinx Vivado 2013.4, editorial changes
1.6	• Update to EtherCAT IP Core V3.0.6/V3.00g • Update to Xilinx Vivado 2014.1
1.7	• Update to Altera Quartus 14.0, Xilinx Vivado 2014.2 • Added Altera MAX10, Xilinx Kintex UltraScale, and Virtex UltraScale
1.8	• Update to EtherCAT IP Core V3.0.9/V3.00j • Update to Xilinx Vivado 2014.3 • Added known designflow issues chapters
1.9	Update to EtherCAT IP Core V3.0.9 Patch 1/V3.00j Patch 1
2.0	• Update to EtherCAT IP Core V2.4.4/V2.04e and V3.0.10/V3.00k • Update to Altera Quartus 14.1, Xilinx Vivado 2014.4 • Added Altera Arria 10
2.1	Update to Altera Quartus 15.0, Xilinx Vivado 2015.1
2.2	Update to Xilinx Vivado 2015.2
2.3	• Update to Altera Quartus Prime 15.1, Xilinx Vivado 2015.3 and 2015.4 • Removed issue “Tristate drivers are not properly connected” with Arria 10, because it produces only a warning, while the implementation is correct
2.4	Update to Xilinx Vivado 2016.1
2.5	Update to Altera Quartus Prime Standard/Pro 16.0
2.6	• Added Vivado 2016.1 known issue “Synthesis fails with an RTL assertion error” • Update to Xilinx Vivado 2016.2 • Update to Altera Quartus Prime Standard/Pro 16.0.1
2.7	• Replaced Altera by Intel • Update to EtherCAT IP Core V3.0.10 Patch 1/V3.00k Patch 1 • Update to Intel Quartus 16.1, Xilinx Vivado 2016.4 • Added Xilinx Zynq UltraScale+ • Added MD5 checksums
2.8	• Update to Intel Quartus 17.0, Xilinx Vivado 2017.1 • Added Intel Cyclone 10 GX and LP • Added Xilinx Spartan-7
2.9	• Update to Intel Quartus 17.1-18.0 • Update to Xilinx Vivado 2017.2-2018.2 • Update known issues
3.0	• Update to EtherCAT IP Core V3.0.10 Patch 2/V3.00k Patch 2 • Update to Intel Quartus Prime Standard 18.1, Quartus Prime Pro 19.1 • Update to Xilinx Vivado 2019.1

CONTENTS

1 Overview

This document provides latest release notes, documentation addendum, supported IP Core design flows, and supported IP Core FPGA types for the following Beckhoff EtherCAT Slave Controllers:

- EtherCAT IP Core for Intel® FPGAs (up to V2.4.4 Patch 1 / V3.0.10 Patch 2)
- EtherCAT IP Core for Xilinx® FPGAs (up to V2.04e Patch 1 / V3.00k Patch 2)

Refer to the ESC data sheets for further information. The ESC data sheets are available from the Beckhoff homepage (http://www.beckhoff.com).

2 EtherCAT IP Core for Intel FPGAs

The former *EtherCAT IP Core for Altera FPGAs* is now called *EtherCAT IP Core for Intel FPGAs*. There is no technical difference.

2.1 FPGA design tool compatibility

Starting with V2.4.0, Qsys is supported (the example designs of V2.4.0 are SoPC builder based).

Table 1: EtherCAT IP Core for Intel FPGAs compatibility with Intel Quartus Tools

IP Core version	Release date	Tool compatibility		Quartus II / Quartus Prime Standard / NIOS EDS version compatibility																Quartus Prime Pro version compatibility	
		SoPC Builder	Qsys/ Platform Designer	5.1 SP2	6.1-7.1 SP1	7.2 SP2-8.0	9.0 SP1	9.1	10.0	10.1	11.0	11.1 SP2	12.0 SP1	12.1 SP1	13.0 SP1	13.1.4	14.0[1]	14.1[1]	15.0-17.0[1]	17.1-18.1[1]	16.0-17.0	17.1-19.1
1.0.0	7/2006	●	-	●																		
1.1.0	11/2006	●	-	●																		
1.1.1	1/2007	●	-	●	●																	
2.0.0	8/2007	●	-	●	●																	
2.2.0	6/2008	●	-	●	●	●																
2.2.1	6/2009	●	-	-	-	-	●															
2.3.0	12/2009	●	-	-	-	-	-	●														
2.3.1	2/2010	●	-	-	-	-	-	●														
2.3.2	3/2010	●	-	-	-	-	-	●	●	●	○											
2.4.0	3/2011	●	●	-	-	-	-	-	-	●	●	●	-									
2.4.0 Patch 5	6/2012	●	●	-	-	-	-	-	-	●	●	●	●									
2.4.3	7/2013	●	●	-	-	-	-	-	-	●	●	●	●	●	●	●	●					
2.4.4	1/2015	●	●	-	-	-	-	-	-	●	●	●	●	●	●	●	●	●				
2.4.4 Patch 1	6/2016	●	●	-	-	-	-	-	-	●	●	●	●	●	●	●	●	●				
3.0.0	3/2013	●	●	-	-	-	-	-	-	-	●	●	●	●								
3.0.1	3/2013	●	●	-	-	-	-	-	-	-	●	●	●	●	●							
3.0.2	5/2013	●	●	-	-	-	-	-	-	-	●	●	●	●	●	●						
3.0.5	2/2014	●	●	-	-	-	-	-	-	-	●	●	●	●	●	●						
3.0.6	4/2014	●	●	-	-	-	-	-	-	-	●	●	●	●	●	●	●					
3.0.9	9/2014	●	●	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●				
3.0.9 Patch 1	11/2014	●	●	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●				
3.0.10	1/2015	●	●	-	-	-	-	-	-	-						●	●	●	●		●	
3.0.10 Patch 2	5/2019	●	●	-	-	-	-	-	-	-						●	●	●	●	●	●	●

Table 2: Tool compatibility legend

Symbol	Description
●	Compatible (maybe solved issues with example designs)
○	Compatible, issues with some example designs
-	Incompatible
	Not tested

[1] Cyclone III devices are not supported anymore by Quartus, corresponding example designs cannot be synthesized. Refer to known issues for more details.

2.2 FPGA device compatibility

Starting with V2.4.0 Patch 5, the family support check of the MegaWizard plugin has been turned off (a license update is now sufficient for new FPGA families).

Stratix 10 and Cyclone 10 GX support cannot be evaluated because the required licenses for Quartus Prime Pro are not available.

Table 3: EtherCAT IP Core for Intel FPGAs compatibility with Intel FPGAs

IP Core version	Cyclone	Cyclone II	Cyclone III	Cyclone III LS	Cyclone IV E	Cyclone IV GX	Cyclone V / Cyclone V SoC	Cyclone 10 LP	Arria GX	Arria II GX	Arria II GZ	Arria V	Arria V GZ	Arria 10	Stratix/ Stratix II	Stratix GX/ Stratix II GX	Stratix III	Stratix IV	Stratix V	Intel Atom E6x5C	MAX10
1.0.0	●	●	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
1.1.0	●	●	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
1.1.1	●	●	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
2.0.0	●	●	●	-	-	-	-	-	●	-	-	-	-	-	●	●	●	-	-	-	-
2.2.0	●	●	●	-	-	-	-	-	●	-	-	-	-	-	●	●	●	●	-	-	-
2.2.1	●	●	●	-	-	-	-	-	●	●	-	-	-	-	●	●	●	●	-	-	-
2.3.0	●	●	●	●	●	●	-	-	●	●	-	-	-	-	●	●	●	●	-	-	-
2.3.1	●	●	●	●	●	●	-	-	●	●	-	-	-	-	●	●	●	●	-	-	-
2.3.2	●	●	●	●	●	●	-	-	●	●	-	-	-	-	●	●	●	●	-	-	-
2.4.0	●	●	●	●	●	●	-	-	●	●	●	-	-	-	●	●	●	●	●	●	-
2.4.0 Patch 5	●	●	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●	●	●
2.4.3	●	●	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●	●	●
2.4.4	●	●	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●	●	●
2.4.4 Patch 1	●	●	●	●	●	●	●		●	●	●	●	●		●	●	●	●	●	●	●
3.0.0		●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●	●
3.0.1		●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●	●
3.0.2		●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●	●
3.0.5		●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●	●
3.0.6			●	●	●	●	●			●	●	●	●				●	●	●	●	●
3.0.9			●	●	●	●	●			●	●	●	●				●	●	●	●	●
3.0.9 Patch 1			●	●	●	●	●			●	●	●	●				●	●	●	●	●
3.0.10			●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●
3.0.10 Patch 2			●	●	●	●	●	●		●	●	●	●	●			●	●	●	●	●

Table 4: FPGA compatibility legend

Symbol	Description
●	Compatible
-	Incompatible
	Not tested

2.3 FPGA device license support

Every license for the EtherCAT IP Core contains restrictions on supported FPGA devices. All FPGA device families which have license support at the date when the license is generated are unlocked. The following table shows the dates of introduction of the FPGA license support. All EtherCAT IP Core licenses which are generated past the date will include the marked FPGA devices.

Table 5: EtherCAT IP Core for Intel FPGAs License support

First date of license support	Cyclone/ Cyclone II	Cyclone III	Cyclone III LS	Cyclone IV E	Cyclone IV GX	Cyclone V / Cyclone V SoC	Cyclone 10 GX / LP	Arria GX	Arria II GX	Arria II GZ	Arria V	Arria V GZ	Arria 10	Stratix/ Stratix II	Stratix GX/ Stratix II GX	Stratix III	Stratix IV	Stratix V	Stratix 10	Intel Atom E6x5C	MAX10
Initial	●	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	-	-	-	-	-
2007-03-28	●	●	-	-	-	-	-	-	-	-	-	-	-	●	●	●	-	-	-	-	-
2007-08-16	●	●	-	-	-	-	-	●	-	-	-	-	-	●	●	●	-	-	-	-	-
2008-06-03	●	●	-	-	-	-	-	●	-	-	-	-	-	●	●	●	●	-	-	-	-
2009-05-27	●	●	-	-	-	-	-	●	●	-	-	-	-	●	●	●	●	-	-	●	-
2009-07-02	●	●	●	-	-	-	-	●	●	-	-	-	-	●	●	●	●	-	-	●	-
2009-11-27	●	●	●	-	●	-	-	●	●	-	-	-	-	●	●	●	●	-	-	●	-
2010-06-07	●	●	●	●	●	-	-	●	●	-	-	-	-	●	●	●	●	-	-	●	-
2011-03-22	●	●	●	●	●	-	-	●	●	●	●	-	-	●	●	●	●	●	-	●	-
2012-06-25	●	●	●	●	●	●	-	●	●	●	●	-	-	●	●	●	●	●	-	●	-
2013-02-15	●	●	●	●	●	●	-	●	●	●	●	●	-	●	●	●	●	●	-	●	-
2014-05-06	●	●	●	●	●	●	-	●	●	●	●	●	-	●	●	●	●	●	-	●	●
2014-05-08	●	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	●	●	●
2017-04-20	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Table 6: License support legend

Symbol	Description
●	Supported
-	Not supported

2.4 Known Designflow Issues

2.4.1 Active (at least until Quartus Prime Standard 18.1/Quartus Prime Pro 19.1)

2.4.1.1 Quartus Prime Pro 17.0 – today: Upgrade to Qsys Pro/Platform Designer Pro fails due to missing simulation support

Upgrading a project with EtherCAT IP Core from Qsys to Qsys Pro/Platform designer Pro fails with an error message

```
Error: EtherCAT_0: EtherCAT does not support generation for Verilog Simulation. Generation is available for: Quartus Synthesis.
```

This is related to simulation of the EtherCAT IP Core, which is not supported.

Solution
Please open the upgraded system in Qsys Pro, and generate the HDL without enabling simulation model creation.

2.4.1.2 Quartus Prime Standard/Pro 16.1 – today: Windows setup does not integrate the EtherCAT IP Core into the Quartus IP catalog

The windows setup does not detect installations of Quartus Prime Standard/Pro since 16.1, so it does not offer integrating the EtherCAT IP Core into these versions. This is caused by the EtherCAT IP Core setup, which is a 32 bit executable. It cannot locate 64 bit applications, so Quartus is not found.

Solution
Integrate the EtherCAT IP Core into Quartus manually by copying the *<IPInst_dir>\quartus_add* content into your Quartus installation folder, according to the EtherCAT IP Core data sheet (section III, chapter 3.6).

2.4.1.3 Quartus Prime Standard 16.0 – today: DE2-115 MII/RGMII example designs: synthesis fails because of an additional ALTPLL signal

Synthesis fails because Qsys does no longer generate the ALTPLL phasedone signal, when this feature is not used. Previous versions of Quartus generated that signal, so it was referenced by the example designs.

Solution
Delete the signal *altpll_0_phasedone_conduit_export* from the component declaration and instantiation in the top level file.

2.4.1.4 Quartus Prime Pro 16.0 – today: Assertions and report messages are not evaluated

Quartus Prime Pro does not evaluate assertions and report messages anymore, even if they are based on constants or generics. As a result, the messages shown by the EtherCAT IP core source code (text configuration box) are no longer visible. Additionally, assertions resulting in errors or failures are no longer causing the synthesis to fail, thus, illegal configurations might become synthesized to dysfunctional logic. This is unlikely to happen since the configuration interface prevents illegal configurations, but manual changes might cause them.

This issue has already been reported to Intel for Quartus Prime Pro 15.1 (beta)

Solution

None.

2.4.1.5 Quartus 14.0 – today with EtherCAT IP Core before V3.0.10 Patch 1: Cyclone IV DBC4CE55 with NIOS example design PLL cannot be upgraded or edited

The PLL IP cannot be upgraded/edited because it contains a Cyclone III device as the target device family, although a Cyclone IV device is actually used:

```
Error (14921): Error upgrading IP component "pll.vhd".
```

Solution
Please change the device family in the file *pll.vhd* from

`"Cyclone III"` to `"Cyclone IV E"`

There are three occurrences of “Cyclone III”, just replace all of them by “Cyclone IV E”, and upgrade or edit the *pll.vhd* IP afterwards.

2.4.1.6 Quartus 14.0 – today: Cyclone III example designs are not synthesizable

Cyclone III devices are not supported anymore by Quartus 14, corresponding example designs cannot be synthesized.

Solution
None. Use previous Quartus versions.

2.4.1.7 EtherCAT IP Core V3.0.0 – today: DE2-115 NIOS RGMII example design is missing a register for DDR input signals

The EtherCAT IP Core evaluates RGMII DDR input signals at the rising clock edge, and it expects that the DDR_H input was sampled before the DDR_L input. For RX_CTL, this means that DDR_H contains RX_DV, while DDR_L contains RX_ER for the same receive clock cycle.

The DE2-115 NIOS RGMII example design uses an Intel DDR input cell which uses a reversed DDR_L/DDR_H ordering. Due to a certain behaviour of the PHYs on this boards, this will not cause problems, but using the example design for other PHYs can result in receive errors.

Solution
An additional register has to be added to the DDR_H path for each RX_CTL/RX_DATA[3:0] signal, according to the following schematic:

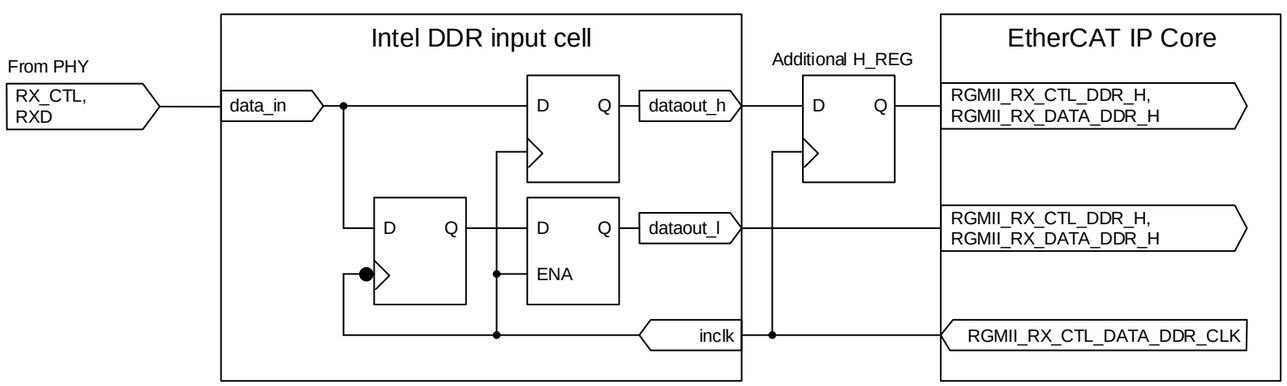

Figure 1: Additional H_REG for RGMII inputs using Intel DDR input cells

Example code for the additional H_REG for one port (duplicate and update port numbers for additional ports):

```
Port0_H_REG: process(RGMII_RX_CTL_DATA_DDR_CLK0)
begin
      if rising_edge(RGMII_RX_CTL_DATA_DDR_CLK0) then            -- port 0 receive clock
            RGMII_RX_CTL_DDR_H0_reg  <= RGMII_RX_CTL_DDR_H0;     -- H_register for port 0 RX_CTL
            RGMII_RX_DATA_DDR_H0_reg <= RGMII_RX_DATA_DDR_H0;    -- H_register for Port 0 RX_DATA[3:0]
      end if;
end process;
```

2.4.1.8 TimeQuest with EtherCAT IP Core V3.0.x: Recovery timing violation inside the EtherCAT IP Core from 25 MHz rising edge to 100 MHz falling edge (5 ns)

In rare cases, a recovery timing violation occurs inside the EtherCAT IP Core between two flip-flops. The start flip-flop is called HNR3530 (using the 25 MHz clock at the rising edge), and the end flip-flop is called BNR3543.ACL2024 (using the 100 MHz clock at the falling edge). Quartus infers a CLKCTRL instance on the path, which sometimes causes the recovery timing violation.

Solution
Quartus can be forced to connect both flip-flops individually, without using the CLKCTRL instance. All other targets of the CLKCTRL will remain untouched. Please use the Assignment Editor to add this assignment:

```
From:               *ETHERCAT_IPCORE_V2:ethercat_0|ETHERCAT_CORE:\CORE_INST:CORE_INST|OCL3344:OCL3344|HNR3530*
To:                 *ETHERCAT_IPCORE_V2:ethercat_0|ETHERCAT_CORE:\CORE_INST:CORE_INST|OCL3344:OCL3344|BNR3543.ACL2024*
Assignemt Name:     Global Signal
Value:              Off
Enabled:            Yes
```

2.4.1.9 Qsys with EtherCAT IP Core V2.x.x: Avalon read error

With Qsys, a 32 bit read access to the EtherCAT IP Core might have gaps between the bytes. The prefetch feature (*Data width of smallest Avalon Master* is set to 2 or 4 byte), which is available in EtherCAT IP Core versions before V3.0.0, does not support these gaps, read errors occur as a consequence.

Solution
Set the configuration option *Data width of smallest Avalon Master* to 1 byte, prefetch is disabled then.

2.4.2 Solved

2.4.2.1 MAX10 with EtherCAT IP Core V2.4.0 – V3.0.10 Patch 1: Some configuration modes are not possible with EtherCAT IP Core memory initialization

The EtherCAT IP core uses memory initialization for the User RAM (0x0F80:0x0FFF), but only for debugging purposes (display selected ESC features). This memory initialization prevents the usage of some MAX10 configuration modes which do not support memory initialization.

Solution
A modified EtherCAT IP Core without memory initialization is available upon request.

2.4.2.2 Quartus Prime Standard 16.0.0: DE2-115 MII/RGMII example designs: ALTPLL cannot be generated

Generating the Qsys fails with an error caused by the ALTPLL:

```
Error (12252): Altpll_0: Port phasecounterselect has width 4 in TCL, but 3 in the design file
```

This issue is caused by Quartus, which incorrectly expects a signal width of 4 for the unused port *phasecounterselect*. This issue has already been reported to Intel.

Solution
Update to Quartus Prime Standard 16.0.1 (partially solves the issue). An intermediate solution is to extend the ALTPLL TCL file located in

<Quartus installation folder>\16.0\ip\altera\sopc_builder_ip\altera_avalon_altpll\altera_avalon_altpll_hw.tcl

with an additional condition for the device family (row 339 ff.):

```
set device_family [get_parameter_value INTENDED_DEVICE_FAMILY]
if { $device_family == "MAX10" || $device_family == "MAX 10" || $device_family == "Cyclone IV E"} {
      set phasectrsel_width 3
} else {
     set phasectrsel_width 4
}
```

2.4.2.3 Quartus 14.0: EtherCAT IP Core V3.0.0-V3.0.6: DE2-115 example designs are not working

The DE2-115 example designs are not working when synthesized using Quartus 14.0 (no communication, no display). The reason is that the PLL remains in reset state, because the areset input pin is not connected. Quartus 14.0 seems to use a different input value for this signal, causing the reset to be active all the time.

Solution
Open Qsys, export the altpll_0 module’s areset_conduit signal, and connect it to GND. The warning regarding this conduit disappears.

2.5 MD5 Checksums

Table 7: MD5 Checksums

IP Core version	Installation file	EtherCAT_IPCore.vhd	EtherCAT_IPCore_TOP.vhd	EtherCAT_IPCore.ocp
1.0.0	Linux: Windows Setup: Solaris:	D1D0C450B8F043D38C2277F0D92EE941 12AF6DE282CAACED8ACDCA5FE66906EE 22569688EAB18E96FC706E755D4746B2	2B587DF40D9CC6E9138122B9CC541C50	-	-
1.1.0	Linux: Windows Setup: Solaris:	D38CF74CDE42193CDC851336E4D48134 395215C741F9A62FED01B22B97988B79 173BB13DA241EDE44ECD213AE144D7EF	886CF05AC182CFDC03DAED1431C1CEDB	-	4CD1C6E5BAAC6B959E3A220A6A60E4BE
1.1.1	Linux: Windows Setup: Solaris:	324D9BE05EB0D1A2C5796845B008D2B6 776C9257CB2A454391B61B7CD38AE8F4 7FC91870F69AEAE49AB2A2FD97E5A2F5	E845C4FC91D3316E2B028129937E875C	-	DC2FA3CA77D7EE1212D0F3D147AB0C30
2.0.0	Linux: Windows Setup: Solaris:	B3E095764023F4EF6C399B16AC213AC8 EDCB8A9C51030947E5FA1B2C2B6D7510 242BD90B56437E61E369B0DCFD7A0563	0315BEF8034EE4E5BC2B17AA0EF7A28B	-	90EB2FF0057380F9A815F7E7180E112C
2.2.0	Linux: Windows Setup: Solaris:	069D2FF2D89BA4AF95E6C09DE30F3A29 17B6ABBFC9A2BA37D120FC0E41E6EB51 1D06CCE0DC542CDB6C3185E79DDFC24C	74BBDCE9FABFC8C86785A26CD03F3F79	-	7846AEC73A78481D25635572D7747278
2.2.1	Linux: Windows Setup: Solaris:	C4566CF9983B4E7AA1366168D0F36124 364B0C44866CDBF7B57E738EF3C9DF4F 96D5D8CAC8B843725F60D996CE94B6EE	60DB043F690F5FEFF4E3BADD8FBD9CA6	-	46F3936B35684565055D7EC119C92617
2.3.0	Linux: Windows Setup: Solaris:	60610D11ED5D8B2A711FACFFEE658731 14EAF65D006D79CD3187C838BB17303D 946855E035A0FE774E68BD06E0581EBA	A04286DC7280A99D46F24234BA0FAD4D	-	2B31E75D24AB4A3E15EBD1CEB71CF0C7
2.3.1	Linux: Windows Setup: Solaris:	41883CEF9C07A98A5D3DB022BE265438 CCA866485E1106A7B55C4E3011A38A4E C5470241D9AEA775761085B1A18B3DB5	5E64B409234AF3FAC47479912C717A32	-	5EDCFE8B16A867F76AA3BE50957BAFDF
2.3.2	Linux: Windows Setup: Solaris:	12887B6A71603E94A5288210453A5A76 AF304B9200F4D96F890AFC90B8249C54 ADF1F4F197A0BE62BE84A742221239A4	CBC425E967A12E11E2C8FE9A5D86405B	-	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0	Linux: Windows Setup:	6A5CCAD8D9960C0EB502B9AA3FB738E8 E67ED1EF67A6F16B1B49E138D57FBB67	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0 Patch 1	Linux: Windows Setup:	AF3BE8493B5988B8296345FDCCC25858 E10582BFE77B1928E85D1352439AEA6F	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0 Patch 2	Linux: Windows Setup:	6D4951038B8BCF5C706463DE347AFC99 2851B75D8B9E8265EDC19CD516519B10	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0 Patch 3	Linux: Windows Setup:	1E14526833B2611B76BD05651DBC485B 5560ED46EC0924654F5DBB3AA9BACD92	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0 Patch 4	Linux: Windows Setup:	437B368ED2DA42ACA8D5D95BFA798D9A 9D91A011346324CD25269422E84B5EBD	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.0 Patch 5	Linux: Windows Setup:	B1B335CB6EDC0932FB8B441C2FDDF888 7BD8EEF74B8BA11DA88832351506090C	2A8778F0A8DAAEAE91DB5A3048873885	337A37409F0E1E52E79CD85409239F80	0947B2C6D54A35A288ADBA2615AB0B5D
2.4.3	Linux: Windows Setup:	8488E660257A22863F870CABAAA48EB5 567EA5895550BD17E58AB2D164CF2A07	C01FE074B35C2F1AC80DB2F01F6C103B	627B7F77F22780F5BD675F5840830B08	A9C68DF57265D6C83837E1514D1B17C2
2.4.4	Linux: Windows Setup:	C5DD598B195B6F61C3229F901979739A 3890A9DA4CB2144C3594E5CE9A4296A2	441C076EA8333C9496A0EFBB37E2C3A7	84C952B1442AE1935E1DE573E4CB95D8	A9C68DF57265D6C83837E1514D1B17C2
2.4.4 Patch 1	Linux: Windows Setup:	465985CD8FBA1017C25956DA25855F4B C9EFB33E6F063DCA80A48EB9F7F63781	441C076EA8333C9496A0EFBB37E2C3A7	84C952B1442AE1935E1DE573E4CB95D8	A9C68DF57265D6C83837E1514D1B17C2

IP Core version	Installation file		EtherCAT_IPCore.vhd	EtherCAT_IPCore_TOP.vhd	EtherCAT_IPCore.ocp
3.0.0	Linux: Windows Setup:	388B42C9FBEA8ECCD342B56842E715C8 617548350798372391D46E44FAFDFB0F	ADA5C6DE4BDCA4A4F0E09DE36E0DC88E	9BAB935BC465767E19AED2C595239DA2	1099F8C3073EC879C36F8BD1AF7DB788
3.0.1	Linux: Windows Setup:	9513ED2AB809654362C6EE1A32F53165 FBC35EBA4251824C37CE629FF7BA76C3	2367AEC85AFD1210301D5A60A083C725	8C2FA562028D97B0A9A50389F3586B36	1099F8C3073EC879C36F8BD1AF7DB788
3.0.2	Linux: Windows Setup:	44AEAE95BA804D8198554A4C5DE7D225 7091E7B412794DEDF279571492836743	0E3F449BD673A9419156968406BDD1C8	3E8C73383C020EDA07D42DD2F1753C1D	1099F8C3073EC879C36F8BD1AF7DB788
3.0.5	Linux: Windows Setup:	2D9F984A740026E25894B20D778FBE2B DDACF189B3DA7B5B07DE0A48C318B60C	C9B95CDEF4617A23B5B9FA2420174003	FF01D708A668A75FC03099AC8CA2374B	1099F8C3073EC879C36F8BD1AF7DB788
3.0.6	Linux: Windows Setup:	3A4CB986424EB7FFE9C2135189B62B5F E8174A0A7659F1A4E065916AF621BC7A	6ED4389E58DD2495219E67A466AF4AB3	2F5C0B16572BD475F67BAEA1B279DE88	C29B6ED1F91E5A438D91EC02D44296FB
3.0.9	Linux: Windows Setup:	46E52598159846B542CB1C17EAA4E54A F03912C7B878168636D51B2727A911F4	AE0072D4778EB09104021E0B658A3973	3113D9903B1FA03366A24979A63BF0F3	42AF256EA5C2A0AB2A2C212CE3110027
3.0.9 Patch 1	Linux: Windows Setup:	177794354C51BCCFDA78CE0359D85DD2 E52F9038DA2A66AAE763C8F72353B4C9	42C87C693592048F97CE739CE7069F9D	4EC1B9349A7E4835745F66192AF64E22	42AF256EA5C2A0AB2A2C212CE3110027
3.0.10	Linux: Windows Setup:	D26E85BC55750E12095E9D805D9563A8 821D6D5C9613E7F2938AEA60C576D7E7	2271C94E27FD58877D7201E65907BFBB	232B26C1A1CE1581864D259C083CAC61	42AF256EA5C2A0AB2A2C212CE3110027
3.0.10 Patch 1	Linux: Windows Setup:	97BEACAC5B513FC537243A89DD29BA77 BBF3646F63CA716F228A638F1269A980	B795DAFBF89EEAF6ECBE34E1E19BDECB	82DCDFF5AE4F029518E9C4ACA12841DE	42AF256EA5C2A0AB2A2C212CE3110027
3.0.10 Patch 2	Linux: Windows Setup:	7CB3223F3C342840FC0995EEECB1B55C 2D6F92E7F17344D67F56B847C4BE76DB	9B9AD989A0B28E03454B39F3EB8CF2D3	777330A73D5449DABD2E8939F4700F35	42AF256EA5C2A0AB2A2C212CE3110027

3 EtherCAT IP Core for Xilinx FPGAs

3.1 Vivado design tool compatibility

Table 8: EtherCAT IP Core for Xilinx FPGAs compatibility with Xilinx Vivado designflow tools

IP Core version	Release date	2012.3 - 2012.4	2013.1	2013.2	2013.3 - 2013.4	2014.1	2014.2	2014.3	2014.4	2015.1	2015.2 - 2015.4	2016.1 - 2016.4	2017.1 - 2017.4	2018.1 - 2018.3	2019.1
1.01b	1/2007														
2.00a	8/2007														
2.02a	6/2008														
2.03a	12/2009														
2.03b	2/2010														
2.03c	3/2010														
2.03d	6/2010														
2.04a	3/2011														
2.04a Patch 1	12/2015														
2.04d Patch 1	7/2013	-	●	●	●										
2.04e Patch 1	1/2015	-	●	●	●	●	●	●							
3.00c	5/2013	-	●	●	●										
3.00c Patch 2	9/2013	-	●	●	●										
3.00f	2/2014	-	●	●	●	●									
3.00g	4/2014		●	●	●	●	●								
3.00j	9/2014		●	●	●	●	●	●	○						
3.00j Patch 1	11/2014		●	●	●	●	●	●	○						
3.00k	1/2015						●	●	○	○	●	●	●	●	●
3.00k Patch 2	5/2019						●	●	○	○	●	●	●	●	●

Table 9: Tool compatibility legend

Symbol	Description
●	Compatible (maybe solved issues with example designs)
○	Compatible, issues with some example designs
-	Incompatible
	Not tested

3.2 ISE design tool compatibility

Table 10: EtherCAT IP Core for Xilinx FPGAs compatibility with Xilinx ISE designflow tools

IP Core version	Release date	Tool compatibility			Version compatibility								
		ISE	EDK	PlanAhead	ISE 8.2.3 EDK 8.2.2	ISE 9.2.4 EDK 9.2.2	10.1.1	11.5	12.1 - 12.3	12.4	13.1	13.2 - 14.2	14.3 - 14.7
1.01b	1/2007	●	●	-	●	●							
2.00a	8/2007	●	●	-	●	●							
2.02a	6/2008	●	●	-	●	●	●						
2.03a	12/2009	●	●	-	-	-	-	●					
2.03b	2/2010	●	●	-	-	-	-	●					
2.03c	3/2010	●	●	-	-	-	-	●					
2.03d	6/2010	●	●	-	-	-	-	-	●	○	○	●	
2.04a	3/2011	●	●	-	-	-	-	-	-	●	●	●	
2.04a Patch 1	12/2015	●	●	-	-	-	-	-	-	●	●	●	
2.04d Patch 1	7/2013	●	●	●	-	-	-	-	-	●	●	●	●
2.04e Patch 1	1/2015	●	●	●	-	-	-	-	-	●	●	●	●
3.00c	5/2013	●	●	●	-	-	-	-	-				●
3.00c Patch 2	9/2013	●	●	●	-	-	-	-	-				●
3.00f	2/2014	●	●	●	-	-	-	-	-				●
3.00g	4/2014	●	●	●	-	-	-	-	-				●
3.00j	9/2014	●	●	●	-	-	-	-	-				●
3.00j Patch 1	11/2014	●	●	●	-	-	-	-	-				●
3.00k	1/2015	●	●	●	-	-	-	-	-				●
3.00k Patch 2	5/2019	●	●	●	-	-	-	-	-				●

Table 11: Tool compatibility legend

Symbol	Description
●	Compatible
○	Compatible, issues with some example designs
-	Incompatible
	Not tested

3.3 FPGA device compatibility

Table 12: EtherCAT IP Core for Xilinx FPGAs compatibility with Xilinx FPGAs

IP Core version	Spartan -3/ -3E	Spartan -3A/ -3AN/ -3AN DSP	Spartan -6	Spartan -7	Virtex -II/ -II Pro/ -II Pro X	Virtex -4	Virtex -5	Virtex -6	Virtex -7	Virtex Ultra-Scale	Virtex Ultra-Scale+	Kintex -7/ -7 L	Kintex Ultra-Scale	Kintex Ultra-Scale+	Artix -7/ -7 L	Zynq-7000	Zynq Ultra-Scale+
1.01b	●	●			●	●	●										
2.00a	●	●			●	●	●										
2.02a	●	●			●	●	●										
2.03a	●	●	●			●	●	●									
2.03b	●	●	●			●	●	●									
2.03c	●	●	●			●	●	●									
2.03d	●	●	●			●	●	●									
2.04a	●	●	●			●	●	●	●			●			●	●	
2.04a Patch 1	●	●	●			●	●	●	●			●			●	●	
2.04d Patch 1	●	●	●			●	●	●	●			●			●	●	
2.04e Patch 1	●	●	●			●	●	●	●	●		●	●		●	●	
3.00c	-	-	●		-	-	-	●	●			●			●	●	
3.00c Patch 2	-	-	●		-	-	-	●	●			●			●	●	
3.00f	-	-	●		-	-	-	●	●			●			●	●	
3.00g	-	-	●		-	-	-	●	●	●		●	●		●	●	
3.00j	-	-	●		-	-	-	●	●	●		●	●		●	●	
3.00j Patch 1	-	-	●		-	-	-	●	●	●		●	●		●	●	
3.00k	-	-	●		-	-	-	●	●	●	●	●	●	●	●	●	
3.00k Patch 2	-	-	●	●	-	-	-	●	●	●	●	●	●	●	●	●	●

Table 13: FPGA compatibility legend

Symbol	Description
●	Compatible
-	Incompatible
	Not tested

3.4 Known Vivado Designflow Issues

3.4.1 Active (at least until Vivado 2019.1)

3.4.1.1 Vivado 2016.1 – today: Synthesis fails with an RTL assertion error

Since Vivado 2016.1, assertions can be enabled again – this feature was removed in Vivado 2015.1. Nevertheless, Vivado 2016.1 does not handle assertions like it did before 2015.1, which results in synthesis failing for the EtherCAT IP Core, which intentionally uses assertions. The problem is that Vivado 2016.1 evaluates assertions without taking care of the context in which they are used in the source code – e.g., surrounding conditional expressions. This causes assertion errors which are actually not applicable; the Vivado versions before 2015.1 calculated them correctly. Please refer to chapter 3.4.1.3 for more details.

Message:

```
[Synth 8-63] RTL assertion: "DIGITAL IO PDI CONFIGURED WITHOUT ENABLING DIGITAL IO SOURCE FEATURE."
```

Solution

Disable the *-assert* switch in *synth_design* since Vivado 2016.1, or disable assertion messages since Vivado 2015.3:

```
set_param synth.elaboration.rodinMoreOptions {rt::set_parameter ignoreVhdlAssertStmts true}
```

Alternatively, use a Vivado version before 2015.1.

3.4.1.2 Vivado 2016.4 – today: Resource consumption too high

The EtherCAT IP Core has a higher resource consumption with synthesis and implementation strategies from older Vivado versions.

Solution

The following settings have been found advantageous:

- Always use the latest synthesis and implementation strategies.
- Use the synthesis strategy *AreaOptimized_High*
- Use the implementation strategy *Area_exploreSequential*

3.4.1.3 Vivado 2015.1 – today: Assertions and report messages are not evaluated

Vivado 2015.1 and later do not evaluate assertions and report messages anymore, even if they are based on constants or generics. As a result, the messages shown by the EtherCAT IP core source code (text configuration box) are no longer visible. Additionally, assertions resulting in errors or failures are no longer causing the synthesis to fail, thus, illegal configurations might become synthesized to dysfunctional logic. This is unlikely to happen since the IP core configuration tool prevents illegal configurations, but manual changes might cause them.

Message:

```
[Synth 8-312] ignoring unsynthesizable construct: assertion statement ["<...>/EtherCAT_IPCore.vhd":44812]
```

Solution

Use a previous Vivado version.

Although it is possible to re-enable assertions and report messages by issuing TCL commands since Vivado 2015.3, the EtherCAT IP Core until V3.00k cannot be synthesized with messages enabled. This is caused by the newer Vivado versions, which use a different kind of evaluation compared with Vivado 2014.4 and before. This results in assertion errors, which are not intended by the EtherCAT IP Core. Please refer to chapter 3.4.1.1 for more details.

Enable assertion messages since Vivado 2015.3:

```
set_param synth.elaboration.rodinMoreOptions {rt::set_parameter ignoreVhdlAssertStmts false}
```

Since Vivado 2016.1, the *-assert* switch in *synth_design* can be used instead. Vivado 2016.2 ignores the *-assert* switch when an XDC file is used for synthesis, you can use the above setting instead, which is still working (or disable the XDC for synthesis only).

The EtherCAT IP core uses only few report messages, you can enable report messages since Vivado 2015.3:

```
set_param synth.elaboration.rodinMoreOptions {rt::set_parameter ignoreVhdlReportStmts false}
```

3.4.1.4 Vivado 2015.1 – today: ZC702_AXI_VIVADO example design constraints fail

Since Vivado 2015, clock signals are renamed inside the Zynq block design. This causes some constraints to fail with such an error message:

```
[Vivado 12-1387] No valid object(s) found for set_false_path constraint with option '-to [get_clocks CLKOUT1]'.
```

Solution

In the *ZC702.xdc* constraint file, rename all clock references using

```
CLKOUT1
```

to the new name

```
clk_out2_ZYNC_BLOCK_clk_wiz_0_0
```

3.4.1.5 Vivado 2015.1 – today: ZC702_AXI_VIVADO example design has timing failures

Vivado introduces BUFGs on the MII_TX_CLK wires, which causes timing failures for paths from ISMNET1_PHY1_MII_TX_CLK and ISMNET1_PHY2_MII_TX_CLK. In addition, it causes suboptimal placement of other signals in the IP core, which may cause additional timing violations.

Solution

Prevent inferring a BUFG by adding this line to the *ZC702.xdc* constraints file:

```
set_property CLOCK_BUFFER_TYPE NONE [get_ports ISMNET1_PHY*_MII_TX_CLK]
```

If Vivado infers a BUFG anyway (experienced with Vivado 2015.3), try relaxing the constraints (3 ns instead of 5 ns):

```
set_input_delay -clock ISMNET1_CARRIER_25MHZ_S 3.000 [get_ports ISMNET1_PHY1_MII_TX_CLK]
set_input_delay -clock ISMNET1_CARRIER_25MHZ_S 3.000 [get_ports ISMNET1_PHY2_MII_TX_CLK]
```

3.4.1.6 Vivado 2014.1 – today: Upgrading ZC702_AXI_VIVADO example results in warning on port differences

When upgrading the ZC702_AXI_VIVADO example design to a newer Vivado version, a warning may be issued regarding port differences:

```
[IP_Flow 19-3298] Detected external port differences while upgrading IP 'ZYNC_BLOCK_processing_system7_0_0'. These changes may impact your design.
```

These warnings can be ignored since the different port signals are not used by the example design.

3.4.1.7 Vivado 2013.1 – today: EtherCAT IP Core is not part of the IP Catalog

The EtherCAT IP Core is not part of the IP Catalog, because the configuration options required by the EtherCAT IP Core are currently not supported by the IP Packager/IP Catalog. Additionally, Vivado does not support Spartan-6 FPGA devices, so ISE support is still required. Maintaining two types of configuration (IPCore_Config tool for ISE and IP Catalog for Vivado) is currently not planned.

Solution

Use The IPCore_Config tool to configure the EtherCAT IP Core, then add it to your project. You can either instantiate the EtherCAT IP Core wrapper directly, or you can use the IP Packager to add it to the IP Catalog (without any configuration options). The last solution allows for automatically connecting the AXI interface.

3.4.2 Solved

3.4.2.1 Vivado 2016.1 – 2016.3: Resource consumption too high, synthesis time is too high

The EtherCAT IP Core has a higher resource consumption then before, when used with Vivado 2016.1 and later (about 30%); it also has a higher resource consumption compared with ISE results. The higher resource consumption is highly related to how Vivado handles protected IP.

The resource consumption reported after synthesis is by factors too high, because Vivado synthesizes every possible functionality of the EtherCAT IP core, although typically many functions are disabled by the configuration. This increases the resource consumption unnecessarily. After implementation, the resource consumption is fairly reasonable.

The synthesis time is too high because Vivado synthesizes every possible functionality of the EtherCAT IP core, although typically many functions are disabled by the configuration. This increases synthesis times unnecessarily.

Solution

Use the following TCL command. It should be issued before synthesis to reduce the resource consumption and compile times:

```
set_param synth.elaboration.rodinMoreOptions "set rt::extractNetlistGenomes false; rt::set_parameter advancedConstPropAcrossHier true"
```

Additionally, the following settings have been found advantageous:

- Always use the latest synthesis and implementation strategies.
- Use the synthesis strategy *AreaOptimized_High*
- Use the implementation strategy *Area_exploreSequential*

3.4.2.2 Vivado 2015.3 – 2015.4: EXCEPTION_ACCESS_VIOLATION during synthesis because of tristate drivers (crash)

Vivado 2015.3/2015.4 crashes during synthesis with an EXCEPTION_ACCESS_VIOLATION, if the EtherCAT IP Core is not part of a block design. This issue is related to the tristate drivers, which are optionally available inside the EtherCAT IP Core. The crash occurs because of the pure existence of the drivers in the source code, regardless of them being enabled or disabled.

Solution
This issue is fixed in Vivado 2016.1. There are two possible workarounds for 2015.x:
Set a “DONT_TOUCH” attribute on the EtherCAT IP Core instance (e.g. *axi_ethercat_user_i* in the ZC702 example design). Vivado does this automatically for IP integrated in a block design, so the issue does not appear in this case.

```
attribute DONT_TOUCH : string;
attribute DONT_TOUCH of axi_ethercat_user_i: label is "TRUE";
```

Alternatively, execute the following TCL command in the Vivado GUI TCL Console and disable all internal tristate drivers inside the IP core:

```
set_param synth.elaboration.rodinMoreOptions "rt::set_parameter enableTristateBubbleUp false"
```

Caution: Using internal tristate drivers with this option causes Vivado to implement tristate drivers wrong (refer to chapter 3.4.2.7 for more information on this older issue).

3.4.2.3 Vivado 2015.1 – 2015.4: Resource consumption too high, synthesis time is too high

The EtherCAT IP Core has a higher resource consumption then before, when used with Vivado 2015.1 – 2015.4 (about 30%); it also has a higher resource consumption compared with ISE results. The higher resource consumption is highly related to how Vivado handles protected IP.

The resource consumption reported after synthesis is by factors too high, because Vivado synthesizes every possible functionality of the EtherCAT IP core, although typically many functions are disabled by the configuration. This increases the resource consumption unnecessarily. After implementation, the resource consumption is fairly reasonable.

The synthesis time is too high because Vivado synthesizes every possible functionality of the EtherCAT IP core, although typically many functions are disabled by the configuration. This increases synthesis times unnecessarily.

Solution

Use Vivado 2016.1 or newer. For Vivado 2015, the following settings have been found advantageous:

- Always use the latest synthesis and implementation strategies.
- Use the synthesis strategy *AreaOptimized_High*
- Use the implementation strategy *Area_exploreSequential*

Settings for Vivado 2015.1

The following synthesis settings have been suggested by Xilinx, and we found them useful until Vivado 2015.1:

- Option -control_set_opt_threshold 0
- Option -fsm_extraction sequential
- Use this TCL command:

```
set_param synth.elaboration.rodinMoreOptions "set rt::extractNetlistGenomes false"
```

3.4.2.4 Vivado 2015.1: EtherCAT IP cores outside the Zynq block design fails in SDK (ZC702_AXI_VIVADO example design also affected)

Vivado 2015.1 generates an *xparameters.h* file without base address defines if the EtherCAT IP core (and probably any other AXI slave) is connected externally to the Zynq block design. This causes a compile error for the example software in the SDK:

```
../src/EtherCAT_TestApp.c:18:28: error: 'XPAR_ETHERCAT_AXI_BASEADDR' undeclared (first use in this function)
```

Solution

Please use Vivado 2015.2 (and newer) or change the base address reference in *EtherCAT_TestApp.c* (line 18) to refer to the actual base address instead of the *XPAR_ETHERCAT_AXI_BASEADDR* define:

```
#define ETHERCAT_BASEADDR            0x43C00000
```

3.4.2.5 Vivado 2014.4: EtherCAT IP cores outside the Zynq block design fails in SDK (ZC702_AXI_VIVADO example design also affected)

Vivado 2014.4 generates a wrong hardware description file if the EtherCAT IP core (and probably any other AXI slave) is connected externally to the Zynq block design. This causes the SDK to fail in generating the example software with such an error message:

```
[Hsi 55-1464] Hardware instance __EMPTY__ not found in the design
```

This issue is partially fixed in Vivado 2015.1. A similar answer record (AR63036) can be found on the Xilinx website.

Solution
Please use Vivado 2015.1 (and newer) or 2014.3 to implement this example design and other designs which have the EtherCAT IP Core connected externally to the Zynq block design.

3.4.2.6 Vivado 2014.3: EXCEPTION_ACCESS_VIOLATION during synthesis (crash)

Vivado 2014.3 may stop during Technology Mapping with an access violation occasionally. Just repeat the synthesis run again and again until it succeeds (1-6 iterations until success have been seen). This issue is already approved by Xilinx and should be fixed in 2014.3 Update 1.

The Vivado console stops with these messages:

```
[...]
---------------------------------------------------------------------------------
Start Technology Mapping
---------------------------------------------------------------------------------
Abnormal program termination (EXCEPTION_ACCESS_VIOLATION)
Please check '<project name>.runs/synth_1/hs_err_pid6336.log' for details
```

The mentioned log file contains no further details except for the dump file:

```
#
# An unexpected error has occurred (EXCEPTION_ACCESS_VIOLATION)
#
Stack:
no stack trace available, please use hs_err_<pid>.dmp instead.
```

Solution
Run the synthesis again several times until it succeeds or update Vivado.

3.4.2.7 Vivado 2014.2 – 2014.3: Resource consumption too high

The EtherCAT IP Core has a higher resource consumption compared with ISE results. The higher resource consumption is highly related to how Vivado handles protected IP.

Solution
Use Vivado 2016.1 or newer. Alternatively, the following synthesis settings have been suggested by Xilinx, and we found them useful until Vivado 2015.1:

- Option -control_set_opt_threshold 0
- Option -fsm_extraction sequential
- Use this TCL command:

```
set_param synth.elaboration.rodinMoreOptions "set rt::extractNetlistGenomes false"
```

3.4.2.8 Vivado 2014.1 – 2014.3: Tri-state drivers inside EtherCAT IP Core

If the EtherCAT IP Core is configured to have internal tri-state buffers (e.g. for EEPROM/MI), Vivado 2014.2 (and maybe others) does not correctly implement the tristate drivers. Instead of a tristate output, a push-pull output is implemented, which might not even toggle. This issue is approved by Xilinx, it should be solved in 2014.3.

Solution
Execute the following Tcl command in the Vivado GUI Tcl Console:

```
set_param synth.elaboration.rodinMoreOptions "rt::set_parameter enableTristateBubbleUp 1"
```

3.4.2.9 Vivado 2014.1 – 2014.2: The resource consumption of the EtherCAT IP Core is too high

The resource consumption of the EtherCAT IP Core is about 30% higher when it is synthesized with Vivado instead of ISE. This issue is approved by Xilinx, it is targeted to be solved in 2014.3.

Solution
Refer to Answer Record 61518 for a Vivado 2014.2 patch (this is not publicly available on the website). After applying the patch, the resource consumption is still higher, but much more acceptable.

3.4.2.10 Vivado 2013.2 – 2014.1: The ZC702 AXI Vivado example design is not synthesizable

The ZC702 AXI Vivado example design was developed for Vivado 2014.2 and later, and it is incompatible with previous Vivado versions.

Solution
Use Vivado 2014.2 with the patch or a later Vivado version.

3.4.2.11 Vivado 2013.1 – 2013.2 with EtherCAT IP Core until V3.00f: IP License issue

The license string used by the EtherCAT IP Core before V3.00g is not accepted by Vivado 2013.2. Note that the license key style change was not communicated to Beckhoff.

Solution
- use Vivado 2013.3 or later, or
- use EtherCAT IP Core V3.00g or later, or
- refer to Answer Record 56630 for a Vivado 2013.2 patch required to use the EtherCAT IP Core before V3.00g

3.5 Known ISE/EDK/PlanAhead 14.7 Designflow Issues

3.5.1 ISE: Crash in libSecurity_FNP.dll

After installing both ISE 14.7 and Vivado 2014.1 on a Microsoft Windows 7 64 bit operating system, a crash occurred while synthesizing the EtherCAT IP Core using ISE 14.7. The crash was caused by libSecurity_FNP.dll. This issue is approved by Xilinx, but there will not be any future version of ISE.

Solution
Refer to Answer Record 59851 for an ISE 14.7 patch.

3.5.2 ISE/EDK/PlanAhead: Additional BUFG inserted

Under certain circumstances, the tools insert additional BUFGs into the clock signals to the EtherCAT IP Core (CLK25, CLK100), or the BUFGs are not placed at optimal sites. This results in an error message suggesting to add a CLOCK_DEDICATED_ROUTE=FALSE constraint to the clocks. This issue occurs especially when the MAP option “Global optimization” is different from OFF. This issue is reported to Xilinx and present at least until ISE 14.5.

Solution
Either set Global optimization to OFF or add the CLOCK_DEDICATED_ROUTE=FALSE constraint.

3.5.3 ISE/EDK/PlanAhead: CLOCK_DEDICATED_ROUTE=FALSE constraint required

The PDI_SPI_SEL, PDI_SPI_DI and MII_RX_CLK0/1/2 signals are used as clock inputs for a few registers at low speed. Since these signals are not placed on dedicated clock inputs – which is not required –, Xilinx ISE sometimes issues an error, which needs to be suppressed.

Solution
Add an additional constraint for the signals causing the error message (or uncomment the appropriate lines in the example UCF files):

```
CLOCK_DEDICATED_ROUTE = FALSE
```

3.6 MD5 Checksums

Table 14: MD5 Checksums

IP Core version	Installation file		EtherCAT_IPCore.vhd (Eval)	EtherCAT_IPCore.vhd (Full)
1.01b	Linux: Windows Setup:	1899C6CDCCEB00A797EF31A885AF01FB 1580A288BC62BC6D58B5CD78E35B763F	-	533F00370A4DDA694CFFB0884F066666
2.00a	Linux: Windows Setup:	AE3423B219360285B6629EAE47B792D1 6BFDB163E0327619CB8B3256A1B52484	-	1CE92D2A82477CD571FBE62A83EC8696
2.02a	Linux: Windows Setup:	2EBD592DE0F2AAC640A8FBE7A367BB05 9AEC59327D1CFB786E7B4634BACE50C4	-	E9F690D616EDB78A96F5F6FE47BE9541
2.03a	Linux: Windows Setup:	86C8B364FB3D6F8407402E03753943AA B3646E28A9B8F94337CF6C209F209BBD	B260421558D83119B81ECBB01BF378A8	4755938B42B2BF9421C9C3A1E5751C3F
2.03b	Linux: Windows Setup:	104352B71715E8678E285104F41ECAC9 F2D7DE753417AC8DCF9488A8A555F8DD	076F6A088AA5E5FAB1CDB4FF408D5D54	62119D0B7B67FF84E96657CAD61C874C
2.03c	Linux: Windows Setup:	93BE44E84E450359AB6F6D02600E7FB2 BFDBD06F80C18CA661297E28DF45A0C1	0D670C0438C58BEA3938D9093CCA331A	37BD82957C0A1E4AE8A46553E875E1EC
2.03d	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	3F1C84F62E83D8B2059A0BCD23B2BD14 FDDF7C665DCEE0D174D08EE854FD2F0A 7F8DB7BA5810B50F06994923A008C682 1ACFC3AFDA23BB0BEAE35ED2D6AD3111	A62C4BBD5F2BDF9ECDBDA009B3759F6D	3F3988B30802E9B36BF30963DD5CB59E
2.04a 2.04a Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	94D6FB1FB690A633B9BF2049834C09DC 0E64282A1A22582597E4D21FBE64D754 59FE9FF62FF39F07064ED6BEA6C50D46 F7C7025A1544C9B56F4F1135F5F97900	ABBD34279229A068E1FE64B85582DB41	28F5C5BD956F8F8CF0115D26BD6C5D87
2.04d	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	63DE895EC03DACD8EAA0DCAF23B10183 3C516DB3AB28F79653E0D5A9C97D1C16 5E49E95DC79FA932A1EA3144C1E45F11 7927711E29251D9FA7D18BEE9516EFE9	2A1A80A323BA65C77A539CAF0C5BAB03	3E80395BD532F7652053284E29F61826
2.04d Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	BCB113FEF3D2278AB0E6E9E7F1576F21 A53E92DA6A482B669C0F6C762B7B93C0 8B14F311183452CC8D83DE094D5AC478 9B0DD9AB57B24227AB3D8B0BA3DBBB99	2A1A80A323BA65C77A539CAF0C5BAB03	3E80395BD532F7652053284E29F61826
2.04e	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	6FC02511752456B2D8E2216737508EB2 A1DA05B331C62ADB83C18F20B23209E4 03281CD779E984FC5EC9D98D778FF287 D616B233DFADAE2A0829E5B08C7D9CD8	BCDA1ABA04E08A6E591E56561EF0266D	4C5FB7C82EFAB9542052C814FD2F220C
2.04e Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	EAEFE65D51C56BC45C14700100178AA0 E6BEA0859E692843108E2B963B3859E4 1F32564BB471D7606F4B576403A2E2E3 F581E224651ED03C8400BC627D5722D7	BCDA1ABA04E08A6E591E56561EF0266D	4C5FB7C82EFAB9542052C814FD2F220C
3.00c	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	C97D18FFF3C8849716C6A3D1393F4EC6 FF25A73EFE5A0C811136BCA92D93F8E9 21B12335D5B24120C0627346B1C5EEEB C441EE0172594E20EC56470A5CF95DAE	659F5EFBDEB242AE32E33E6D06BF0731	D37283329FEE5FAB345BE1768C9A8818
3.00c Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	D5FFF04441CA5D4D9B3F77664426387D 0827A2D98087D7DFE375C94D0932B013 AD48638BBF22ED5E15374D426B100B05 52A053701584F67A1C65CAC3C83E7201	659F5EFBDEB242AE32E33E6D06BF0731	D37283329FEE5FAB345BE1768C9A8818
3.00c Patch 2	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	0545AF7A1B46121F897266EE5136CEA9 C969D2A2AA6636BA3D3CB04BEE35B015 F307F89B2BBCF3021A65AA3766E20008 2F433D0E1803B8A92A7ACE5FB2EF7B7B	659F5EFBDEB242AE32E33E6D06BF0731	D37283329FEE5FAB345BE1768C9A8818

IP Core version	Installation file		EtherCAT_IPCore.vhd (Eval)	EtherCAT_IPCore.vhd (Full)
3.00f	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	294515E21668C0ADDA2C37597C0CDBA4 8BC5A1E2DD6609102545EE2CDC772AE7 28A438CD00ECAE4D77BA457F51190420 042B9D52D1C3AAC13B8F1BD01A6749A8	97EF0F962ACCDD0AC580151B7BD52F91	3B4A3C2C5B4E0970CA4CD42C3FF6DD86
3.00g	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	50534B6786BB923BBFC5F326FC8AF4CB EEC32E8C2A588D2214354F7BC05C99FA 3B2119B7EFB7B8A982AE74A9ACA229EB 5CD161DC71177D19A9DCFC741AAC6AEF	6C23218D87593192334EAC269E11B1E3	8DFBC11C6FB5057C043E9D9FE15E6BE8
3.00j	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	28E475E6677044BFE8598B793FE39B5E 333FF21F763A1223EC7484328DC6C5A5 1BEF71D6A0EC36E3B4C5D685C8E57C31 1E644A26DC77CF96D5603AEF42718DBA	15959C029DEC5E67FA84677A833DAC7F	E5F133A1AE76DEEE630C1AB6C553FCCF
3.00j Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	A8C6133FC4DB1911949F38A92E65B6D0 178CEF6BA83451CDF040E805DD872D2C 1D255128983C34252420AE4BF51B1D77 90FDED5A67FDDE0A195A7C4B1BE2DC23	88659137CABD12A9A669597432D6161E	4B5744B909BBA46E7FB2AEC229E86036
3.00k	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	8527D72AD00564400D63D36B7B1ED013 2AB585AABA2D0AB05F38F7C492CF708D DF7901A9B51EFC66EEF4FF4BF6B1FCAA 67D48CB3BC6B3234D26E530F3A29346A	1977EB97B234A713BB7DE32CF7DB1896	D232B7CB0DDBB982FD07E2E009BB3087
3.00k Patch 1	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	1D449671BF0ACD19A69699E44381F70F 91412378289AA3F0BA04F79D27E4D165 EBEA4E2E9129E3CE85B0E7D19184E077 2ACE077D3CBE73D132813CA8A372CAEB	20C2C65C00F2096D61BDABE542F24628	87E8BCC1F70D85C899EB90AB7D39A4FD
3.00k Patch 2	Linux Eval: Linux Full: Windows Setup Eval: Windows Setup Full:	4F854499CDED2A563BD6ECBCB2FDD234 CC65A10604A900A3CD373B87BFED2543 BAC15168AB5C2AE3626D2A592DCE11C6 6BD8FA509BFFDF08D18D64E9A75D2498	6D47775631D8F545F8DCBD35E66F791A	199FC1DBB74323290E1A6DBC7A9E9056

4 Appendix

4.1 Support and Service

Beckhoff and our partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

4.1.1 Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

4.2 Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 (0) 5246 963-0

Fax: +49 (0) 5246 963-198

E-mail: info@beckhoff.com

Web: www.beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- world-wide support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 (0) 5246 963-157

Fax: +49 (0) 5246 963-9157

E-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 (0) 5246 963-460

Fax: +49 (0) 5246 963-479

E-mail: service@beckhoff.com