TE0808 IBERT

Table of contents

Overview

Xilinx IBERT with TE0808 Starterkit (TEBF0808 Carrier).

Refer to http://trenz.org/te0808-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- TEBF0808
- PL IBERT
- Modified FSBL for Si5338 programming
- Special FSBL for QSPI Programming

Revision History

Date	Vivado	Project Built	Authors	Description
2020-09-29	2019.2	TE0808-test_board-vivado_2019.2-build_15_20200929070725.zip TE0808-test_board_noprebuilt-vivado_2019.2-build_15_20200929070740.zip	John Hartfiel	• new assembly variants
2020-03-25	2019.2	???	John Hartfiel	• 2019.2 initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround/Solution	To be fixed version
--	--	--	**--**

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2019.2	needed, Vivado is included into Vitis installation
PetaLinux	2019.2	needed
SI ClockBuilder Pro	---	optional

Software

Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.

Complete List is available on <design name>/board_files/*_board_files.csv

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
~~TE0808-ES1~~	es1_2gb	REV03|REV02	2GB	64MB	NA	NA	Not longer supported by vivado
~~TE0808-ES2~~	es2_2gb	REV04|REV03	2GB	64MB	NA	NA	Not longer supported by vivado
~~TE0808-2ES2~~	2es2_2gb	REV04|REV03	2GB	64MB	NA	NA	Not longer supported by vivado
TE0808-04-09EG-1EA	9eg_1e_2gb	REV04	2GB	64MB	NA	NA	
TE0808-04-09EG-1EB	9eg_1e_4gb	REV04	4GB	64MB	NA	NA	
TE0808-04-09EG-1ED	9eg_1e_4gb	REV04	4GB	64MB	NA	1 mm connectors	

TE0808-04-09EG-2IB	9eg_2i_4gb	REV04	4GB	64MB	NA	NA	
TE0808-04-15EG-1EB	15eg_1e_4gb	REV04	4GB	64MB	NA	NA	
TE0808-04-09EG-1EE	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-09EG-1EL	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-09EG-2IE	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-15EG-1EE	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-06EG-1EE	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-06EG-1E3	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-6GI21-L	6eg_2i_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-6GI21-A	6eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-6BI21-A	6eg_1i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9GI21-A	9eg_2i_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9BE21-A	9eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-6BE21-L	6eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-6BE21-A	6eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-9BE21-L	9eg_1e_4gb	REV04	4GB	128MB	NA	1 mm connectors	
TE0808-04-BBE21-A	15eg_1e_4gb	REV04	4GB	128MB	NA	NA	
TE0808-04-6BI21-X	6eg_1i_4gb	REV04	4GB	128MB	NA	NA	U41 replaced with schottky diodes
TE0808-05-6BE21-A	6eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-6BI21-D	6eg_1i_4gb	REV05	4GB	128MB	NA	1 mm connectors	SoC without encryption
TE0808-05-6BI21-X	6eg_1i_4gb	REV05	4GB	128MB	NA	NA	U41 replaced with schottky diodes
TE0808-05-6BI41-X	6eg_1i_8gb	REV05	8GB	128MB	NA	NA	U41 replaced with schottky diodes
TE0808-05-9BE21-A	9eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9BE21-L	9eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA
TE0808-05-9BI41-X	9eg_1i_8gb	REV05	8GB	128MB	NA	NA	U41 replaced with schottky diodes
TE0808-05-9GI21-A	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-9GI21-C	9eg_2i_4gb	REV05	4GB	128MB	NA	NA	SoC without encryption
TE0808-05-BBE21-A	15eg_1e_4gb	REV05	4GB	128MB	NA	NA	NA
TE0808-05-BBE21-L	15eg_1e_4gb	REV05	4GB	128MB	NA	1 mm connectors	NA

Hardware Modules

Note: Design contains also Board Part Files for TE0808 only configuration, this boart part files are not used for this reference design.

Design supports following carriers:

Carrier Model	Notes
TEBF0808	Used as reference carrier. **Important**: CPLD Firmware REV07 or newer is recommended

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
Heat Sink for the SoC	Important!
FMC Loopback Karte	Optional HW
SFP+ Loopback Adapter	
Firefly cable	loopback possible with second connector on the carrier

PCIe Card	Optional HW
SD card	with fat32 partiton

Additional Hardware

Content

For general structure and of the reference design, see Project Delivery - AMD devices

Design Sources

Type	Location	Notes
Vivado	<design name>/block_design <design name>/constraints <design name>/ip_lib <design name>/hdl	Vivado Project will be generated by TE Scripts
Vitis	<design name>/sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration

Design sources

Additional Sources

Type	Location	Notes
SI5345	<design name>/misc/Si5345	SI5345 Project with current PLL Configuration

Additional design sources

Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIN-File	*.bin	Flash Configuration File with Boot-Image (Zynq-FPGAs)
BIT-File	*.bit	FPGA (PL Part) Configuration File
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

Prebuilt files (only on ZIP with prebult content)

Download

Reference Design is only usable with the specified Vivado/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0808 "IBERT" Reference Design

Design Flow

> ⚠ Reference Design is available with and without prebuilt files. It's recommended to use TE prebuilt files for first lunch.

Trenz Electronic provides a tcl based built environment based on Xilinx Design Flow.

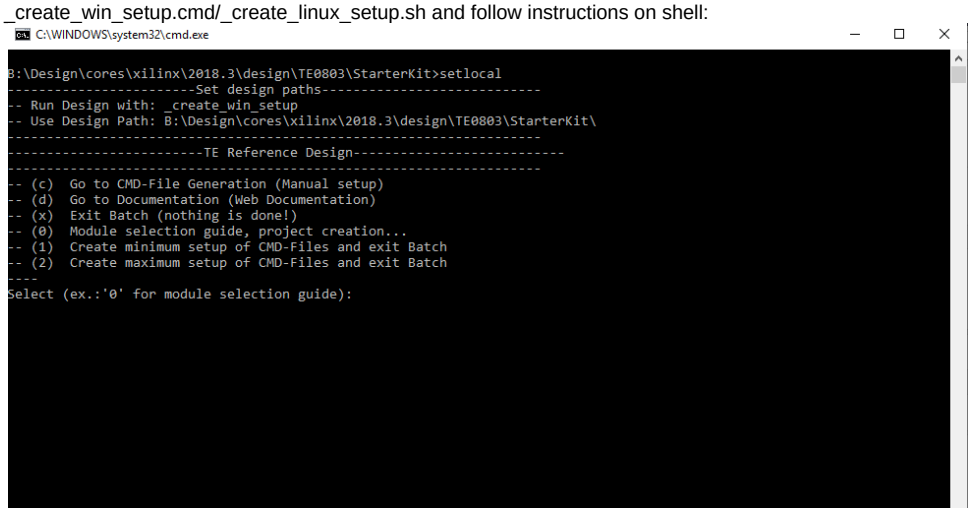
See also:

- AMD Development Tools#XilinxSoftware-BasicUserGuides
- Vivado Projects - TE Reference Design
- Project Delivery.

The Trenz Electronic FPGA Reference Designs are TCL-script based project. Command files for execution will be generated with "_create_win_setup.cmd" on Windows OS and "_create_linux_setup.sh" on Linux OS.

TE Scripts are only needed to generate the vivado project, all other additional steps are optional and can also executed by Xilinx Vivado/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

1. _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:



2. Press 0 and enter to start "Module Selection Guide"
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project (follow instruction of the product selection guide), settings file will be configured automatically during this process
    a. (optional for manual changes) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"
    Note: Select correct one, see TE Board Part Files
    Important: Use Board Part Files, which ends with *_tebf0808
5. Create XSA and export to prebuilt folder
    a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
    Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder
6. Generate Programming Files with Vitis
    a. Run on Vivado TCL: TE::sw_run_vitis -all
    Note: Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv"
    b. (alternative) Start SDK with Vivado GUI or start with TE Scripts on Vivado TCL: TE::sw_run_vitis
    Note: TCL scripts generate also platform project, this must be done manuelly in case GUI is used. See Vitis

Launch

For basic board setup, LEDs... see: TEBF0808 Getting Started

Programming

> ⚠ Check Module and Carrier TRMs for proper HW configuration before you try any design.

Xilinx documentation for programming and debugging: Vivado/SDK/SDSoC-Xilinx Software Programming and Debugging

Get prebuilt boot binaries

1. _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
    a. Select assembly version
    b. Validate selection
    c. Select Create and open delivery binary folder
    Note: Folder (<project foler>/_binaries_<Artikel Name>) with subfolder (boot_<app name>) for different applications will be generated

QSPI

Optional for Boot.bin on QSPI Flash and image.ub on SD.

1. Connect JTAG and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"
3. Type on Vivado TCL Console: TE::pr_program_flash_binfile -swapp hello_ibert
   Note: To program with SDK/Vivado GUI, use special FSBL (zynqmp_fsbl_flash) on setup
4. Set Boot Mode to QSPI-Boot
    a. Depends on Carrier, see carrier TRM.
    b. TEBF0808 change automatically the Boot Mode to SD, if SD is insered, optional CPLD Firmware without Boot Mode changing for mircoSD Slot is available on the download area

SD

1. Copy Boot.bin on SD-Card
    - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries
    - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
2. Set Boot Mode to SD-Boot.
    - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TE0808 StarterKit#Programming
2. Connect UART USB (JTAG XMOD)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
   Note: See TRM of the Carrier, which is used.
4. (Optional) Connect MGT loopback adapter
   Note: 1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD into OCM, 2. FSBL loads application from SD/QSPI into DDR.

Linux

1. Open Serial Console (e.g. putty)
    a. Speed: 115200
    b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)
2. Console: 'Hello IBERT (TE0808) (Loop: %i) * is running in endless loop

Vivado Hardware Manager

Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).

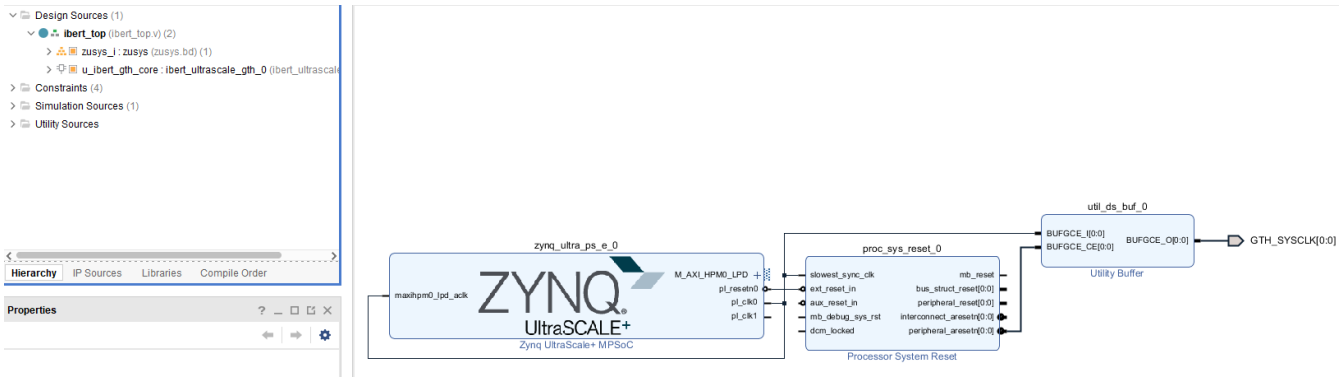
- Xilinx IBERT interface will be available

TODO picture
Vivado Hardware Manager

TODO mapping table

System Design - Vivado

Block Design



*Note: IBERT is used as RTL IP, TOP entity is modified version from Xilinx eIBERT example export
Block Design

PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD1	MIO
I2C0	MIO
UART0	MIO
GPIO0	MIO
SWDT0..1	

TTC0..3	

PS Interfaces

Constrains

Basic module constrains

_i_bitgen.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLNONE [current_design]
```

Design specific constrain

_i_example_ibert_ultrascale_gth_0.xdc

```
# file: ibert_ultrascale_gth_0.xdc
##############################################################################
##   ____  ____
##  /   /\/   /
## /___/  \  /    Vendor: Xilinx
## \   \   \/     Version : 2017.1
##  \   \         Application : IBERT Ultrascale
##  /   /         Filename : example_ip_ibert_ultrascale_gth_0.xdc
## /___/   /\
## \   \  /  \
##  \___\/\___\
##
##
##
## Generated by Xilinx IBERT
##************************************************************************
##
## TX/RX out clock clock constraints
##
# GT X0Y4
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[0].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[0].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y5
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[1].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[1].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y6
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[2].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[2].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y7
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[3].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
```

```
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[0].u_q/CH
[3].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y4
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[0].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[0].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y5
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[1].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[1].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y6
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[2].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[2].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y7
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[3].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[1].u_q/CH
[3].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y8
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[0].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[0].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y9
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[1].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[1].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y10
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[2].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[2].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y11
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[3].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[2].u_q/CH
[3].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y12
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[0].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[0].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y13
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[1].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[1].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y14
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[2].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[2].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
# GT X0Y15
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
[3].u_ch/u_gthe4_channel/RXOUTCLK}] -include_generated_clocks]
set_clock_groups -asynchronous -group [get_clocks -of_objects [get_pins {u_ibert_gth_core/inst/QUAD[3].u_q/CH
```

```
[3].u_ch/u_gthe4_channel/TXOUTCLK}] -include_generated_clocks]
```

_i_example_ibert_ultrascale_gth_0.xdc

```
# file: ibert_ultrascale_gth_0.xdc
##############################################################################
##   ____  ____
##  /   /\/   /
## /___/  \  /    Vendor: Xilinx
## \   \   \/     Version : 2012.3
##  \   \         Application : IBERT Ultrascale
##  /   /         Filename : example_ibert_ultrascale_gth_0.xdc
## /___/   /\
## \   \  /  \
##  \___\/\___\
##
##
##
## Generated by Xilinx IBERT 7Series
##************************************************************************
##
## Icon Constraints
##
create_clock -name D_CLK -period 10.0 [get_ports gth_sysclkp_i]
set_clock_groups -group [get_clocks D_CLK -include_generated_clocks] -asynchronous
set_property C_CLK_INPUT_FREQ_HZ 100000000 [get_debug_cores dbg_hub]
set_property C_ENABLE_CLK_DIVIDER true [get_debug_cores dbg_hub]

##gth_refclk lock constraints
##
set_property PACKAGE_PIN F25 [get_ports gth_refclk0p_i[0]]
set_property PACKAGE_PIN F26 [get_ports gth_refclk0n_i[0]]
set_property PACKAGE_PIN D25 [get_ports gth_refclk1p_i[0]]
set_property PACKAGE_PIN D26 [get_ports gth_refclk1n_i[0]]
set_property PACKAGE_PIN R8 [get_ports gth_refclk0p_i[1]]
set_property PACKAGE_PIN R7 [get_ports gth_refclk0n_i[1]]
set_property PACKAGE_PIN N8 [get_ports gth_refclk1p_i[1]]
set_property PACKAGE_PIN N7 [get_ports gth_refclk1n_i[1]]
set_property PACKAGE_PIN L8 [get_ports gth_refclk0p_i[2]]
set_property PACKAGE_PIN L7 [get_ports gth_refclk0n_i[2]]
set_property PACKAGE_PIN J8 [get_ports gth_refclk1p_i[2]]
set_property PACKAGE_PIN J7 [get_ports gth_refclk1n_i[2]]
set_property PACKAGE_PIN G8 [get_ports gth_refclk0p_i[3]]
set_property PACKAGE_PIN G7 [get_ports gth_refclk0n_i[3]]
set_property PACKAGE_PIN E8 [get_ports gth_refclk1p_i[3]]
set_property PACKAGE_PIN E7 [get_ports gth_refclk1n_i[3]]
##
## Refclk constraints
##
create_clock -name gth_refclk0_1 -period 8.0 [get_ports gth_refclk0p_i[0]]
create_clock -name gth_refclk1_1 -period 8.0 [get_ports gth_refclk1p_i[0]]
set_clock_groups -group [get_clocks gth_refclk0_1 -include_generated_clocks] -asynchronous
set_clock_groups -group [get_clocks gth_refclk1_1 -include_generated_clocks] -asynchronous
create_clock -name gth_refclk0_3 -period 8.0 [get_ports gth_refclk0p_i[1]]
create_clock -name gth_refclk1_3 -period 8.0 [get_ports gth_refclk1p_i[1]]
set_clock_groups -group [get_clocks gth_refclk0_3 -include_generated_clocks] -asynchronous
set_clock_groups -group [get_clocks gth_refclk1_3 -include_generated_clocks] -asynchronous
create_clock -name gth_refclk0_4 -period 8.0 [get_ports gth_refclk0p_i[2]]
```

```
create_clock -name gth_refclk1_4 -period 8.0 [get_ports gth_refclk1p_i[2]]
set_clock_groups -group [get_clocks gth_refclk0_4 -include_generated_clocks] -asynchronous
set_clock_groups -group [get_clocks gth_refclk1_4 -include_generated_clocks] -asynchronous
create_clock -name gth_refclk0_5 -period 8.0 [get_ports gth_refclk0p_i[3]]
create_clock -name gth_refclk1_5 -period 8.0 [get_ports gth_refclk1p_i[3]]
set_clock_groups -group [get_clocks gth_refclk0_5 -include_generated_clocks] -asynchronous
set_clock_groups -group [get_clocks gth_refclk1_5 -include_generated_clocks] -asynchronous
##
## System clock pin locs and timing constraints
##
#set_property PACKAGE_PIN AH7 [get_ports gth_sysclkp_i]
#set_property IOSTANDARD DIFF_SSTL15 [get_ports gth_sysclkp_i]
```

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

SDK template in ./sw_lib/sw_apps/ available.

zynqmp_fsbl

TE modified 2019.2 FSBL

General:

- Modified Files: xfsbl_main.c, xfsbl_hooks.h/.c, xfsbl_board.h/.c(search for 'TE Mod' on source code)
- Add Files: te_xfsbl_hooks.h/.c (for hooks and board)\n\
- General Changes:
    - Display FSBL Banner and Device Name

Module Specific:

- Add Files: all TE Files start with te_*
    - Si5345 Configuration
    - OTG+PCIe Reset over MIO
    - I2C MUX for EEPROM MAC

zynqmp_fsbl_flash

TE modified 2019.2 FSBL

General:

- Modified Files: xfsbl_initialisation.c, xfsbl_hw.h, xfsbl_handoff.c, xfsbl_main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

hello_ibert

Hello TE0808 IBERTis a Xilinx Hello World example as endless loop instead of one console output.

Additional Software

SI5345

File location <design name>/misc/Si5345/Si5345-*.slabtimeproj

General documentation how you work with these project will be available on Si5345

Appx. A: Change History and Legal Notices

Document Change History

To get content of older revision got to "Change History" of this page and select older document revision number.

Date	Document Revision	Authors	Description

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	- new assembly variants
2020-03-25	v.1	John Hartfiel	- initial release

	All	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	

Document change history.

Legal Notices

Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]