TE0712 Test Board

Revision v.30

Exported on 2019-02-14

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0712+Test+Board

1 Table of Contents

2 Table of Figures

3 Table of Tables

Online version of this manual and other related documents can be found at https://wiki.trenz-electronic.de/display/PD/Trenz+Electronic+Documentation

4 Overview

4.1 Key Features

- PetaLinux
- MicroBlaze
- SREC
- I2C
- Flash
- MIG
- FMeter
- SI5338 initialisation with MCS
- ETH

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2018-09-05	2018.2	te0712-test_board-vivado_2018.2-build_03_20180906071356.zip te0712-test_board_noprebuilt-vivado_2018.2-build_03_20180906071434.zip	John Hartfiel	• chance block design: qspi clks, clock wizard(REV01 only) • change timing constrains • add hello_te0712 application • new SREC bootloader version • change linux device tree
2018-05-25	2017.4	te0712-test_board-vivado_2017.4-build_10_20180525155402.zip te0712-test_board_noprebuilt-vivado_2017.4-build_10_20180525155555.zip	John Hartfiel	• solved eth issue for REV01 • changed design + second design for REV01

Date	Vivado	Project Built	Authors	Description
2018-04-12	2017.4	te0712-test_board-vivado_2017.4-build_07_20180412081225.zip te0712-test_board_noprebuilt-vivado_2017.4-build_07_20180412081253.zip	John Hartfiel	• bugfix constrain file - ETH REFCLK, timing
2018-03-28	2017.4	te0712-test_board-vivado_2017.4-build_07_20180328145151.zip te0712-test_board_noprebuilt-vivado_2017.4-build_07_20180328145135.zip	John Hartfiel	• new assembly variant
2018-01-08	2017.4	te0712-test_board-vivado_2017.4-build_02_20180108155712.zip te0712-test_board_noprebuilt-vivado_2017.4-build_02_20180108155735.zip	John Hartfiel	• no design changes • small constrain changes
2017-12-15	2017.2	te0712-test_board-vivado_2017.2-build_07_20171215172447.zip te0712-test_board_noprebuilt-vivado_2017.2-build_07_20171215172514.zip	John Hartfiel	• add SI5338 initialisation with MCS • add Ethernet IP

Date	Vivado	Project Built	Authors	Description
2017-11-07	2017.2	te0712-test_board-vivado_2017.2-build_05_20171107172917.zip te0712-test_board_noprebuilt-vivado_2017.2-build_05_20171107172939.zip	John Hartfiel	• add wiki link in Boart Part Files • set correct short link for te0712-02-200-2c
2017-10-05	2017.2	te0712-test_board-vivado_2017.2-build_03_20171005082148.zip te0712-test_board_noprebuilt-vivado_2017.2-build_03_20171005082225.zip	John Hartfiel	• initial release

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
For PCB REV01 only: prebuilt does not boot	There is a Pullup missing on REV01 I2C SCL, so SI5338 configuration over MCS fails	Remove MCS	**solved** with 20180528 update
For PCB REV01 only: CLK1B is not available on	additional clk is not connected on PCB	use other internal generated CLK, maybe more effort is needed to get ETH running	**solved** with 20180528 update
SREC SPI BootLoader default Offset	Default load offset is set to 0x400000	Change manually on SDK to 0x5E0000	**solved** with 20180412 update

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vivado	2018.2	needed
SDK	2018.2	needed
PetaLinux	2018.2	needed

4.4.2 Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.[1]

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	Others	Notes
te0712-01-100-1i	01_100_1i	REV01	1GB	32MB		
te0712-01-100-2c	01_100_2c	REV01	1GB	32MB		
te0712-01-100-2c3	01_100_2c	REV01	1GB	32MB	2,5 mm connector	
te0712-01-200-1i	01_200_1i	REV01	1GB	32MB		
te0712-01-200-2i	01_200_2i	REV01	1GB	32MB		
te0712-01-200-2c	01_200_2c	REV01	1GB	32MB		
te0712-01-200-2c3	01_200_2c	REV01	1GB	32MB	2,5 mm connector	

1 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	Others	Notes
te0712-02-35-2i	35_2i	REV02	1GB	32MB		
te0712-02-100-1i	100_1i	REV02	1GB	32MB		
te0712-02-100-2c	100_2c	REV02	1GB	32MB		
te0712-02-100-2c3	100_2c	REV02	1GB	32MB	2,5 mm connector	
te0712-02-100-2ca	100_2ca	REV02	1GB	32MB		Micron QSPI Flash
te0712-02-200-1i	200_1i	REV02	1GB	32MB		
te0712-02-200-1i3	200_1i	REV02	1GB	32MB	2,5 mm connector	
te0712-02-200-2i	200_2i	REV02	1GB	32MB		
te0712-02-200-2c	200_2c	REV02	1GB	32MB		
te0712-02-200-2c3	200_2c	REV02	1GB	32MB	2,5 mm connector	

Design supports following carriers:

Carrier Model	Notes
TE0701	
TE0703	used as reference carrier
TE0705	
TE0706	
TEBA0841	

Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct typ
XMOD Programmer	Carrier Board dependent, only if carrier has no own FTDI

4.5 Content

For general structure and of the reference design, see Project Delivery[2]

4.5.1 Design Sources

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

4.5.2 Additional Sources

Type	Location	Notes
SI5338 Project	\misc\SI5338	

4.5.3 Prebuilt

File	File-Extension	Description
BIT-File	*.bit	FPGA (PL Part) Configuration File

2 https://wiki.trenz-electronic.de/display/PD/Project+Delivery

File	File-Extension	Description
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.hdf	Exported Vivado Hardware Specification for SDK/HSI and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
MCS-File	*.mcs	Flash Configuration File with Boot-Image (MicroBlaze or FPGA part only)
MMI-File	*.mmi	File with BRAM-Location to generate MCS or BIT-File with *.elf content (MicroBlaze only)
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems
SREC-File	*.srec	Converted Software Application for MicroBlaze Processor Systems

4.5.4 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:

- TE0712 "Test Board" Reference Design[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0712/Reference_Design/2018.2/test_board

5 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.

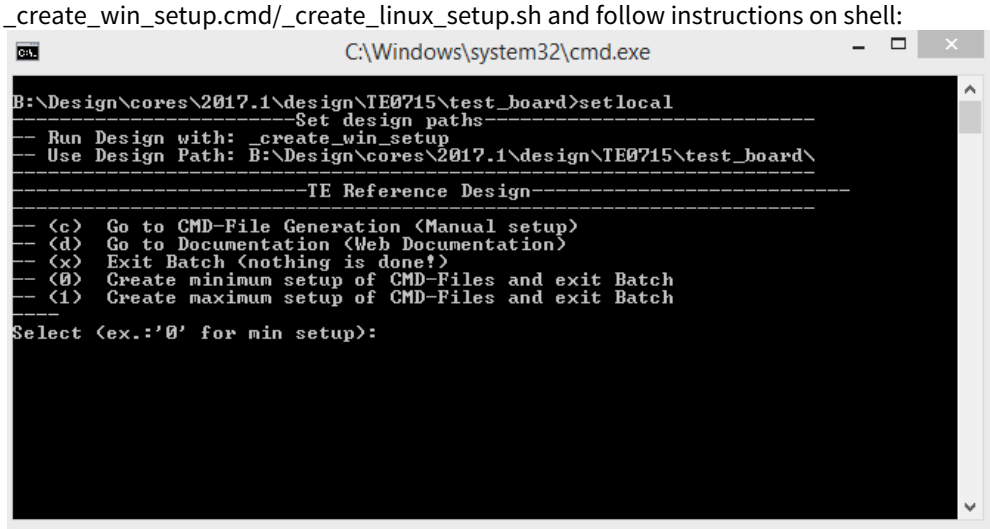
See also:Vivado/SDK/SDSoC[4]

- Vivado/SDK/SDSoC[5]
- Vivado Projects[6]
- Project Delivery.[7]

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[8]

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



2. Press 0 and enter for minimum setup
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x: \<design name>)
4. Create Project
   a. 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[9]
5. Create HDF 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

4 https://wiki.trenz-electronic.de/pages/viewpage.action?pageId=14746264#Vivado/SDK/SDSoC-XilinxSoftware-BasicUserGuides
5 https://wiki.trenz-electronic.de/pages/viewpage.action?pageId=14746264#Vivado/SDK/SDSoC-XilinxSoftware-BasicUserGuides
6 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects
7 https://wiki.trenz-electronic.de/display/PD/Project+Delivery
8 https://wiki.trenz-electronic.de/display/PD/Project+Delivery#ProjectDelivery-Currentlylimitationsoffunctionality
9 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

6. Create Linux (uboot.elf and image.ub) with exported HDF
    a. HDF is exported to "prebuilt\hardware\<short name>"
       Note: HW Export from Vivado GUI create another path as default workspace.
       Create Linux images on VM, see PetaLinux KICKstart[10]
        i. Use TE Template from /os/petalinux
           Note: run init_config.sh before you start petalinux config. This will set correct temporary path variable.
           Important Note: Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings → Flash Settings, FPGA+Boot+bootenv=0x900000 (increase automatically generate Boot partition), increas image size to A:, see Config (see page 23)
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
    a. "prebuilt\os\petalinux\default" or "prebuilt\os\petalinux\<short name>"
       Notes: Scripts select "prebuilt\os\petalinux\<short name>", if exist, otherwise "prebuilt\os\petalinux\default"
8. (not longer needed manually: This will be done with Step 10.a automatically with newer scripts (2017.4.10))
   Generate UBoot SREC:
    a. Create SDK Project with TE Scripts on Vivado TCL: TE::sw_run_sdk
    b. Create "uboot-dummy" application
       Note: Use Hello World Example
    c. Copy u-boot.elf into "\workspace\sdk\uboot-dummy\Debug"
    d. Open "uboot-dummy" properties → C/C++ Build → Settings and go into Build Steps Tap.
    e. Add to Post-build steps: mb-objcopy -O srec u-boot.elf u-boot.srec
    f. Press Apply or regenerate project
       Note: SREC is generated on "\workspace\sdk\uboot-dummy\Debug\u-boot.srec"
9. Generate MCS Firmware (optional):
    a. Create SDK Project with TE Scripts on Vivado TCL: TE::sw_run_sdk
    b. Create "SCU" application
       Note: Select MCS Microblaze and SCU Application
    c. Select Release Built
    d. Regenerate App
10. Generate Programming Files with HSI/SDK
    a. Run on Vivado TCL: TE::sw_run_hsi
       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_sdk
       Note: See SDK Projects[11]
11. Copy "\prebuilt\software\<short name>\srec_spi_bootloader.elf" into "\firmware\microblaze_0\"
12. (optional) Copy "\\workspace\sdk\scu\Release\scu.elf" into "\firmware\microblaze_mcs_0\"
13. Regenerate Vivado Project or Update Bitfile only with "srec_spi_bootloader.elf" and "scu.elf"

10 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
11 https://wiki.trenz-electronic.de/display/PD/SDK+Projects

6 Launch

6.1 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[12]

6.1.1 QSPI

1. Connect JTAG and power on PCB
2. (if not done) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd" or open with "vivado_open_project_guimode.cmd", if generated.
3. Type on Vivado Console: TE::pr_program_flash_mcsfile -swapp u-boot
   Note: Alternative use SDK or setup Flash on Vivado manually
   optional "TE::pr_program_flash_binfile -swapp hello_te0712" possible
4. Reboot (if not done automatically)

6.1.2 SD

Not used on this Example.

6.1.3 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming (see page 16)
2. Connect UART USB (most cases same as JTAG)
3. Power on PCB
   Note: FPGA Loads Bitfile from Flash,MCS Firmware configure SI5338 and starts Microblaze, SREC Bootloader from Bitfile Firmware loads U-Boot into DDR (This takes a while), U-boot loads Linux from QSPI Flash into DDR

Boot process takes a while, please wait.

12 https://wiki.trenz-electronic.de/pages/viewpage.action?pageId=14746264#Vivado/SDK/SDSoC-XilinxSoftwareProgrammingandDebugging

```
SREC SPI Bootloader (TE modified): Start initialization

SREC SPI Bootloader (TE modified): SPI driver Init passed

SREC SPI Bootloader (TE modified): Serial Flash Library Init passed

SREC SPI Bootloader (TE modified): Load Image
Loading SREC image from flash @ address: 005e0000
Please wait...
```

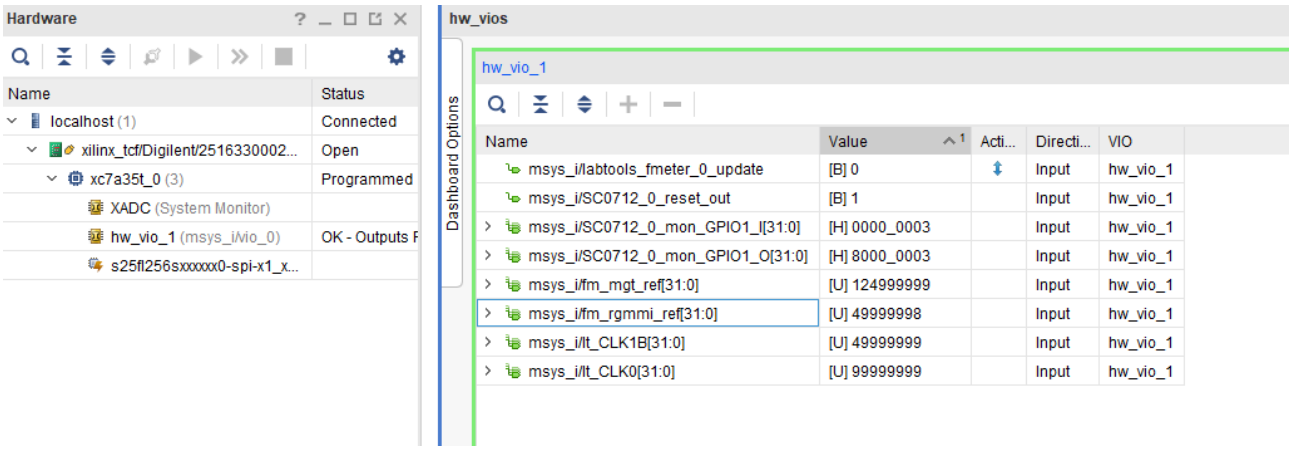
6.2.1 Linux

Note: Linux boot process is slower on Microblaze.

1. Open Serial Console (e.g. putty)
   a. Speed: 9600
   b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)
2. Linux Console:
   Note: Wait until Linux boot finished For Linux Login use:
   a. User Name: root
   b. Password: root
3. You can use Linux shell now.
   a. ETH0 works with udhcpc

6.2.2 Vivado HW Manager:

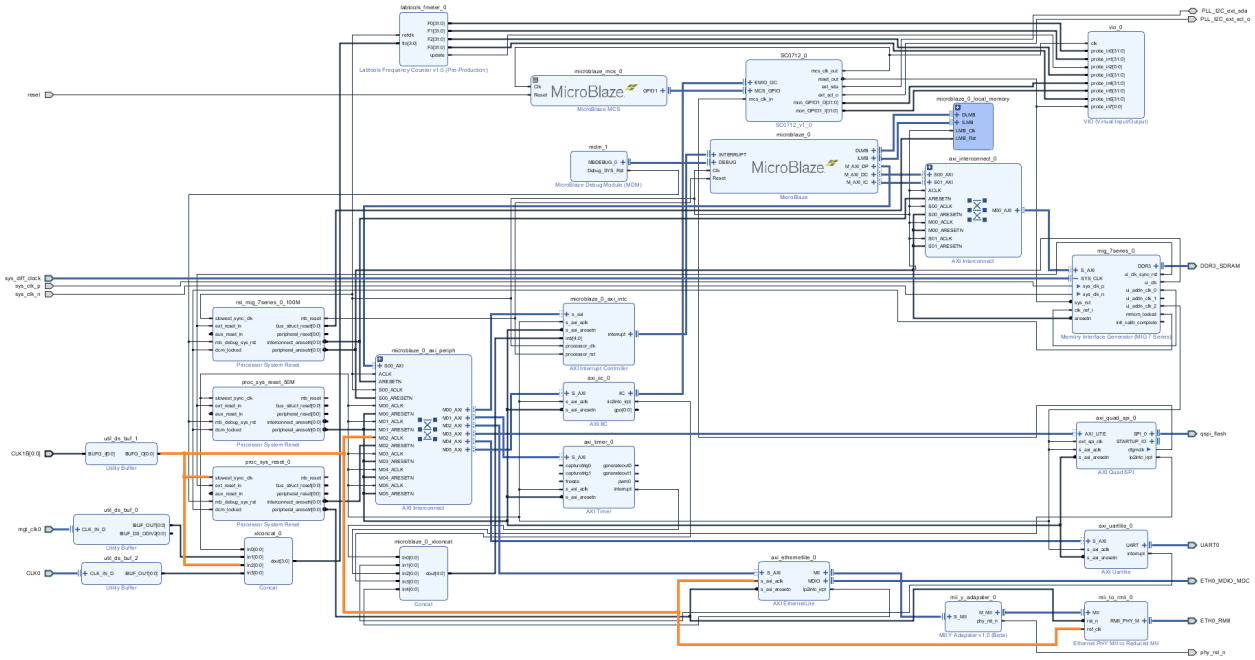
1. Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
   a. Set radix from VIO signals (MGT REF, MIG_OUT, CLK1B, CLK0) to unsigned integer.
      Note: Frequency Counter is inaccurate and displayed unit is Hz
   b. MGT REF~125MHz, MIG_50MHZ~50MHz., CLK1B ~50MHz, CLK0~100MHz
   c. Additional Infos: System reset from MCS and GIO outputs



7 System Design - Vivado

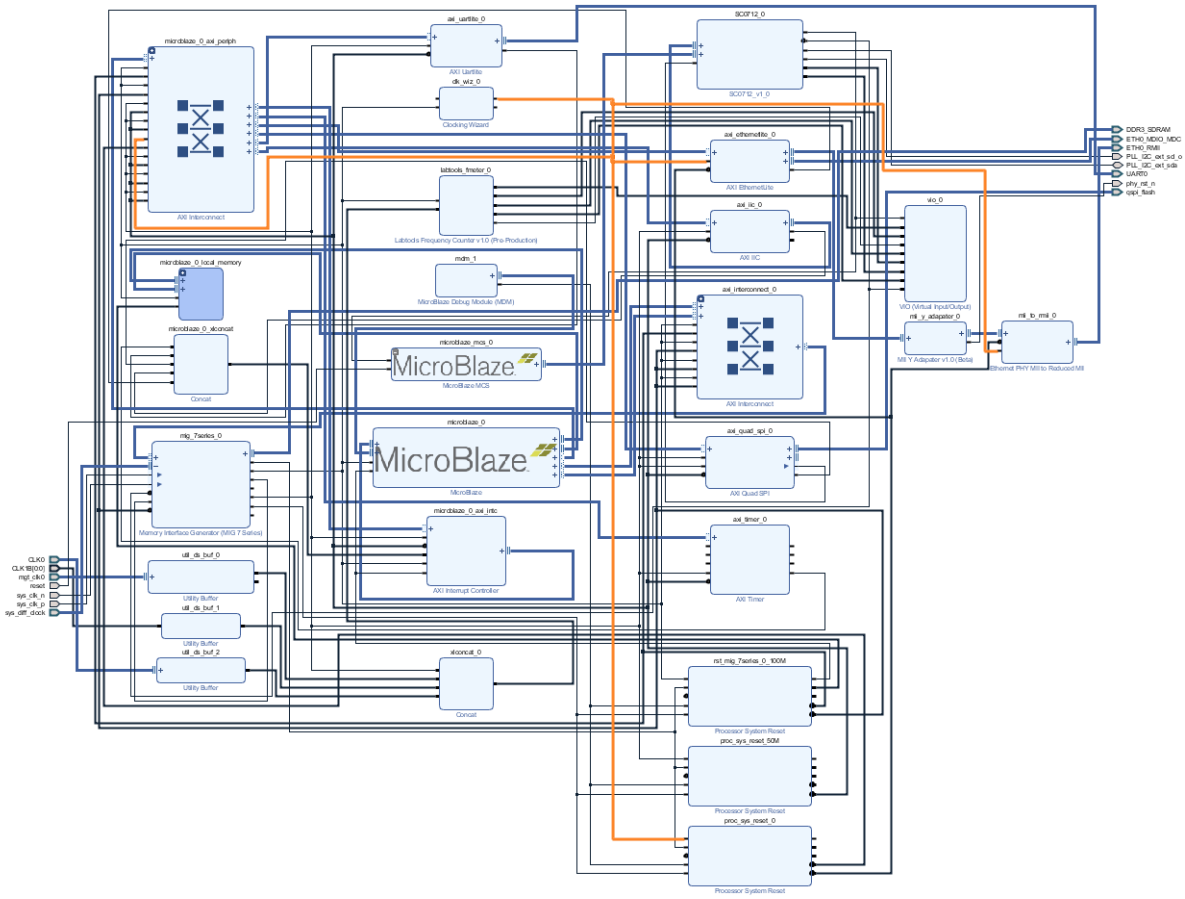
7.1 Block Design

7.1.1 REV02



7.1.2 REV01

Same as REV02 but 50 MHz ETH REV CLK is generated from MIG output with 180° Phase shift.



7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen_common.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 66 [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.CONFIG.M1PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M2PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M0PIN PULLNONE [current_design]

set_property BITSTREAM.CONFIG.USR_ACCESS TIMESTAMP [current_design]
```

_i_bitgen.xdc

```
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLDOWN [current_design]
```

7.2.2 Design specific constrain

_i_reset.xdc

```
set_property PULLDOWN true [get_ports reset]
```

_i_io.xdc

```
#I2C
#set_property PACKAGE_PIN W21 [get_ports PLL_I2C_scl_io]
#set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_scl_io]
#set_property PACKAGE_PIN T20 [get_ports PLL_I2C_sda_io]
#set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_sda_io]
set_property PACKAGE_PIN W21 [get_ports PLL_I2C_ext_scl_o]
set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_ext_scl_o]
set_property PACKAGE_PIN T20 [get_ports PLL_I2C_ext_sda]
set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_ext_sda]

#Reset
set_property PACKAGE_PIN T3 [get_ports reset]
set_property IOSTANDARD LVCMOS15 [get_ports reset]
#CLKS
set_property PACKAGE_PIN R4 [get_ports {CLK1B[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {CLK1B[0]}]
set_property PACKAGE_PIN K4 [get_ports {CLK0_clk_p[0]}]
set_property IOSTANDARD DIFF_SSTL15 [get_ports {CLK0_clk_p[0]}]

#ETH PHY
set_property PACKAGE_PIN N17 [get_ports phy_rst_n]
set_property IOSTANDARD LVCMOS33 [get_ports phy_rst_n
```

_i_timing.xdc

```
create_clock -period 8.000 -name mgt_clk0_clk_p -waveform {0.000 4.000} [get_ports
mgt_clk0_clk_p]


create_clock -period 10.000 -name {CLK0_clk_p[0]} -waveform {0.000 5.000} [get_ports
{CLK0_clk_p[0]}]
create_clock -period 20.000 -name {CLK1B[0]} -waveform {0.000 10.000} [get_ports
{CLK1B[0]}]
create_clock -period 15.152 -name CFGMCLK -waveform {0.000 7.576} [get_pins -
hierarchical -filter {NAME =~*NO_DUAL_QUAD_MODE.QSPI_NORMAL/
*STARTUP_7SERIES_GEN.STARTUP2_7SERIES_inst/CFGMCLK}]


set_false_path -from [get_clocks {CLK0_clk_p[0]}] -to [get_clocks clk_pll_i]
set_false_path -from [get_clocks mgt_clk0_clk_p] -to [get_clocks clk_pll_i]
set_false_path -from [get_pins {msys_i/SC0712_0/U0/rst_delay_i_reg[3]/C}] -to
[get_pins -hierarchical -filter {NAME =~*u_msys_mig_7series_0_0_mig/
u_ddr3_infrastructure/rstdiv0*/PRE}]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/mig_7series_0/
u_msys_mig_7series_0_0_mig/u_ddr3_infrastructure/gen_ui_extra_clocks.mmcm_i/
CLKFBOUT]] -to [get_clocks mgt_clk0_clk_p]
set_false_path -from [get_clocks clk_pll_i] -to [get_clocks {msys_i/util_ds_buf_0/U0/
IBUF_OUT[0]}]
set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/F_reg[*]/C}] -to
[get_pins {msys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[*]/D}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/COUNTER_REFCLK_inst/
bl.DSP48E_2/CLK] -to [get_pins {msys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[*]/
D}]
set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/
FMETER_gen[*].COUNTER_F_inst/bl.DSP48E_2/CLK}] -to [get_pins {msys_i/
labtools_fmeter_0/U0/F_reg[*]/D}]
```

8 Software Design - SDK/HSI

For SDK project creation, follow instructions from:

SDK Projects[13]

8.1 Application

Template location: ./sw_lib/sw_apps/

8.1.1 scu

MCS Firmware to configure SI5338 and Reset System.

8.1.2 srec_spi_bootloader

Boadloader to load app or second bootloader from flash into DDR

Changes:

- Add some vonsole outputs and changed bootloader read address.
- Add bugfix for 2018.2 qspi flash

8.1.3 xilisf_v5_11

- Changed default Flash Typ to 5.

8.1.4 hello_te0712

Hello TE0712 is a Xilinx Hello World example as endless loop instead of one console output.

8.1.5 u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate u-boot.srec. Vivado to generate *.mcs

13 https://wiki.trenz-electronic.de/display/PD/SDK+Projects

9 Software Design - PetaLinux

- PetaLinux KICKstart[14]

Description currently not available.

9.1 Config

- Set kernel flash Address to 0x900000 and Kernel size to 0xA00000:
  (--> Subsystem Auto Hardware Settings --> Flash Settings)
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = 0x5E0000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = 0x300000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = 0x20000
  - SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = 0xA00000

9.2 U-Boot

```
#include <configs/platform-auto.h>

#define CONFIG_SYS_BOOTM_LEN 0xF000000

/* ethernet - axi_ethernetlite_0 */
#undef CONFIG_PHY_XILINX
#undef XILINX_EMACLITE_BASEADDR    0x40E00000
#undef CONFIG_MII
#undef CONFIG_NET_MULTI
#undef CONFIG_NETCONSOLE    1
#undef CONFIG_SERVERIP    192.168.150.127
#undef CONFIG_IPADDR

/* PREBOOT */
#define CONFIG_PREBOOT    "echo U-BOOT for petalinux;setenv preboot; echo;"
```

14 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

9.3 Device Tree

```
/include/ "system-conf.dtsi"
/ {
};

/* QSPI PHY */

&axi_quad_spi_0 {
    #address-cells = <1>;
    #size-cells = <0>;
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        spi-tx-bus-width=<1>;
        spi-rx-bus-width=<4>;
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
        spi-max-frequency = <25000000>;
    };
};


/* ETH PHY */
&axi_ethernetlite_0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};
```

9.4 Kernel

No changes.

9.5 Rootfs

No changes.

9.6 Applications

No changes.

10 Additional Software

10.1 SI5338

Download ClockBuilder Desktop for SI5338[15]

1. Install and start ClockBuilder
2. Select SI5338
3. Options → Open register map file
   Note: File location <design name>/misc/Si5338/RegisterMap.txt
4. Modify settings
5. Options → save C code header files
6. Replace Header files from FSBL template with generated file

15 https://www.silabs.com/products/development-tools/software/clock

11 Appx. A: Change History and Legal Notices

11.1 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
2018-09-06	v.30 (see page 6)	John Hartfiel[16]	• 2018.2 release
25.05.2018	v.28	John Hartfiel[17]	• Design update
08.05.2018	v.27	John Hartfiel[18]	• Know Issues • Documentation
12.04.2018	v.23	John Hartfiel[19]	• Design Update
28.03.2018	v.22	John Hartfiel[20]	• Know Issue for PCB REV01 only • Fix typo • New assembly variant
2018-02-13	v.19	John Hartfiel[21]	• Release 2017.4
2018-01-08	v.16	John Hartfiel[22]	• Add SCU source path
2017-12-15	v.15	John Hartfiel[23]	• Update Design and Description
2017-11-07	v.11	John Hartfiel[24]	• Update Design Files
2017-10-06	v.10	John Hartfiel[25]	• small Document Update
2017-10-05	v.8	John Hartfiel[26]	• Release 2017.2

16 https://wiki.trenz-electronic.de/display/~j.hartfiel
17 https://wiki.trenz-electronic.de/display/~j.hartfiel
18 https://wiki.trenz-electronic.de/display/~j.hartfiel
19 https://wiki.trenz-electronic.de/display/~j.hartfiel
20 https://wiki.trenz-electronic.de/display/~j.hartfiel
21 https://wiki.trenz-electronic.de/display/~j.hartfiel
22 https://wiki.trenz-electronic.de/display/~j.hartfiel
23 https://wiki.trenz-electronic.de/display/~j.hartfiel
24 https://wiki.trenz-electronic.de/display/~j.hartfiel
25 https://wiki.trenz-electronic.de/display/~j.hartfiel
26 https://wiki.trenz-electronic.de/display/~j.hartfiel

Date	Document Revision	Authors	Description
2017-09-11	v.1	John Hartfiel[27]	• Initial release
	All	John Hartfiel[28]	

11.2 Legal Notices

11.3 Data privacy

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

11.4 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.

11.5 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.

11.6 Copyright Notice

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Trenz Electronic.

11.7 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.

27 https://wiki.trenz-electronic.de/display/~j.hartfiel
28 https://wiki.trenz-electronic.de/display/~j.hartfiel

11.8 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.

11.9 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[29]. 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[30] 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)[31].

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.

2018-09-18

29 http://guidance.echa.europa.eu/
30 https://echa.europa.eu/candidate-list-table
31 http://www.echa.europa.eu/