TE0782 Test Board

Revision v.9

Exported on 2023-10-10

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0782+Test+Board

1 Table of Contents

Copyright © Trenz Electronic GmbH

Copyright © Trenz Electronic GmbH

2 Table of Figures

Copyright © Trenz Electronic GmbH

3 Table of Tables

Copyright © Trenz Electronic GmbH

4 Overview

Zynq PS Design with Linux example and monitoring over VIO with Vivado HW-Manager.

Refer to http://trenz.org/te0782-info for the current online version of this manual and other available documentation.

4.1 Key Features

- PetaLinux
- 2x ETH
- 2x USB
- I2C
- RTC
- Modified FSBL for SI5338 programming
- Special FSBL for QSPI programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2018-10-10	2018.2	TE0782-test_board-vivado_2018.2-build_03_20181009164622.zip TE0782-test_board_noprebuilt-vivado_2018.2-build_03_20181009164650.zip	John Hartfiel	initial release

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
No known issues	---	---	---

Table 2: Known Issues

Copyright © Trenz Electronic GmbH

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vivado	2018.2	needed
SDK	2018.2	needed
PetaLinux	2018.2	needed
SI5338 Clock Builder	---	optional

Table 3: Software

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
TE0782-02-035-2I	35	REV02	1GB	32MB	Hyperflash not soldered	
TE0782-02-045-2I	45	REV02	1GB	32MB	Hyperflash not soldered	

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

Copyright © Trenz Electronic GmbH

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	Others	Notes
TE0782-02-100_2I	100	REV02	1GB	32MB	Hyperflash not soldered	

Table 4: Hardware Modules
Design supports following carriers:

Carrier Model	Notes
TEBT0782	

Table 5: Hardware Carrier
Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	for XMOD
XMOD Programmer	for JTAG and UART

Table 6: Additional Hardware

4.5 Content

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

4.5.1 Design Sources

Type	Location	Notes
Vivado	<design name>/block_design <design name>/constraints <design name>/ip_lib	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

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

Copyright © Trenz Electronic GmbH

Type	Location	Notes
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration

Table 7: Design sources

4.5.2 Additional Sources

Type	Location	Notes
SI5338	<design name>/misc/Si5338	SI5345 Project with current PLL Configuration

Table 8: Additional design sources

4.5.3 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	*.hdf	Exported Vivado Hardware Specification for SDK/HSI and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File

Copyright © Trenz Electronic GmbH

File	File-Extension	Description
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

Table 9: Prebuilt files (only on ZIP with prebult content)

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:

- TE0782 "Test Board" Reference Design[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/8.5x8.5/TE0782/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.

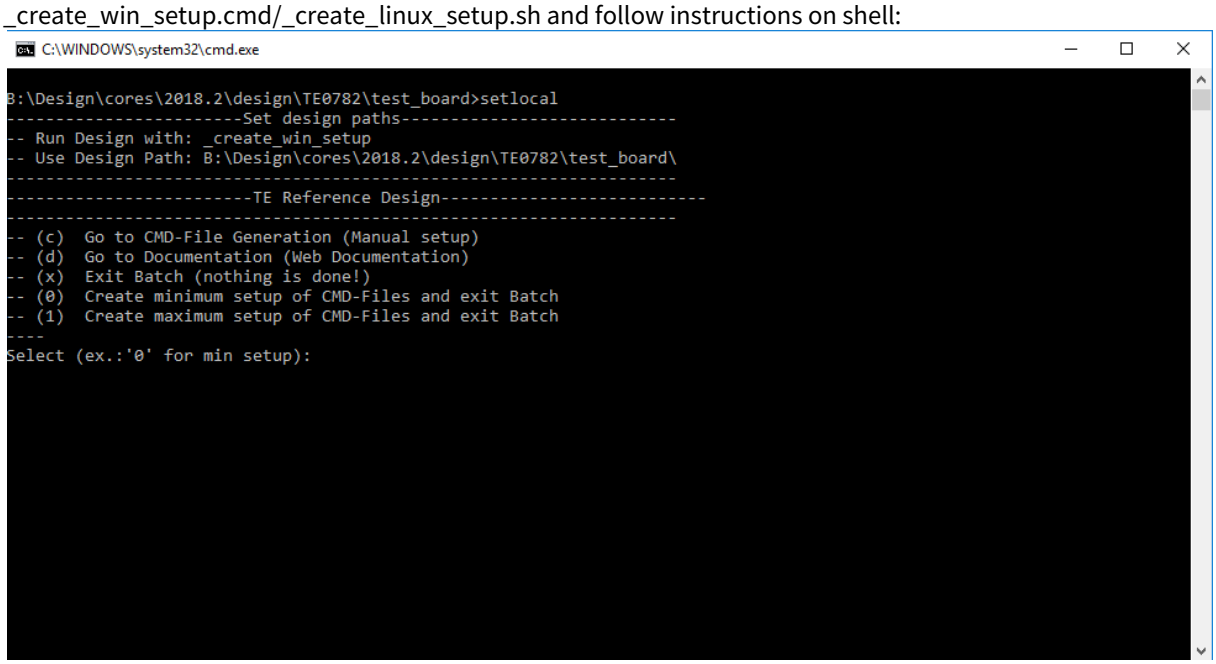
See also:

- AMD Development Tools[4]
- Vivado Projects - TE Reference Design[5]
- Project Delivery.[6]

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[7]

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[8]
5. Create HDF and export to prebuilt folder

4 https://wiki.trenz-electronic.de/display/PD/AMD+Development+Tools#AMDDevelopmentTools-XilinxSoftware-BasicUserGuides
5 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
6 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+AMD+devices
7 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+AMD+devices#ProjectDeliveryAMDdevices-Currentlylimitationsoffunctionality
8 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

Copyright © Trenz Electronic GmbH

   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. 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[9]
      i. Use TE Template from /os/petalinux
         Note: Important Note: Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings → Flash Settings, FPGA+bootenv=0x520000
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. 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[10]

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

Copyright © Trenz Electronic GmbH

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[11]

6.1.1 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 u-boot
   Note: To program with SDK/Vivado GUI, use special FSBL (zynqmp_fsbl_flash) on setup
   optional "TE::pr_program_flash_binfile -swapp hello_te0882" possible
4. Note: Linux image will be included into Boot.bin with HSI scripts

6.1.2 SD

Not supported with this module.

6.1.3 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming(see page 13)
2. Connect UART USB (most cases same as JTAG)
3. Power On PCB
   Note: 1. Zynq Boot ROM loads FSBL from QSPI Flash into OCM, 2. FSBL loads U-boot from QSPI Flash into DDR, 3. U-boot load Linux from QSPI Flash into DDR

6.2.1 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. Linux Console:
   Note: Wait until Linux boot finished For Linux Login use:

11 https://wiki.trenz-electronic.de/display/PD/AMD+Development+Tools#AMDDevelopmentTools-XilinxSoftwareProgrammingandDebugging

Copyright © Trenz Electronic GmbH

   a. User Name: root
   b. Password: root
3. You can use Linux shell now.
   a. I2C1 Bus type: i2cdetect -y -r 0
   b. RTC check: dmesg | grep rtc
   c. ETH0: works with udhcpc
   d. ETH1: ifconfig eth1 up
      ifconfig eth1 <ip>
      Note: "macb e000c000.ethernet eth1: unable to generate target frequency: 125000000 Hz" can be ignored
   e. USB0: insert USB device
   f. USB1: insert USB device

6.2.2 Vivado HW Manager

6.2.3 Monitoring:

- ETH1 and 2 PHY LED outputs
- ETH2 GMII_TO_RGMII IP Status

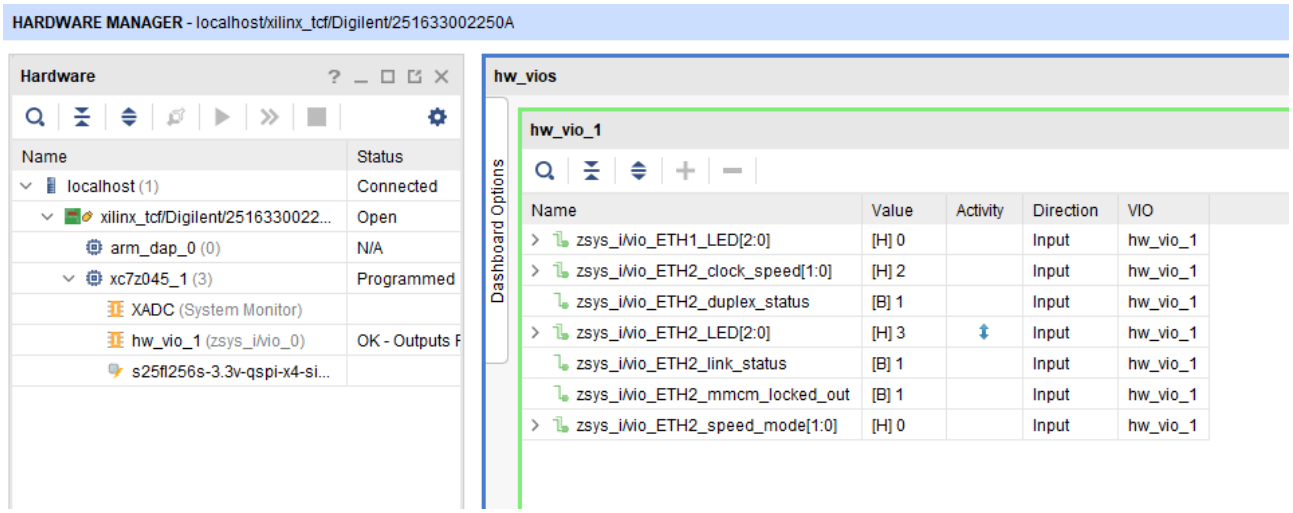


Figure 1: Vivado Hardware Manager

7 System Design - Vivado

7.1 Block Design

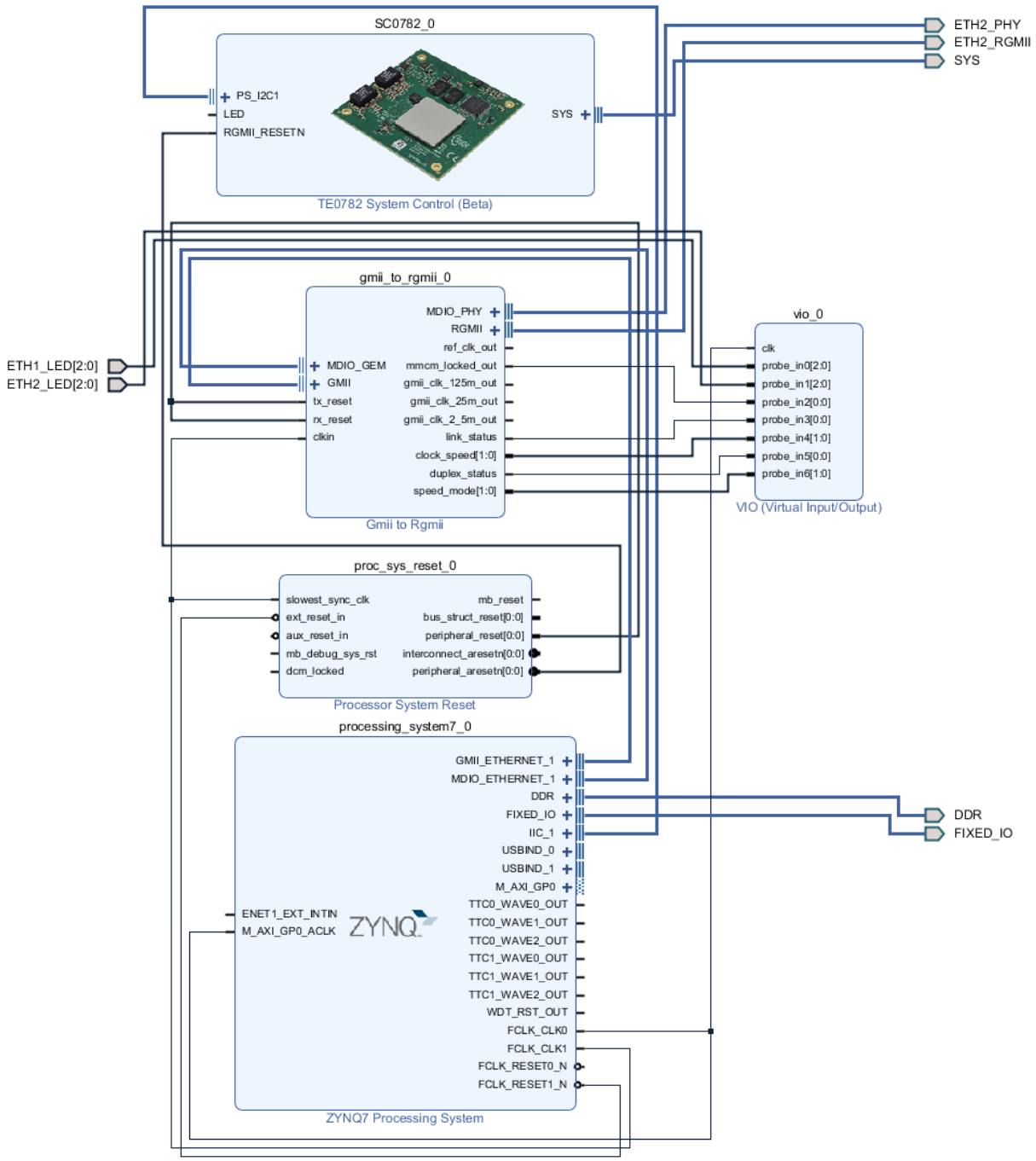


Figure 2: Block Design

7.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
ETH0	MIO
ETH1	EMIO
USB0	MIO
USB1	MIO
SD1	MIO
UART1	MIO
I2C1	EMIO
GPIO0	MIO, plus ETH0 and USB0 reset
WDT	
TTC0..1	

Table 10: PS Interfaces

7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen_common.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
```

Copyright © Trenz Electronic GmbH

7.2.2 Design specific constrain

_i_TE0782_io.xdc

```
##############################################################################
# Eternet2
set_property PACKAGE_PIN C17  [get_ports ETH2_PHY_mdc]
set_property PACKAGE_PIN B17  [get_ports ETH2_PHY_mdio_io]
set_property PACKAGE_PIN AD20 [get_ports {ETH2_RGMII_rd[0]}]
set_property PACKAGE_PIN AD19 [get_ports {ETH2_RGMII_rd[1]}]
set_property PACKAGE_PIN AB20 [get_ports {ETH2_RGMII_rd[2]}]
set_property PACKAGE_PIN AB19 [get_ports {ETH2_RGMII_rd[3]}]
set_property PACKAGE_PIN AE20 [get_ports ETH2_RGMII_rx_ctl]
set_property PACKAGE_PIN AD18 [get_ports ETH2_RGMII_rxc]
set_property PACKAGE_PIN AA20 [get_ports {ETH2_RGMII_td[0]}]
set_property PACKAGE_PIN Y20  [get_ports {ETH2_RGMII_td[1]}]
set_property PACKAGE_PIN AA19 [get_ports {ETH2_RGMII_td[2]}]
set_property PACKAGE_PIN AA18 [get_ports {ETH2_RGMII_td[3]}]
set_property PACKAGE_PIN AC18 [get_ports ETH2_RGMII_tx_ctl]
set_property PACKAGE_PIN AC19 [get_ports ETH2_RGMII_txc]
set_property IOSTANDARD LVCMOS18 [get_ports ETH2*]
set_property IOSTANDARD LVCMOS18 [get_ports ETH2_PHY_mdio_io]
##############################################################################
set_property PACKAGE_PIN B12 [get_ports {ETH1_LED[0]}]
set_property PACKAGE_PIN C12 [get_ports {ETH1_LED[1]}]
set_property PACKAGE_PIN A15 [get_ports {ETH1_LED[2]}]
set_property PACKAGE_PIN K15 [get_ports {ETH2_LED[0]}]
set_property PACKAGE_PIN B16 [get_ports {ETH2_LED[1]}]
set_property PACKAGE_PIN A17 [get_ports {ETH2_LED[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports ETH1*]
#set_property IOSTANDARD LVCMOS18 [get_ports ETH2*]
##############################################################################
#-----------------------------------------------------------------------------
#set_property IDELAY_VALUE "20" [get_cells -hier -filter {name =~ *gmii_to_rgmii/
*delay_rgmii_rx_ctl }]
#set_property IDELAY_VALUE "20" [get_cells -hier -filter {name =~ *gmii_to_rgmii/
*delay_rgmii_rxd* }]
#-----------------------------------------------------------------------------
#set_property IODELAY_GROUP "grp1" [get_cells -hier -filter {name =~ *gmii_to_rgmii/
*delay_rgmii_rx_ctl }]
#set_property IODELAY_GROUP "grp1" [get_cells -hier -filter {name =~ *gmii_to_rgmii/
*delay_rgmii_rxd* }]
create_clock -add -name rgmii_rxc -period 8.000 [get_ports ETH2_RGMII_rxc]
##############################################################################
# VIO false path
set_false_path -from [get_pins zsys_i/gmii_to_rgmii_0/U0/i_gmii_to_rgmii_block/
zsys_gmii_to_rgmii_0_0_core/i_gmii_to_rgmii/i_gmii_to_rgmii/link_status_reg/C] -to
[get_pins {zsys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[7]/D}]
set_false_path -from [get_pins {zsys_i/gmii_to_rgmii_0/U0/i_gmii_to_rgmii_block/
zsys_gmii_to_rgmii_0_0_core/i_gmii_to_rgmii/i_gmii_to_rgmii/clock_speed_reg[0]/C}]
-to [get_pins {zsys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[8]/D}]
```

```
set_false_path -from [get_pins zsys_i/gmii_to_rgmii_0/U0/i_gmii_to_rgmii_block/
zsys_gmii_to_rgmii_0_0_core/i_gmii_to_rgmii/i_gmii_to_rgmii/duplex_status_reg/C] -to
[get_pins {zsys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[10]/D}]
set_false_path -from [get_pins {zsys_i/gmii_to_rgmii_0/U0/i_gmii_to_rgmii_block/
zsys_gmii_to_rgmii_0_0_core/i_gmii_to_rgmii/i_gmii_to_rgmii/clock_speed_reg[1]/C}]
-to [get_pins {zsys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[9]/D}]
##############################################################################
```

Copyright © Trenz Electronic GmbH

8 Software Design - SDK/HSI

For SDK project creation, follow instructions from:

SDK Projects[12]

8.1 Application

Template location: ./sw_lib/sw_apps/

8.1.1 zynq_fsbl

TE modified 2018.2 FSBL

Changes:

- Si5338 Configuration
  - see main.c, fsbl_hooks.c (Add/remove define RECONFIGURE_SI5338 to enable PLL programming with given register_map.h setup (default activate))
  - Add register_map.h, si5338.c, si5338.h

8.1.2 zynq_fsbl_flash

TE modified 2018.2 FSBL

Changes:

- Set FSBL Boot Mode to JTAG
- Disable Memory initialisation

8.1.3 hello_te0782

Hello TE0782 is a Xilinx Hello World example as endless loop instead of one console output.

8.1.4 u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate Boot.bin.

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

Copyright © Trenz Electronic GmbH

9 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[13]

9.1 Config

changes:

- CONFIG_SUBSYSTEM_FLASH_PS7_QSPI_0_BANKLESS_PART0_SIZE=0x500000
- CONFIG_SUBSYSTEM_FLASH_PS7_QSPI_0_BANKLESS_PART2_SIZE=0xA80000
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_BOOT_MEDIA_FLASH_SELECT=y
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_MEDIA_FLASH_SELECT=y

9.2 U-Boot

```
#include <configs/platform-auto.h>
#define CONFIG_SYS_BOOTM_LEN 0xF000000
#define DFU_ALT_INFO_RAM \
                "dfu_ram_info=" \
        "setenv dfu_alt_info " \
        "image.ub ram $netstart 0x1e00000\0" \
        "dfu_ram=run dfu_ram_info && dfu 0 ram 0\0" \
        "thor_ram=run dfu_ram_info && thordown 0 ram 0\0"

#define DFU_ALT_INFO_MMC \
        "dfu_mmc_info=" \
        "set dfu_alt_info " \
        "${kernel_image} fat 0 1\\\\;" \
        "dfu_mmc=run dfu_mmc_info && dfu 0 mmc 0\0" \
        "thor_mmc=run dfu_mmc_info && thordown 0 mmc 0\0"


/*Required for uartless designs */
#ifndef CONFIG_BAUDRATE
#define CONFIG_BAUDRATE 115200
#ifdef CONFIG_DEBUG_UART
#undef CONFIG_DEBUG_UART
#endif
#endif

/*Define CONFIG_ZYNQ_EEPROM here and its necessaries in u-boot menuconfig if you had
EEPROM memory. */
#ifdef CONFIG_ZYNQ_EEPROM
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN         1
#define CONFIG_SYS_I2C_EEPROM_ADDR             0x54
#define CONFIG_SYS_EEPROM_PAGE_WRITE_BITS      4
#define CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS  5
```

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

```
#define CONFIG_SYS_EEPROM_SIZE                  1024 /* Bytes */
#define CONFIG_SYS_I2C_MUX_ADDR                 0x74
#define CONFIG_SYS_I2C_MUX_EEPROM_SEL           0x4
#endif
```

9.3 Device Tree

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


/* QSPI PHY */
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
    };
};





/* ETH PHY ETH0 */
&gem0{
    status = "okay";
    phy-handle = <&phy0>;
    xlnx,has-mdio = <0x1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@1 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <1>;
            marvell,reg-init = <0x3 0x10 0x0000 0x0501 0x3 0x11 0x0000 0x4415>;
        };
    };
};

/* ETH PHY ETH1 RGMII over PL */

&gem1 {
    gmii2rgmii-phy-handle = <&gmii_to_rgmii_0>;
    reg = <0xe000c000 0x1000>;
    phy-handle = <&phy2>;
```

```
    local-mac-address = [00 0a 35 00 00 01];
    compatible = "cdns,zynq-gem", "cdns,gem";
    clock-names = "pclk", "hclk", "tx_clk";
    clocks = <&clkc 31>, <&clkc 31>, <&clkc 14>;
    phy-mode = "gmii";
    status = "okay";
    ps7_ethernet_1_mdio: mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        gmii_to_rgmii_0: phy@8 {
            compatible = "xlnx,gmii-to-rgmii-1.0";
            phy-handle = <&phy2>;
            reg = <8>;
        };
        phy2: phy@2 {
                    compatible = "marvell,88e1510";
                    device_type = "ethernet-phy";
                    reg = <1>;
            marvell,reg-init = <0x3 0x10 0x0000 0x0501 0x3 0x11 0x0000 0x4415>;
                } ;
    };
};



/* USB 0 PHY */
/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb0 {
  dr_mode = "host";
    usb-phy = <&usb_phy0>;
} ;


/* USB 1 PHY */
/{
    usb_phy1: usb_phy@0 {
        compatible = "ulpi-phy";
        #phy-cells = <0>;
        reg = <0xe0003000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb1 {
  dr_mode = "host";
    usb-phy = <&usb_phy1>;
```

Copyright © Trenz Electronic GmbH

```
} ;


/* RTC over I2C1 */
&i2c1 {
    rtc@6F {        // Real Time Clock
       compatible = "isl12022";
       reg = <0x6F>;
   };
};
```

9.4 Kernel

Activate:

- RTC_DRV_ISL12022
- XILINX_GMII2RGMII

9.5 Rootfs

Activate:

- i2c-tools

9.6 Applications

Not included.

Copyright © Trenz Electronic GmbH

10 Additional Software

No additional software is needed.

10.1 SI5338

File location <design name>/misc/Si5338/RegisterMap.txt

General documentation how you work with these project will be available on Si5338[14]

14 https://wiki.trenz-electronic.de/display/PD/Si5338

Copyright © Trenz Electronic GmbH

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
2018-10-10	v.9(see page 6)	John Hartfiel[15]	• 2018.2 initial release
--	all	John Hartfiel[16]	--

Table 11: Document change history.

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

15 https://wiki.trenz-electronic.de/display/~j.hartfiel
16 https://wiki.trenz-electronic.de/display/~j.hartfiel

Copyright © Trenz Electronic GmbH

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.

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[17]. 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[18] 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)[19].

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.

17 http://guidance.echa.europa.eu/
18 https://echa.europa.eu/candidate-list-table
19 http://www.echa.europa.eu/

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.

2019-06-07

Copyright © Trenz Electronic GmbH