TE0782 Test Board

Table of contents

Overview

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

Key Features

- Vitis/Vivado 2022.2
- PetaLinux
- 2x ETH
- 2x USB
- I2C
- RTC
- Modified FSBL for SI5338 programming

Revision History

Date	Vivado	Project Built	Authors	Description
2023-12-05	2022.2	TE0782-test_board-vivado_2022.2-build_9_20231205104243.zip TE0782-test_board_noprebuilt-vivado_2022.2-build_9_20231205104243.zip	Waldemar Hanemann	update 2022.2 ▪ new features
2023-10-10	2022.2	TE0782-test_board-vivado_2022.2-build_9_20231010162944.zip TE0782-test_board_noprebuilt-vivado_2022.2-build_9_20231010162944.zip	Waldemar Hanemann	update 2022.2
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

Design Revision History

Release Notes and Know Issues

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

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2022.2	needed, Vivado is included into Vitis installation
PetaLinux	2022.2	needed
ClockBuilder Pro	---	optional

Software

Hardware

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

Complete List is available on "<project folder>\board_files*_board_files.csv"

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0782-02-035-2I	35	REV02	1GB	32MB	Hyperflash not soldered		
TE0782-02-045-2I*	45	REV02	1GB	32MB	Hyperflash not soldered		
TE0782-02-100_2I	100	REV02	1GB	32MB	Hyperflash not soldered		
TE0782-02-92I33MA	45	REV02	1GB	32MB	Hyperflash not soldered		
TE0782-02-A2I33MA	100	REV02	1GB	32MB	Hyperflash not soldered		

*used as reference

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TEBT0782-01*	

*used as reference

Hardware Carrier

Additional HW Requirements:

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

*used as reference

Additional Hardware

Content

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

Design Sources

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

Design sources

Additional Sources

Type	Location	Notes
SI5338	<design name>/misc/Si5338	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	*.hdf	Exported Vivado Hardware Specification for SDK/HSI and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File

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

Prebuilt files (only on ZIP with prebuilt content)

Download

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

Reference Design is available on:

- TE0782 "Test Board" Reference Design

Design Flow

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

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/Vitis GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

> ⚠ **Caution!** Win OS has a 260 character limit for path lengths which can affect the Vivado tools. To avoid this issue, use Virtual Drive or the shortest possible names and directory locations for the reference design (for example "x:\<project folder>")

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

_create_win_setup.cmd/_create_linux_setup.sh

```
-----------------------Set design paths---------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
------------------------------------------------------------------
-----------------------TE Reference
Design-------------------------
------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. Create project and follow instructions of the product selection guide, settings file will be configured automatically during this process.
    - 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 also Vivado Board Part Flow

4. Create hardware description file (.xsa file) for PetaLinux project and export to prebuilt folder

run on Vivado TCL (Script generates design and export files into "<project folder>\prebuilt\hardware\<short name>")

```
TE::hw_build_design -export_prebuilt
```

Using Vivado GUI is the same, except file export to prebuilt folder.

5. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart
    - use TE Template from "<project folder>\os\petalinux"
    - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.
    - The build images are located in the "<plnx-proj-root>/images/linux" directory
6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr
7. Generate Programming Files with Vitis (recommended)
    a. Copy PetaLinux build image files to prebuilt folder
        - copy **u-boot.elf**, **system.dtb**, **image.ub** and **boot.scr** from "<plnx-proj-root>/images/linux" to prebuilt folder

> "<project folder>\prebuilt\os\petalinux\<ddr size>" or "<project folder>\prebuilt\os\petalinux\<short name>"

b. Generate Programming Files with Vitis

run on Vivado TCL (Script generates applications and bootable files, which are defined in "test_board\sw_lib\apps_list.csv")

```
TE::sw_run_vitis -all
TE::sw_run_vitis (optional; Start Vitis from Vivado GUI or start with TE Scripts on Vivado TCL)
```

> TCL scripts generate also platform project, this must be done manually in case GUI is used. See Vitis

8. Generate Programming Files with Petalinux (alternative), see PetaLinux KICKstart

Launch

Programming

> Check Module and Carrier TRMs for proper HW configuration before you try any design.
>
> Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

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

Get prebuilt boot binaries

1. Run _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 folder>_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

QSPI-Boot mode

Option for **Boot.bin** on QSPI Flash.

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"

run on Vivado TCL (Script programs BOOT.bin on QSPI flash)

```
TE::pr_program_flash -swapp u-boot
TE::pr_program_flash -swapp hello_te0820 (optional)
```

Note: Linux image will be included into Boot.bin with

SD-Boot mode

Not used on this example.

JTAG

1. Connect JTAG cable and set bootmode to JTAG. Power on PCB.
2. Open Vivado HW Manager
3. Program FPGA with Bitfile from "prebuilt\hardware\<short dir>\"

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Select QSPI as Boot Mode

> Note: See TRM of the Carrier, which is used.

> Starting with Petalinux version 2020.1, the industry standard "Distro-Boot" boot flow for U-Boot was introduced, which significantly expands the possibilities of the boot process and has the primary goal of making booting much more standardised and predictable.
> The boot options described above describe the common boot processes for this hardware; other boot options are possible.
> For more information see Distro Boot with Boot.scr

4. Power On PCB
    1. Zynq Boot ROM loads FSBL from SD/QSPI into OCM,
    2. FSBL init PS, programs PL using the bitstream and loads U-boot from QSPI into DDR,
    3. U-boot loads Linux (**image.ub**) from QSPI/... into DDR

Linux

1. Open Serial Console (e.g. putty)
    - Speed: 115200
    - select COM Port

    > Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console shows up after boot up

> Note: Wait until Linux boot finished

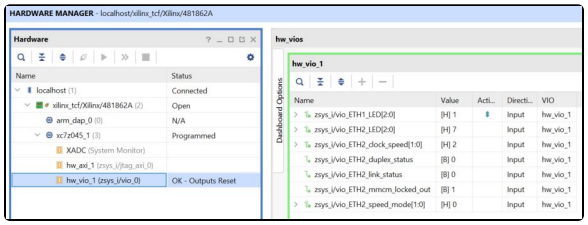
3. You can use Linux shell now.

```
i2cdetect -y -r 0        (check I2C 0 Bus)
dmesg | grep rtc        (RTC check)
udhcpc                                (ETH0 check)
lsusb                                (USB check)
```

Vivado HW Manager

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

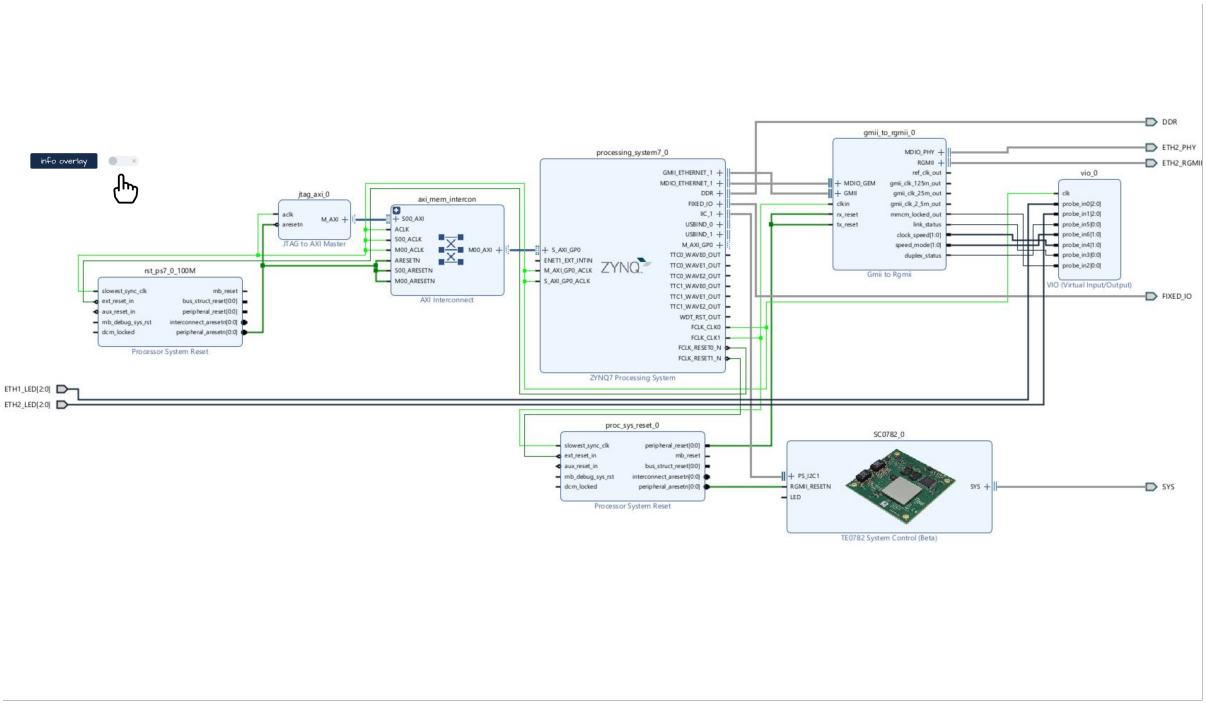
- Control: --
- Monitoring:
  - ETH1 and 2 PHY LED outputs
  - ETH2 GMII_TO_RGMII IP Status



Vivado Hardware Manager

System Design - Vivado

Block Design



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	

PS Interfaces

Constraints

Basic module constraints

_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]
```

Design specific constraints

_i_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 IOSTANDARD LVCMOS18 [get_ports SYS_eth1_clk125]
set_property IOSTANDARD LVCMOS18 [get_ports SYS_eth1_config]
set_property IOSTANDARD LVCMOS18 [get_ports SYS_eth2_clk125]
set_property IOSTANDARD LVCMOS18 [get_ports SYS_eth2_config]
set_property PACKAGE_PIN E16 [get_ports SYS_eth1_clk125]
set_property PACKAGE_PIN F14 [get_ports SYS_eth1_config]
set_property PACKAGE_PIN F15 [get_ports SYS_eth2_clk125]
set_property PACKAGE_PIN E15 [get_ports SYS_eth2_config]
#-----------------------------------------------------------------------------
-----
#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}]
#########################################################################
######
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
/gmii_to_rgmii_core_non_versal.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
/gmii_to_rgmii_core_non_versal.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
/gmii_to_rgmii_core_non_versal.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
/gmii_to_rgmii_core_non_versal.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}]
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

Template location: "<project folder>\sw_lib\sw_apps\"

zynq_fsbl

TE modified 2020.2 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

zynq_fsbl_flash

TE modified 2020.2 FSBL

Changes:

- Set FSBL Boot Mode to JTAG
- Disable Memory initialisation

hello_te0782

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

u-boot

U-Boot.elf is generated with PetaLinux. Vitis is used to generate Boot.bin.

Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

Start with **petalinux-config** or **petalinux-config --get-hw-description**

Changes:

- CONFIG_SUBSYSTEM_FLASH_PS7_QSPI_0_BANKLESS_PART0_SIZE=0x500000
- CONFIG_SUBSYSTEM_FLASH_PS7_QSPI_0_BANKLESS_PART2_SIZE=0x1040000

U-Boot

Start with **petalinux-config -c u-boot**

Changes:

- No changes.

Change bsp.cfg in path <.. petalinux\project-spec\meta-user\recipes-bsp\u-boot\files>:

```
CONFIG_SYS_CONFIG_NAME="platform-top"
CONFIG_BOOT_SCRIPT_OFFSET=0x520000
```

Device Tree

Change system-user.dtsi in path < .. petalinux\project-spec\meta-user\recipes-bsp\device-tree\files>:

```
/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>;

    nvmem-cells = <&eth0_addr>;
    nvmem-cell-names = "mac-address";

    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 {

        reg = <0xe000c000 0x1000>;
        phy-handle = <&phy1>;
  gmii2rgmii-phy-handle = <&gmii_to_rgmii_0>;

  nvmem-cells = <&eth1_addr>;
  nvmem-cell-names = "mac-address";

        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: gmii_to_rgmii_0@8 {
                        compatible = "xlnx,gmii-to-rgmii-1.0";
                        phy-handle = <&phy1>;
                        reg = <8>;
                };
                phy1: ethernet-phy@0 {
                    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>;
} ;


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

  //MAC EEPROM U24
   eeprom: eeprom@51 {
     compatible = "microchip,24aa025", "atmel,24c02";
     reg = <0x51>;

     #address-cells = <1>;
     #size-cells = <1>;
     eth0_addr: eth-mac-addr@FA {
       reg = <0xFA 0x06>;
     };
    };


     //MAC EEPROM U22
   eeprom50: eeprom@50 {
     compatible = "microchip,24aa025", "atmel,24c02";
     reg = <0x50>;

     #address-cells = <1>;
     #size-cells = <1>;
     eth1_addr: eth-mac-addr@FA {
       reg = <0xFA 0x06>;
     };
    };


};
```

Kernel

Start with **petalinux-config -c kernel**

Changes:

- RTC_DRV_ISL12022
- XILINX_GMII2RGMII

Rootfs

Start with **petalinux-config -c rootfs**

Changes:

- CONFIG_i2c-tools=y
- CONFIG_auto-login=y
- CONFIG_RTC_DRV_ISL12022=y
- util-linux-blkid =y
- util-linux-mount =y
- uutil-linux-umount =y
- usbutils = y

FSBL patch (alternative for vitis fsbl trenz patch)

See "<project folder>\os\petalinux\project-spec\meta-user\recipes-bsp\embeddedsw"

> ⚠ te_* files are identical to files in "<project folder>\sw_lib\sw_apps\zynqmp_fsbl\src" except for the PLL files (SI5345) which depend on PLL revision. The PLL files may have to be copied again manually into the appropriate petalinux folder "<project folder>\os\petalinux\project-spec\meta-user\recipes-bsp\embeddedsw\fsbl-firmware\git\lib\sw_apps\zynqmp_fsbl\src"

Applications

Not included.

Additional Software

No additional software is needed.

SI5338

File location "<project folder>\misc\Si5338\Si5338-*.slabtimeproj"

General documentation how you work with this project will be available on Si5338

App. A: Change History and Legal Notices

Document Change History

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

Date	Document Revision	Authors	Description
			• 2022.2 update + features

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#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.

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.

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]	atlassian . confluence.core. ContentEntityObject]	atlassian . confluence.core. ContentEntityObject]	
2023-10-10	v.5	Waldemar Hanemann	• 2022.2 update
2018-10-10	v.0.9	Waldemar Hanemann	• 2018.2 initial release
--	all	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#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.confluen

		ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	

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.proxy279.$Proxy4022#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]