

TEB0911 Test Board

Table of contents

- 1 [Table of contents](#)
- 2 [Overview](#)
 - 2.1 [Key Features](#)
 - 2.2 [Revision History](#)
 - 2.3 [Release Notes and Know Issues](#)
 - 2.4 [Requirements](#)
 - 2.4.1 [Software](#)
 - 2.4.2 [Hardware](#)
 - 2.5 [Content](#)
 - 2.5.1 [Design Sources](#)
 - 2.5.2 [Additional Sources](#)
 - 2.5.3 [Prebuilt](#)
 - 2.5.4 [Download](#)
- 3 [Design Flow](#)
- 4 [Launch](#)
 - 4.1 [Programming](#)
 - 4.1.1 [Get prebuilt boot binaries](#)
 - 4.1.2 [QSPI](#)
 - 4.1.3 [SD](#)
 - 4.1.4 [JTAG](#)
 - 4.2 [Usage](#)
 - 4.2.1 [Linux](#)
 - 4.2.2 [Vivado HW Manager](#)
- 5 [System Design - Vivado](#)
 - 5.1 [Block Design](#)
 - 5.1.1 [PS Interfaces](#)
 - 5.2 [Constrains](#)
 - 5.2.1 [Basic module constrains](#)
 - 5.2.2 [Design specific constrain](#)
- 6 [Software Design - Vitis](#)
 - 6.1 [Application](#)
 - 6.1.1 [zynqmp_fsbl](#)
 - 6.1.2 [zynqmp_fsbl_flash](#)
 - 6.1.3 [zynqmp_pmufw](#)
 - 6.1.4 [hello_teb0911](#)
 - 6.1.5 [u-boot](#)
- 7 [Software Design - PetaLinux](#)
 - 7.1 [Config](#)
 - 7.2 [U-Boot](#)
 - 7.3 [Device Tree](#)
 - 7.4 [Kernel](#)
 - 7.5 [Rootfs](#)
 - 7.6 [Applications](#)
 - 7.6.1 [startup](#)
 - 7.6.2 [webfwu](#)
- 8 [Additional Software](#)
 - 8.1 [SI5338](#)
 - 8.2 [SI5345](#)
- 9 [Appx. A: Change History and Legal Notices](#)
 - 9.1 [Document Change History](#)
 - 9.2 [Legal Notices](#)
 - 9.3 [Data Privacy](#)
 - 9.4 [Document Warranty](#)
 - 9.5 [Limitation of Liability](#)
 - 9.6 [Copyright Notice](#)
 - 9.7 [Technology Licenses](#)
 - 9.8 [Environmental Protection](#)
 - 9.9 [REACH, RoHS and WEEE](#)

Overview

Design example with Linux and MGT-CLK frequency monitoring over VIO.

Refer to <http://trenz.org/teb0911-info> for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- SD
- ETH
- MAC from EEPROM
- USB
- I2C
- PCIe
- DP
- FMeter
- LED
- Modified FSBL for SI5338 and SI5345 programming
- Special FSBL for QSPI programming

Revision History

Date	Vivado	Project Built	Authors	Description
2020-06-03	2019.2	TEB0911-test_board-vivado_2019.2-build_12_20200603131549.zip TEB0911-test_board_noprebuilt-vivado_2019.2-build_12_20200603131603.zip	John Hartfiel	<ul style="list-style-type: none">• bugfix usb3• add nvme driver
2020-03-25	2019.2	TEB0911-test_board_noprebuilt-vivado_2019.2-build_8_20200325084706.zip TEB0911-test_board-vivado_2019.2-build_8_20200325084633.zip	John Hartfiel	<ul style="list-style-type: none">• script update
2020-02-24	2019.2	TEB0911-test_board_noprebuilt-vivado_2019.2-build_6_20200224080741.zip TEB0911-test_board-vivado_2019.2-build_6_20200224080728.zip	John Hartfiel	<ul style="list-style-type: none">• bugfix PL Design (all MGT buffer enabled)
2020-02-13	2019.2	TEB0911-test_board_noprebuilt-vivado_2019.2-build_5_20200213114513.zip TEB0911-test_board-vivado_2019.2-build_5_20200213112730.zip	John Hartfiel	<ul style="list-style-type: none">• 2019.2 update• new assembly variants• Vitis support• FSBL SI programming procedure update• petalinux device tree and u-boot update• reduced DDR speed (see Xilinx Datasheet)
2018-11-26	2018.2	TEB0911-test_board_noprebuilt-vivado_2018.2-build_03_20181126132622.zip TEB0911-test_board-vivado_2018.2-build_03_20181126132607.zip	John Hartfiel	<ul style="list-style-type: none">• new assembly variant• add init.sh
2018-07-20	2018.2	TEB0911-test_board_noprebuilt-vivado_2018.2-build_02_20180719153443.zip TEB0911-test_board-vivado_2018.2-build_02_20180719153429.zip	John Hartfiel	<ul style="list-style-type: none">• 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	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
TEB0911-02-ES1	es1_4gb	REV02 REV01	4GB	64MB	4GB	SODIMM_KVR24S17S8/8	Not longer supported by vivado
TEB0911-04-09EG1E	9eg_1e_8gb	REV04 REV03 REV02	8GB	64MB	8GB	SODIMM_CT8G4SFS824A	
TEB0911-04-15EG1E	15eg_1e_8gb	REV04	8GB	128MB	8GB	SODIMM_CT8G4SFS824A	
TEB0911-04-ZU9EG1A	9eg_1e_8gb	REV04	8GB	128MB	8GB	SODIMM_CT8G4SFS824A	
TEB0911-04-ZU15EGA	15eg_1e_8gb	REV04	8GB	128MB	8GB	SODIMM_CT8G4SFS824A	
TEB0911-04-9BEX1FA	9eg_1e_8gb	REV04	8GB	128MB	8GB	SODIMM_CT8G4SFS824A	
TEB0911-04-BBEX1FA	15eg_1e_8gb	REV04	8GB	128MB	8GB	SODIMM_CT8G4SFS824A	

Hardware Modules

Additional HW Requirements:

Additional Hardware	Notes
DDR4	example configured for CT8G4SFS824A

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	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
SI5338	<design name>/misc/SI5338	SI5338 Project with current PLL Configuration
SI5345	<design name>/misc/SI5345	SI5345 Project with current PLL Configuration
init.sh	<design name>/misc/init_script	Additional Initialization Script for Linux

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
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/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- [TEB0911 "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 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:

```
C:\WINDOWS\system32\cmd.exe
B:\Design\cores\2018.2\design\TEB0911\test_board>setlocal
-- Set design paths --
-- Run Design with: _create_win_setup
-- Use Design Path: B:\Design\cores\2018.2\design\TEB0911\test_board\
-- TE Reference Design --
-- (c) Go to CMD-File Generation (Manual setup)
-- (d) Go to Documentation (Web Documentation)
-- (x) Exit Batch (nothing is done!)
-- (0) Create minimum setup of CMD-Files and exit Batch
-- (1) Create maximum setup of CMD-Files and exit Batch
Select (ex.: '0' for min setup):
```

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_gui mode.cmd"
Note: Select correct one, see also [TE Board Part Files](#)
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
6. Create Linux (uboot.elf and image.ub) with exported XSA
 - a. XSA 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](#)
 - i. Use TE Template from /os/petalinux
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
 - a. "prebuilt\os\petalinux\<ddr size>" or "prebuilt\os\petalinux\<short name>"
8. 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 manually in case GUI is used. See [Vitis](#)

Launch

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 folder>/_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_gui mode.cmd`" or if not created, create with "`vivado_create_project_gui mode.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_teb0911`" possible
4. Copy image.ub and optional misc/sd/init.sh on SD-Card
 - use files from (`<project folder>/_binaries_<Artikel 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`
5. Insert SD-Card

SD

1. Copy image.ub, Boot.bin and misc/sd/init.sh on SD-Card.
 - use files from (`<project folder>/_binaries_<Artikel 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 CPLD Firmware, see [SC0911 CPLD#BootMode](#)
3. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section [TEB0911 Test Board#Programming](#)
2. Connect UART USB (same as FPGA JTAG)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
4. (Optional) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional) Connect DisplayPort Monitor (List of usable Monitors: <https://www.xilinx.com/support/answers/68671.html>)
6. (Optional) Connect Network Cable
7. Power On PCB
Note: 1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD into OCM, 2. FSBL loads ATF(bl31.elf) and U-boot from SD/QSPI into DDR, 3. U-boot load Linux from SD 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. 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. I2C 0 Bus type: `i2cdetect -y -r 0`
 - b. ETH0 works with `udhcpc`
 - c. USB type "lsusb" or connect USB device
 - d. PCIe type "lspci"

Vivado HW Manager

Control:

- User LED Control (D16, D15)

Monitoring:

- MGT CLK Measurement:
 - Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).Set radix from VIO signals to unsigned integer.Note: Frequency Counter is inaccurate and displayed unit is Hz
 - Default B229_CLK1: 78,8MHz, B128_CLK1: 150MHz, B129_CLK1: 175MHz, B130_CLK1: 200MHz, B228_CLK1: 125MHz, B230_CLK1: 100MHz

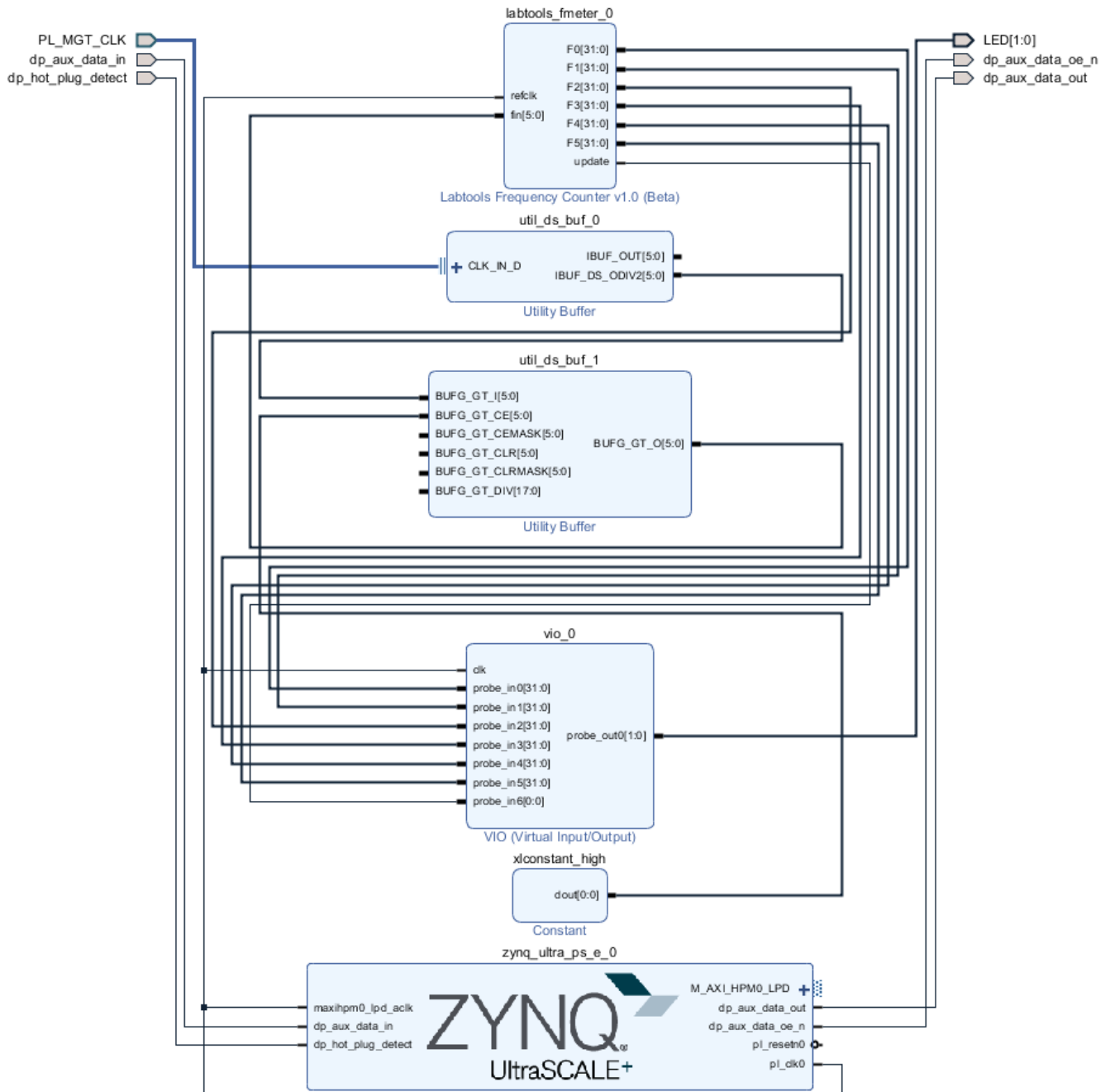
Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/251633001E...	Open
xczu15_0 (2)	Programmed
SystemMon (System Monitor)	
hw_vio_1 (zusys_i/vio_0)	OK
arm_dap_1 (1)	N/A
SystemMon (System Monitor)	

Name	Value	Activity	Direction	VIO
> zusys_ifm_B229_CLK1[31:0]	[U] 78800229	↑	Input	hw_vio_1
> zusys_ifm_B128_CLK1[31:0]	[U] 150000435	↓	Input	hw_vio_1
> zusys_ifm_B129_CLK1[31:0]	[U] 175000509	↓	Input	hw_vio_1
> zusys_ifm_B130_CLK1[31:0]	[U] 200000581	↓	Input	hw_vio_1
> zusys_ifm_B228_CLK1[31:0]	[U] 125000363	↓	Input	hw_vio_1
> zusys_ifm_B230_CLK1[31:0]	[U] 100000291	↑	Input	hw_vio_1
zusys_i/labtools_fmter_0_update	[B] 0	↓	Input	hw_vio_1
> zusys_i/LED[1:0]	[H] 3		Output	hw_vio_1

Vivado Hardware Manager

System Design - Vivado

Block Design



Block Design

PS Interfaces

Activated interfaces:

Type	Note
DDR	SODIMM, setting depends on used memory
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
PJTAG0	MIO
UART0	MIO
GPIO0	MIO
SWDT0..1	
TTC0..3	
GEM3	MIO
USB0	MIO/GTP
PCIe	MIO/GTP
DisplayPort	EMIO/GTP

PS Interfaces

Constraints

Basic module constraints

_i_bitgen_common.xdc

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

Design specific constrain

i_TEB0911.xdc

```
# GT Clocks
#B128-1
set_property PACKAGE_PIN N27 [get_ports {PL_MGT_CLK_clk_p[0]}]
#B129-1
set_property PACKAGE_PIN J27 [get_ports {PL_MGT_CLK_clk_p[1]}]
#B228-1
set_property PACKAGE_PIN J8 [get_ports {PL_MGT_CLK_clk_p[2]}]
#B130-1
set_property PACKAGE_PIN E27 [get_ports {PL_MGT_CLK_clk_p[3]}]
#B229-1
set_property PACKAGE_PIN E8 [get_ports {PL_MGT_CLK_clk_p[4]}]
#B230-1
set_property PACKAGE_PIN B10 [get_ports {PL_MGT_CLK_clk_p[5]}]

## DP
set_property PACKAGE_PIN AB1 [get_ports dp_aux_data_in]
set_property PACKAGE_PIN V9 [get_ports dp_hot_plug_detect]
set_property PACKAGE_PIN AA8 [get_ports dp_aux_data_out]
set_property PACKAGE_PIN AA3 [get_ports dp_aux_data_oe_n]
set_property IOSTANDARD LVCMOS18 [get_ports dp_*]
## LED
set_property PACKAGE_PIN K14 [get_ports {LED[0]}]
set_property PACKAGE_PIN K10 [get_ports {LED[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {LED*}]
```

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_
 - SI5338 and SI5345 Configuration
 - PCIe reset

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

zynqmp_pmufw

Xilinx default PMU firmware.

hello_teb0911

Hello TEB0911 is a Xilinx Hello World example as endless loop instead of one console output.

u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI 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:

- SUBSYSTEM_PRIMARY_SD_PSU_SD_1_SELECT
- CONFIG_SUBSYSTEM_ETHERNET_PSU_ETHERNET_3_MAC=""

U-Boot

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

Changes:

- CONFIG_ENV_IS_NOWHERE=y
- # CONFIG_ENV_IS_IN_SPI_FLASH is not set
- CONFIG_I2C_EEPROM=y
- CONFIG_ZYNQ_GEM_I2C_MAC_OFFSET=0xFA
- CONFIG_SYS_I2C_EEPROM_ADDR=0x54
- CONFIG_SYS_I2C_EEPROM_BUS=5
- CONFIG_SYS_EEPROM_SIZE=256
- CONFIG_SYS_EEPROM_PAGE_WRITE_BITS=0

- CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS=0
- CONFIG_SYS_I2C_EEPROM_ADDR_LEN=1
- CONFIG_SYS_I2C_EEPROM_ADDR_OVERFLOW=0

Change platform-top.h

Device Tree

```
/include/ "system-conf.dtsi"
/ {
    chosen {
        xlnx,eeprom = &eeprom;
    };
};

/* USB */

&dwc3_0 {
    status = "okay";
    dr_mode = "host";
    snps,usb3_lpm_capable;
    snps,dis_u3_susphy_quirk;
    snps,dis_u2_susphy_quirk;
    phy-names = "usb2-phy", "usb3-phy";
    phys = <&lane1 4 0 1 100000000>;
    maximum-speed = "super-speed";
};

/* QSPI */

&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 */

&gem3 {
    phy-handle = <&phy0>;
    phy0: phy0@1 {
        device_type = "ethernet-phy";
        reg = <1>;
    };
};
```

```

/* SD1 */

&sdhci1 {
    // disable-wp;
    no-1-8-v;
};

&i2c0 {
    i2cswitch@76 { // I2C Switch U13
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x76>;
        i2c-mux-idle-disconnect;

        i2c@2 { // FMCD (/dev/i2c-3)
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <2>;
        };
        i2c@3 { // FMCE (/dev/i2c-4)
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <3>;
        };
        i2c@4 { // FMCB (/dev/i2c-5)
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <4>;
        };
        i2c@5 { // FMCC (/dev/i2c-6)
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <5>;
        };
        i2c@6 { // PLL (/dev/i2c-7)
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <6>;

            si570_2: clock-generator3@5d {
                #clock-cells = <0>;
                compatible = "silabs,si570";
                reg = <0x5d>;
                temperature-stability = <50>;
                factory-fout = <156250000>;
                clock-frequency = <78800000>;
            };
        };
    };

    i2cswitch@77 { // I2C Switch U37
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x77>;
        i2c-mux-idle-disconnect;
    };
};

```

```

i2c@0 { // SFP2 (/dev/i2c-9)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0>;
};
i2c@1 { // FMCA (/dev/i2c-10)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <1>;
};
i2c@2 { // FMCF (/dev/i2c-11)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <2>;
};
i2c@3 { // SFP0 (/dev/i2c-12)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <3>;
};
i2c@4 { // SFP1 (/dev/i2c-13)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <4>;
};
i2c@5 { // MEM (/dev/i2c-14)
    // Low frequency to work with CPLD
    clock-frequency = <100000>;
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <5>;
    eeprom: eeprom@54 {
        compatible = "atmel,24c08";
        reg = <0x54>;
    };
};
i2c@6 { // DDR4 (/dev/i2c-15)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <6>;
};
i2c@7 { // USBH (/dev/i2c-16)
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <7>;
};
};
};

```

Kernel

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

Changes:

- # CONFIG_CPU_IDLE is not set (only needed to fix JTAG Debug issue)
- # CONFIG_CPU_FREQ is not set (only needed to fix JTAG Debug issue)

- CONFIG_EDAC_CORTEX_ARM64=y (only needed to fix JTAG Debug issue)
- CONFIG_NVME_CORE=y
- CONFIG_BLK_DEV_NVME=y
- # CONFIG_NVME_MULTIPATH is not set
- CONFIG_NVME_TARGET=y
- # CONFIG_NVME_TARGET_LOOP is not set
- # CONFIG_NVME_TARGET_FC is not set
- CONFIG_NVM=y
- CONFIG_NVM_PBLK=y
- CONFIG_NVM_PBLK_DEBUG=y

Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_packagegroup-petalinux-utils(util-linux,cpufrequtils,bridge-utils,mtd-utils,usbutils,pciutils,canutils,i2c-tools,smartmontools,e2fsprogs)

Applications

See: `\os\petalinux\project-spec\meta-user\recipes-apps\`

startup

Script App to load init.sh from SD Card if available.

webfwu

Webserver application accembles for Zynq access. Need busybox-httpd

Additional Software

No additional software is needed.

SI5338

File location <design name>/misc/SI5338/SI5338-*.slabtimeproj

General documentation how you work with these project will be available on [SI5338](#)

SI5345

File location <design name>/misc/SI5345/SI5345-RevD-0911-Project.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
<div>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]</div>	<div>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]</div>	<div>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]</div>	<ul style="list-style-type: none">• typo correction
2020-06-03	v.10	John Hartfiel	<ul style="list-style-type: none">• Design update (bugfix)
2020-03-25	v.9	John Hartfiel	<ul style="list-style-type: none">• script update
2020-02-24	v.8	John Hartfiel	<ul style="list-style-type: none">• Design update (bugfix)

2020-02-13	v7	John Hartfiel	<ul style="list-style-type: none"> • new assembly variants • Release 2019.2
2019-02-07	v.6	John Hartfiel	<ul style="list-style-type: none"> • some notes
2018-11-26	v.5	John Hartfiel	<ul style="list-style-type: none"> • new assembly variant • documenta- tion style update
2018-07-20	v.4	John Hartfiel	<ul style="list-style-type: none"> • 2018.2 release

--	all	<div> <p>Error rendering macro 'page-info'</p> <p>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]</p> </div>	--
----	-----	---	----

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

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.

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.

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]`