TE0729 Test Board

Table of contents

Overview

TE0729 Basic-System with Watchdog example via VIO Interface.
Refer to http://trenz.org/te0729-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- SD
- ETH (1 x 1 GBit, 2 x 100 MBit)
- USB
- I2C
- RTC
- Watchdog Test example over VIO
- Modified FSBL to select optional eMMC instead of SD
- Special FSBL for QSPI programming

Revision History

Date	Vivado	Project Built	Authors	Description
2020-10-01	2019.2	TE0729-test_board_noprebuilt-vivado_2019.2-build_15_20201001141241.zip TE0729-test_board-vivado_2019.2-build_15_20201001141230.zip	John Hartfiel	• Vitis update • FSBL template update • Petalinux webfwu, interface features • uboot doen't check qspi for enviroment • new Script Features
2018-07-16	2018.2	TE0729-test_board-vivado_2018.2-build_02_20180716161110.zip TE0729-test_board_noprebuilt-vivado_2018.2-build_02_20180716161138.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	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
TE0729-02-2IF	2if_512mb	REV02, REV01	512MB	32MB		2IF-K is the same with head sink	
TE0729-02-2IR	2ir_512mb	REV02	512MB	32MB	PL ETHs, RTC are no assembled		
TE0729-02-2IRA	2ir_512mb	REV02	512MB	32MB	PL ETHs, RTC are no assembled	ISSI Flash	

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TEB0729	Used as reference carrier.

Hardware Carrier

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

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
init.sh	<design name>/misc/sd/	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 prebult 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:

- TE0729 "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.

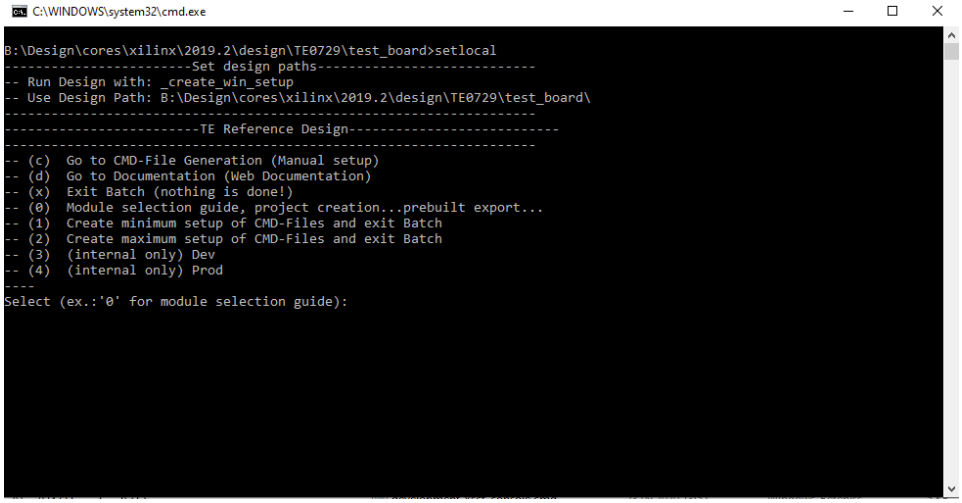
See also:

- 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:



2. Press 0 and enter to start "Module Selection Guide"
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project (follow instruction of the product selection guide), settings file will be configured automatically during this process)
    a. (optional for manual changes) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"
    Note: Select correct one, see alsoTE Board Part Files
5. Create XSA and export to prebuilt folder
    a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
    Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder
6. 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 manuelly 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 foler>/_binaries_<Artikel Name>) with subfolder (boot_<app name>) for different applications will be generated

QSPI

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

1. Connect JTAG and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"
3. Type on Vivado TCL Console: TE::pr_program_flash -swapp u-boot
   Note: To program with SDK/Vivado GUI, use special FSBL (zynqmp_fsbl_flash) on setup
   optional "TE::pr_program_flash -swapp hello_te0820" possible
4. Copy image.ub on SD-Card
   - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries
   - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
5. Insert SD-Card

SD

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

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TE0729 Test Board#Programming
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode
   Note: See TRM of the Carrier, which is used.
4. Power On PCB
   Note: 1. Zynq Boot ROM loads FSBL from SD into OCM, 2. FSBL loads U-boot from SD into DDR, 3. U-boot load Linux from SD into DDR

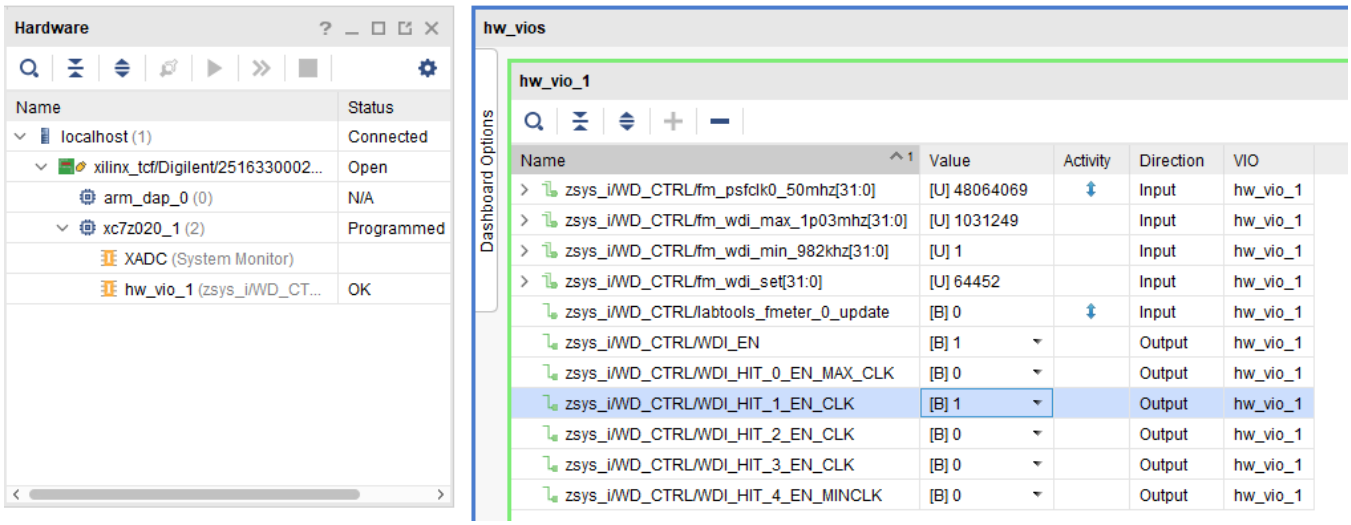
Linux

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. I2C 0 Bus type: i2cdetect -y -r 1
   c. ETH0 works with udhcpc
   d. ETH1 works with udhcpc
   e. ETH2 works with udhcpc
   f. RTC check: dmesg | grep rtc
   g. USB: insert USB Stick or lsusb
4. Option Features
   a. Webserver to get access to Zynq
      i. insert IP on web browser to start web interface
   b. init.sh scripts
      i. add init.sh script on SD, content will be load automatically on startup (template included in ./misc/SD)

Vivado HW Manager

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

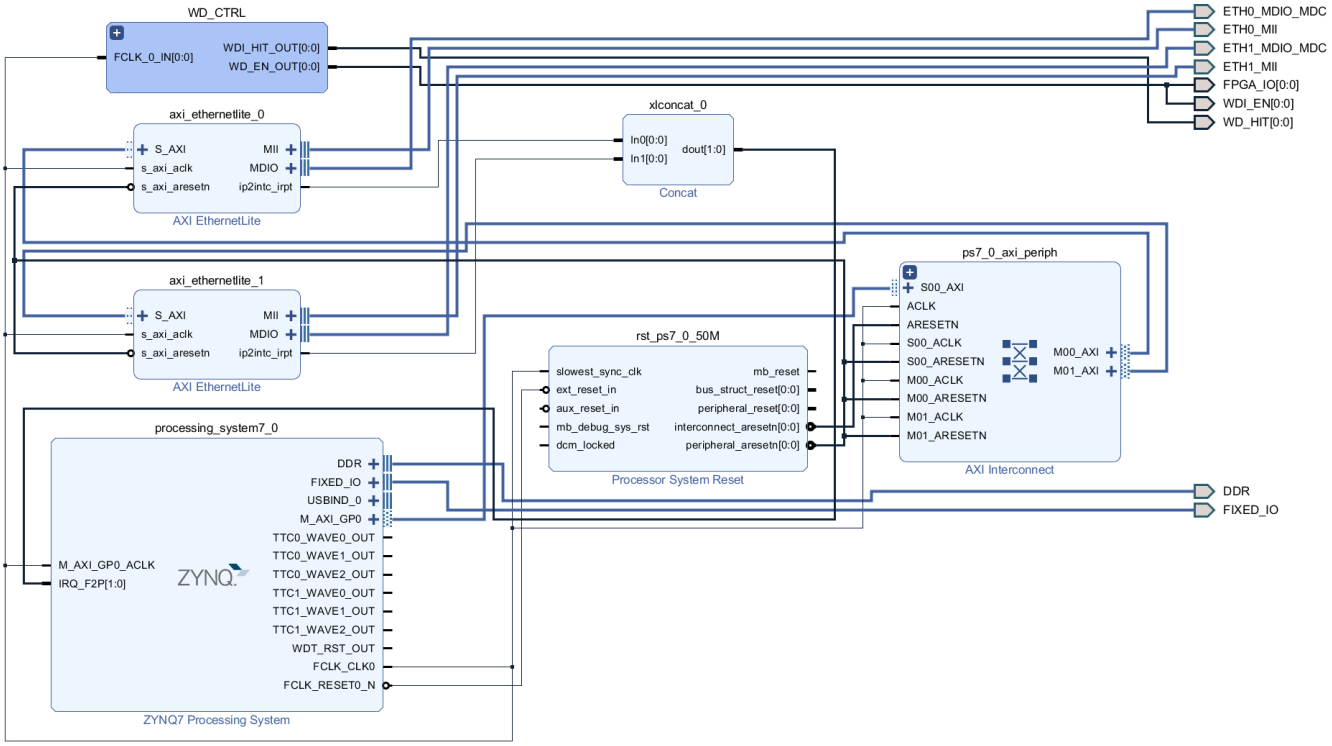
- Control:
  - "WDI_EN" and "WDI_HIT_*_EN_CLK" enables FPGA watchdog control.
  - Force WD to system reboot:
    1. Check on Hardware window VIO status is ok. (right click on vio symbol and click "commit output values to VIO core" for update).
    2. Enable one of the "WDI_HIT_*_EN_CLK" signals
    3. Enable "WDI_EN"
    4. To force system to reboot, disable WDI_HIT clocks.
- Monitoring:
  - Set radix for "fm_*" signals to unsigned integer to see frequ in Hz.
  - "fm_*" shows some clk frequencies (unit Hz). Note: inaccurate Reference CLK is used for frequency measurement.



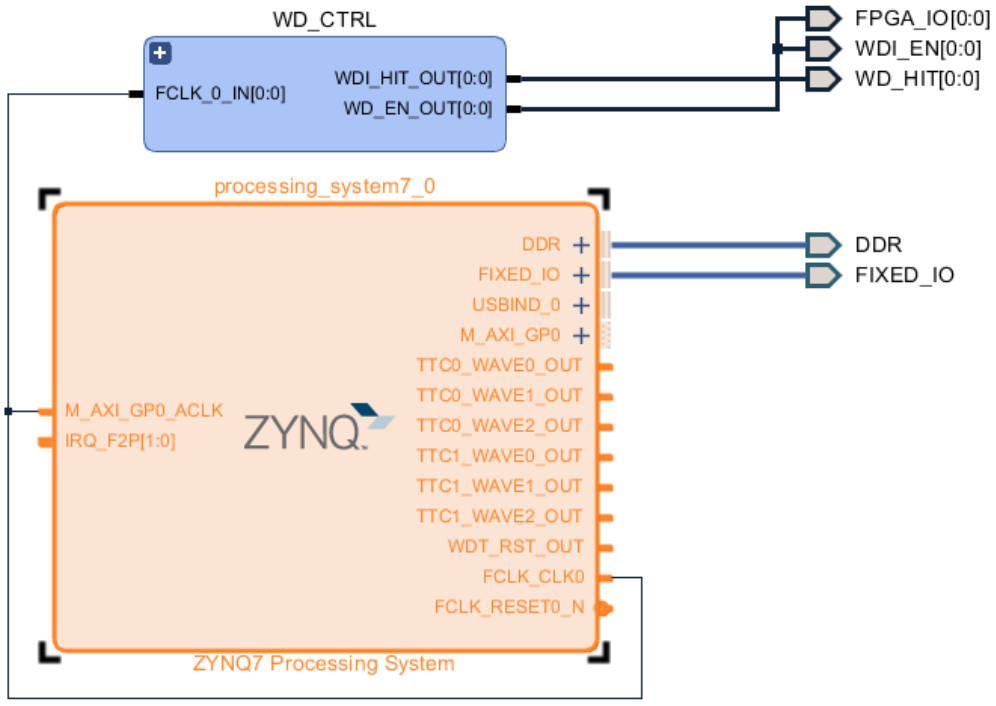
Vivado Hardware Manager

System Design - Vivado

Block Design


WD_CTRL
FCLK_0_IN[0:0]
WDI_HIT_OUT[0:0]
WD_EN_OUT[0:0]
ETH0_MDIO_MDC
ETH0_MII
ETH1_MDIO_MDC
ETH1_MII
FPGA_IO[0:0]
WDI_EN[0:0]
WD_HIT[0:0]
axi_ethernetlite_0
S_AXI
s_axi_aclk
s_axi_aresetn
MII
MDIO
ip2intc_irpt
AXI EthernetLite
xlconcat_0
In0[0:0]
In1[0:0]
dout[1:0]
Concat
axi_ethernetlite_1
S_AXI
s_axi_aclk
s_axi_aresetn
MII
MDIO
ip2intc_irpt
AXI EthernetLite
ps7_0_axi_periph
S00_AXI
ACLK
ARESETN
S00_ACLK
S00_ARESETN
M00_ACLK
M00_ARESETN
M01_ACLK
M01_ARESETN
M00_AXI
M01_AXI
AXI Interconnect
rst_ps7_0_50M
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
processing_system7_0
DDR
FIXED_IO
USBIND_0
M_AXI_GP0
TTC0_WAVE0_OUT
TTC0_WAVE1_OUT
TTC0_WAVE2_OUT
TTC1_WAVE0_OUT
TTC1_WAVE1_OUT
TTC1_WAVE2_OUT
WDT_RST_OUT
FCLK_CLK0
FCLK_RESET0_N
M_AXI_GP0_ACLK
IRQ_F2P[1:0]
ZYNQ
ZYNQ7 Processing System
DDR
FIXED_IO

R Variant:



Block Design

PS Interfaces

Type	Note
Typ	Note
DDR	
QSPI	MIO
SD0	MIO
I2C0	MIO
I2C1	MIO
UART0	MIO
GPIO0	MIO
SWDT0	

TTC0..1	
ETH00	MIO
USB0	MIO
PL-PS IRQ	

PS Interfaces

Constrains

Basic module constrains

_i_bitgen_common.xdc

```
#
# Common bitgen related settings
#

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

_i_unused_io.xdc

```
#
# Set unused pin pullup: PULLNONE, PULLUP, PULLDOWN
#

set_property BITSTREAM.CONFIG.UNUSEDPIN PULLNONE [current_design]

#set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
#set_property BITSTREAM.CONFIG.UNUSEDPIN PULLDONE [current_design]
```

Design specific constrain

_i_io.xdc

```
set_property PACKAGE_PIN F16 [get_ports {FPGA_IO[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {FPGA_IO[0]}]
set_property PACKAGE_PIN H15 [get_ports {WDI_EN[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {WDI_EN[0]}]
set_property PACKAGE_PIN R15 [get_ports {WD_HIT[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {WD_HIT[0]}]
```

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

Source location: \sw_lib\sw_apps

zynq_fsbl

TE modified 2019.2 FSBL

General:

- Modified Files:main.c, fsbl_hooks.h/.c (search for 'TE Mod' on source code)
- Add Files: te_fsbl_hooks.h/.c(for hooks and board)\n\
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- Add Files: all TE Files start with te_*
    - Optional define for eMMC selection with FSBL (default SD selected)
        - uncomment #define USE_EMMC on fsbl_hooks.c to select eMMC instead of SD

zynq_fsbl_flash

TE modified 2019.2 FSBL

General:

- Modified Files: main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

hello_te0729

Hello TE0729 is a Xilinx Hello World example as endless loop instead of one console output.

u-boot

U-Boot.elf is generated with PetaLinux. VitisI 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:

- No changes.

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=0x50
- CONFIG_SYS_I2C_EEPROM_BUS=0
- 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

Note: for R assembly variant, remove ETH1, ETH2 and RTC

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

/* 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 */
&gem0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <0>;
        };
```

```
    };
};

/* AXI ETH PHY0 */

&axi_ethernetlite_0 {
    local-mac-address = [00 0a 35 00 22 02];
    phy-handle = <&phy1>;
    xlnx,has-mdio = <0x1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy1: phy@1 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};

/* AXI ETH PHY1 */
&axi_ethernetlite_1 {
    local-mac-address = [00 0a 35 00 22 03];
    phy-handle = <&phy2>;
    xlnx,has-mdio = <0x1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy2: phy@1 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};


/* RTC */
&i2c0 {
    rtc@6F {        // Real Time Clock
       compatible = "isl12022";
       reg = <0x6F>;
   };
  //MAC EEPROM
  eeprom: eeprom@50 {
    compatible = "atmel,24c08";
    reg = <0x54>;
  };
};

/* USB PHY */

/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        //compatible = "usb-nop-xceiv";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};
```

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

Kernel

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

Changes:

- CONFIG_RTC_DRV_ISL12022=y

Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_usbutils=y

Applications

startup

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

webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

Core

init-ifupdown

Enable dhcp for ETH1 and ETH2

Additional Software

No additional software is needed.

Appx. A: Change History and Legal Notices

Document Change History

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

Date	Document Revision	Authors	Description
Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$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]	**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]	• 2019.2 Release
2018-07-06	v.5	John Hartfiel	• 2018.2 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.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	

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]