TE0820 TD TEF1002

Table of contents

Overview

ZynqMP PS Design with Linux Example and simple frequency counter to measure SI5338 Reference CLK with Vivado HW-Manager. Only for use with TEF1002 Carrier.
Wiki Resources page: http://trenz.org/te0820-info

Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- PCIe (endpoint)
- SATA
- SD
- ETH
- USB
- I2C
- RTC
- FMeter
- MAC from EEPROM
- User LED (PCB REV03 only)
- Modified FSBL for SI5338 programming
- Special FSBL for QSPI programming

Revision History

Date	Vivado	Project Built	Authors	Description
2020-03-25	2019.2	TE0820-TD_TEF1002_noprebuilt-vivado_2019.2-build_8_20200325084054.zip TE0820-TD_TEF1002-vivado_2019.2-build_8_20200325084033.zip	Martin Rohrmüller/John Hartfiel	• script update • Board Part update (minor changes)
2020-02-14	2019.2	TE0820-TD_TEF1002_noprebuilt-vivado_2019.2-build_6_20200217120248.zip TE0820-TD_TEF1002-vivado_2019.2-build_6_20200217120209.zip	Martin Rohrmüller	• initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
-	-	-	-

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
~~TE0820-ES1~~	~~es1~~	~~REV01~~	~~1GB~~	~~64MB~~	~~4GB~~	~~NA~~	Not longer supported by vivado
TE0820-02-02EG-1E	2eg_1e_1gb	REV02	1GB	64MB	4GB	NA	NA
TE0820-02-02EG-1E3	2eg_1e_1gb	REV02	1GB	64MB	4GB	2.5 mm connectors	NA
TE0820-02-02CG-1E	2cg_1e_1gb	REV02	1GB	64MB	4GB	NA	NA
TE0820-02-03EG-1E	3eg_1e_1gb	REV02	1GB	64MB	4GB	NA	NA
TE0820-02-03EG-1E3	3eg_1e_1gb	REV02	1GB	64MB	4GB	2.5 mm connectors	NA
TE0820-02-03CG-1E	3cg_1e_1gb	REV02	1GB	64MB	4GB	NA	NA
TE0820-02-02EG-1EA	2eg_1e_1gb	REV02	1GB	128MB	4GB	NA	NA
TE0820-02-02EG-1EL	2eg_1e_1gb	REV02	1GB	128MB	4GB	2.5 mm connectors	NA
TE0820-02-02CG-1EA	2cg_1e_1gb	REV02	1GB	128MB	4GB	NA	NA
TE0820-02-03EG-1EA	3eg_1e_1gb	REV02	1GB	128MB	4GB	NA	NA
TE0820-02-03EG-1EL	3eg_1e_1gb	REV02	1GB	128MB	4GB	2.5 mm connectors	NA
TE0820-02-03CG-1EA	3cg_1e_1gb	REV02	1GB	128MB	4GB	NA	NA
TE0820-02-04CG-1EA	4cg_1e_1gb	REV02	1GB	128MB	4GB	NA	NA
TE0820-03-04EV-1EA	4ev_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-02CG-1EA	2cg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-02EG-1EA	2eg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-02EG-1EL	2eg_1e_2gb	REV03	2GB	128MB	4GB	2.5 mm connectors	NA
TE0820-03-03CG-1EA	3cg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-04CG-1EA	4cg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-03EG-1EA	3eg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-03EG-1EL	3eg_1e_2gb	REV03	2GB	128MB	4GB	2.5 mm connectors	NA
TE0820-03-2AI21FA	2cg_1i_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-2BE21FL	2eg_1e_2gb	REV03	2GB	128MB	8GB	2.5 mm connectors	NA
TE0820-03-3AI210A	3cg_1i_2gb	REV03	2GB	128MB	0GB	NA	NA
TE0820-03-3BE21FA	3eg_1e_2gb	REV03	2GB	128MB	4GB	NA	NA
TE0820-03-3BE21FL	3eg_1e_2gb	REV03	2GB	128MB	4GB	2.5 mm connectors	NA
TE0820-03-02CG-1ED	2cg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-2AE21FA	2cg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-2BE21FA	2eg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-3AE21FA	3cg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-3AI21FA	3cg_1i_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-4AE21FA	4cg_1e_2gb	REV03	2GB	128MB	8GB	NA	NA

TE0820-03-4DE21FA	4ev_1e_2gb	REV03	2GB	128MB	8GB	NA	NA
TE0820-03-4DI21FA	4ev_1i_2gb	REV03	2GB	128MB	8GB	NA	NA

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TEF1002	• from REV02 onwards

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct typ
Cooler	It's recommended to use cooler on ZynqMP device

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
init.sh	<design name>/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 forVitis 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/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0820 "TD TEF1002" 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.

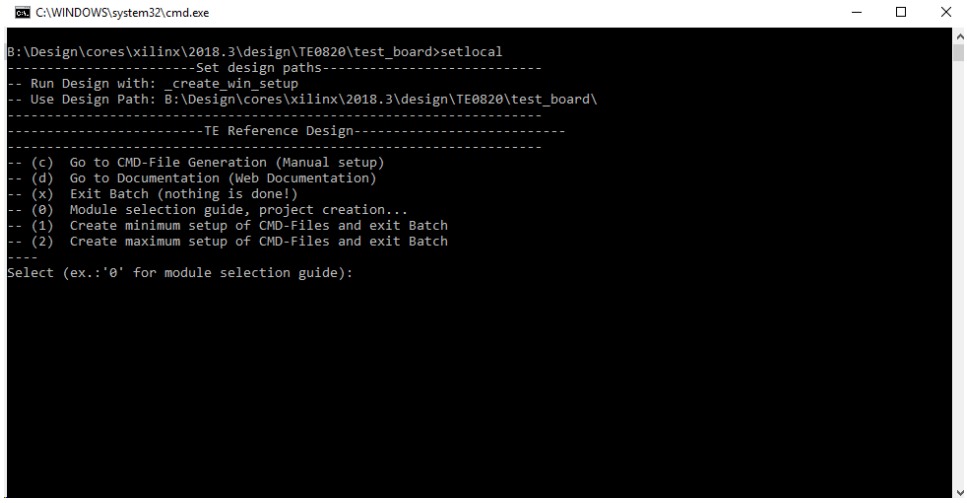
See also:

- 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/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 also TE 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 (bl31.elf, 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 (bl31.elf, uboot.elf and image.ub) to prebuilt folder
    a. "prebuilt\os\petalinux\<DDR size>" or "prebuilt\os\petalinux\<short name>"
    Notes: Scripts select "prebuilt\os\petalinux\<short name>", if exist, otherwise "prebuilt\os\petalinux\<DDR size"
8. Generate Programming Files with Vitis
    a. Run on Vivado TCL: TE::sw_run_vitis -all
    Note: Depending of PC performance this can take several minutes. Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv" and open Vitis
    b. (alternative) Start Vitis 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.

Note: Depending on CPLD Firmware and Boot Mode settings, QSPI boot with Linux image on SD or complete SD Boot is possible.

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_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_te0820" possible
4. Copy image.ub on SD-Card
   - use files from (<project foler>/_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

Use this description for CPLD Firmware with SD Boot selectable.

1. Copy image.ub and Boot.bin on SD-Card.
   - For correct prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
2. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TEF1002 Getting Started
   a. for PCIe insert TEF1002 in PCIe slot and connect PC PCIe 6 pin power connector
2. Connect JTAG/UART USB
3. Connect SATA device
4. Select Boot Mode (S3-1)
5. Power On PCB
   Note: 1. ZynqMP Boot ROM loads PMU Firmware and FSBL from SD/QSPI Flash into OCM, 2. FSBL loads ATF(bl31.elf) and 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. RTC check: dmesg | grep rtc
   c. ETH0 works with udhcpc
   d. USB type "lsusb" or connect USB2.0 device
   e. SATA device
      i. dmesg | grep -i sata | grep 'link up'
      ii. fdisk -l /dev/sd*
      iii. partitions are mounted to /run/media/sdXY: df -h
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

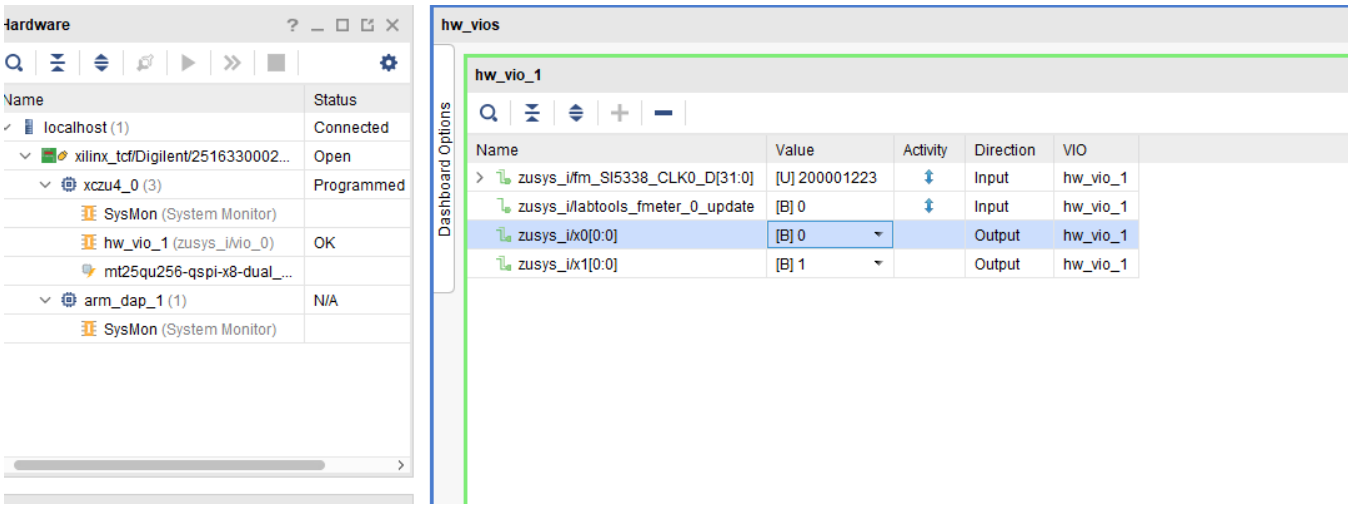
SI5338_CLK0 Counter:

1. Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
   a. Set radix from VIO signals to unsigned integer.
      Note: Frequency Counter is inaccurate and displayed unit is Hz

SI5338 CLK is configured to 200MHz by default.

PCB REV03 Design:

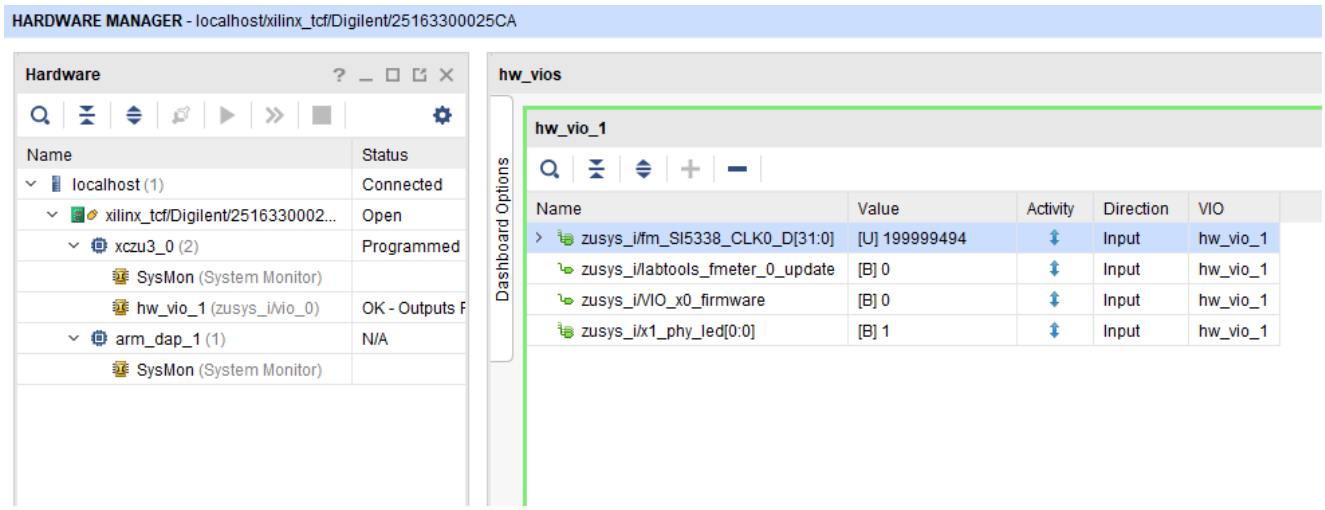
- User LED, see: TE0820 CPLD#LED



Vivado Hardware Manager

PCB REV01, REV02 Design:

- PHY LEDS, see: TE0820-REV01_REV02 CPLD#X0/X1Pin
- CPLD Firmware, see: TE0820-REV01_REV02 CPLD#X0/X1Pin

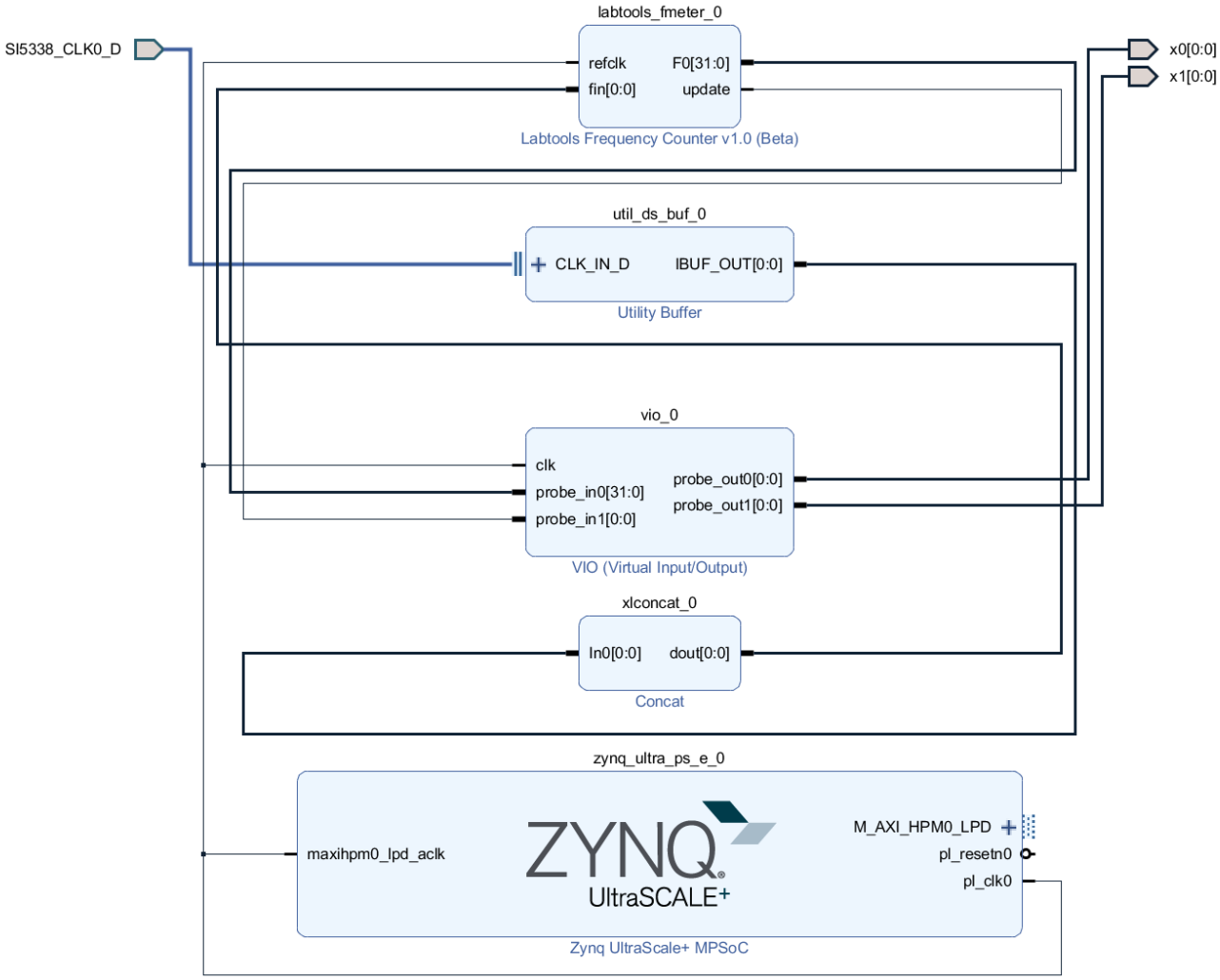


Vivado Hardware Manager PCB REV01,REV02

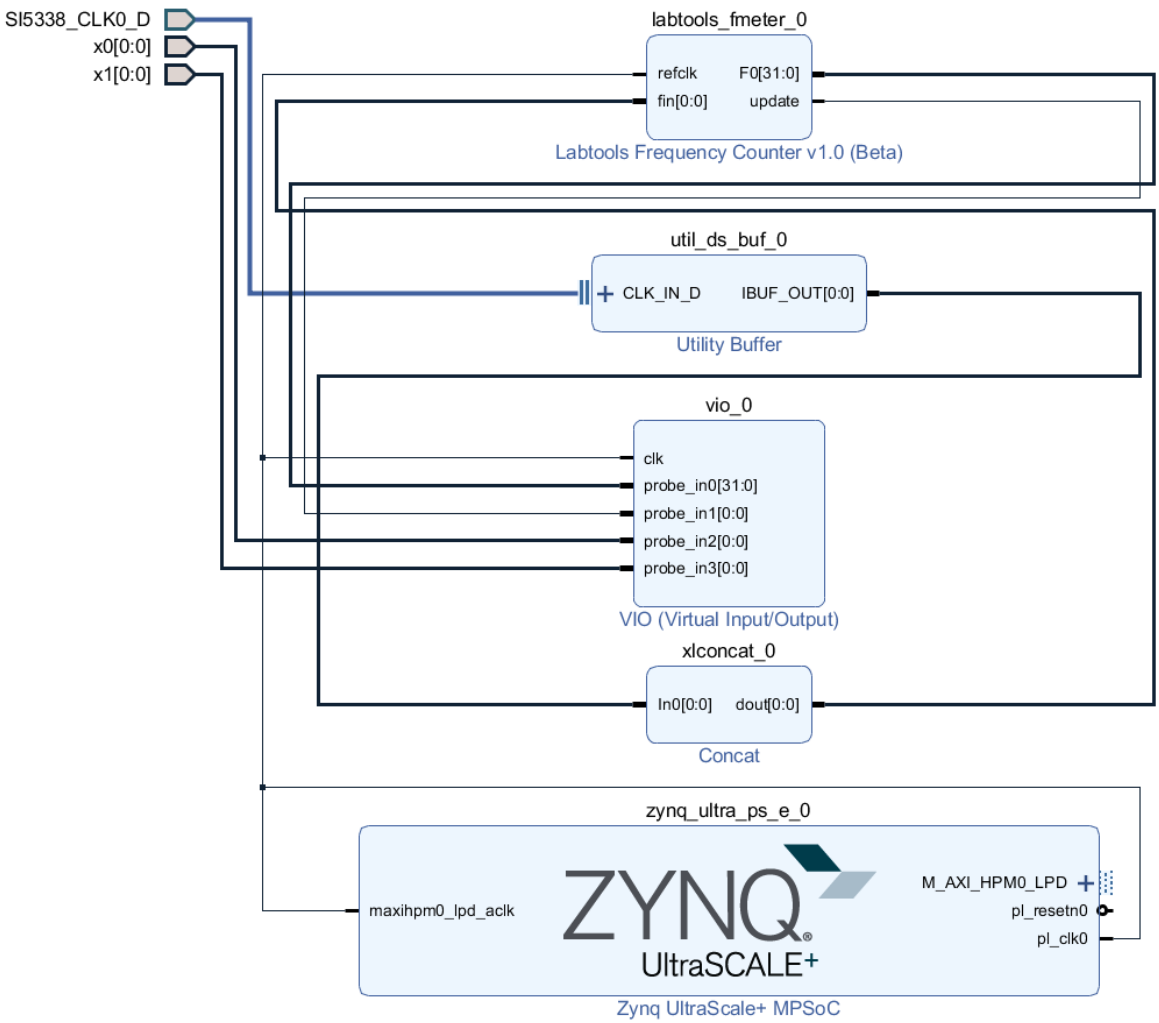
System Design - Vivado

Block Design

PCB REV03



PCB REV01 REV02



PS Interfaces

Activated interfaces:

Type	Note
DDR	
PCIe	GTR Lane0, 100MHz endpoint, RESET# MIO33
SATA	GTR Lane3, 125MHz
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
UART0	MIO

GPIO0	MIO
SWDT0..1	
TTC0..3	
GEM3	MIO
USB0	MIO, USB2 only

Constrains

Basic module constrains

_i_bitgen_common.xdc

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

Design specific constrain

_i_io.xdc

```
set_property PACKAGE_PIN K9 [get_ports {SI5338_CLK0_D_clk_p[0]}]
set_property IOSTANDARD LVDS [get_ports {SI5338_CLK0_D_clk_p[0]}]
set_property DIFF_TERM TRUE [get_ports {SI5338_CLK0_D_clk_p[0]}]

set_property PACKAGE_PIN H1 [get_ports {x0[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {x0[0]}]
set_property PACKAGE_PIN J1 [get_ports {x1[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {x1[0]}]
```

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

Template location: ./sw_lib/sw_apps/

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 Configuration
    - ETH+OTG Reset over MIO

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_te0820

Hello TE0820 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:

- CONFIG_SUBSYSTEM_PRIMARY_SD_PSU_SD_1_SELECT=y
- 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=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

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


/* SDIO */

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

/* ETH PHY */
&gem3 {

        status = "okay";
  ethernet_phy0: ethernet-phy@0 {
                compatible = "marvell,88e1510";
                device_type = "ethernet-phy";
                    reg = <1>;
        };
};
/* USB 2.0 */

/* USB  */
&dwc3_0 {
    status = "okay";
    dr_mode = "host";
    maximum-speed = "high-speed";
    /delete-property/phy-names;
    /delete-property/phys;
    /delete-property/snps,usb3_lpm_capable;
          snps,dis_u2_susphy_quirk;
          snps,dis_u3_susphy_quirk;
};

&usb0 {
    status = "okay";
    /delete-property/ clocks;
    /delete-property/ clock-names;
    clocks = <0x3 0x20>;
```

```
    clock-names = "bus_clk";
};



/* 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>;
    };
};

&i2c0 {
  eeprom: eeprom@50 {
     compatible = "atmel,24c08";
     reg = <0x50>;
  };
};
```

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

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)
- CONFIG_e2fsprogs=y (filesystems maintanace)
- CONFIG_e2fsprogs-dev=y (for filesystems maintanace)
- CONFIG_e2fsprogs-mke2fs=y (filesystems maintanace)
- CONFIG_e2fsprogs-dbg=y (filesystems maintanace)
- CONFIG_e2fsprogs-resize2fs=y (filesystems maintanace)
- CONFIG_e2fsprogs-tune2fs=y (filesystems maintanace)
- CONFIG_libss=y (for filesystems maintanace)
- CONFIG_libcomerr=y (for filesystems maintanace)
- CONFIG_libext2fs=y (filesystems maintanace)
- CONFIG_libe2p=y (filesystems maintanace)
- CONFIG_e2fsprogs-badblocks=y (filesystems maintanace)

Applications

startup

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

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

webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

Additional Software

SI5338

File location <design name>/misc/Si5338/Si5338-*.slabtimeproj

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

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.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$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	**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	• script update • Boart PArt update (Minor changes)

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]	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]	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]	
2020-03-11	v.6	Martin Rohrmüller	• initial release 2019.2

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]