TE0803 SK Demo1

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_te0803
    - 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
- 8 Additional Software
  - 8.1 SI5338
- 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

Refer to http://trenz.org/te0803-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- Linux Debian 9 (Stretch) or Linux Ubuntu 18.04 (Bionic Beaver)
- DisplayPort
- SD
- ETH (use EEPROM MAC)
- MAC from EEPROM
- PCIe
- USB
- I2C
- TEBF0808
- RGPIO
- SATA
- user LED access
- Modified FSBL for SI5338 programming
- Special FSBL for QSPI programming

Revision History

Date	Vivado	Project Built	Authors	Description
2020-05-08	2019.2	TE0803-SK_DEMO1_noprebuilt-vivado_2019.2-build_11_20200508143345.zip TE0803-SK_DEMO1-vivado_2019.2-build_11_20200508143327.zip	Mohsen Chamanbaz	• 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
SD Card Formatter		format SD Card
Win32 DiskImager		burn generated image on SD
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
~~TE0803-ES1~~	es1_2gb	REV01	2GB	64MB	NA	NA	Not longer supported by vivado
TE0803-01-02EG-1E	2eg_2gb	REV01	2GB	64MB	NA	NA	NA
TE0803-01-02CG-1E	2cg_2gb	REV01	2GB	64MB	NA	NA	NA
TE0803-01-03EG-1E	3eg_2gb	REV01	2GB	64MB	NA	NA	NA
TE0803-01-03CG-1E	3cg_2gb	REV01	2GB	64MB	NA	NA	NA
TE0803-01-02EG-1EA	2eg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-02CG-1EA	2cg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-03EG-1EA	3eg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-03CG-1EA	3cg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-02-03EG-1EB	3eg_4gb	REV02|REV01	4GB	128MB	NA	NA	NA
TE0803-01-04CG-1EA	4cg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-04EV-1EA	4ev_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-04EV-1E3	4ev_2gb	REV01	2GB	128MB	NA	1 mm connectors	NA
TE0803-01-04EG-1EA	4eg_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-04CG-1EB	4cg_2gb	REV01	2GB	256MB	NA	NA	NA
TE0803-01-05EV-1EA	5ev_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-01-05EV-1IA	5ev_i_2gb	REV01	2GB	128MB	NA	NA	NA
TE0803-02-04EV-1EB	4ev_4gb	REV02	4GB	128MB	NA	NA	NA
TE0803-02-04EV-1E3	4ev_4gb	REV02	4GB	128MB	NA	1 mm connectors	NA
TE0803-02-04EG-1E3	4eg_4gb	REV02	4GB	128MB	NA	1 mm connectors	NA
TE0803-03-2AE11-A	2cg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-2BE11-A	2eg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-3AE11-A	3cg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-3BE11-A	3eg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4AE11-A	4cg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4BE11-A	4eg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4BE21-L	4eg_4gb	REV03	4GB	128MB	NA	1 mm connectors	NA
TE0803-03-4BI21-A	4eg_i_4gb	REV03	4GB	128MB	NA	NA	NA
TE0803-03-4DE11-A	4ev_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4DE21-L	4ev_4gb	REV03	4GB	128MB	NA	1 mm connectors	NA
TE0803-03-4GE21-L	4eg_2_4gb	REV03	4GB	128MB	NA	1 mm connectors	NA
TE0803-03-5DE11-A	5ev_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-5DI21-A	5ev_i_4gb	REV03	4GB	128MB	NA	NA	NA
TE0803-03-3RI21-A	3eg_li_4gb	REV03	4GB	128MB	NA	NA	NA
TE0803-03-3BI21-A	3eg_i_4gb	REV03	4GB	128MB	NA	NA	NA
TE0803-03-4DI21-L	4ev_i_4gb	REV03	4GB	128MB	NA	1 mm connectors	NA
TE0803-03-4GI11-A	4eg_2i_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4GE11-A	4eg_2_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-4GI21-A	4eg_2i_4gb	REV03	4GB	128MB	NA	NA	NA

TE0803-03-5BE11-A	5eg_2gb	REV03	2GB	128MB	NA	NA	NA
TE0803-03-5DI24-A	5ev_i_4gb	REV03	4GB	512MB	NA	NA	NA
TE0803-03-4BI21-X	4eg_i_4gb	REV03	4GB	128MB	NA	NA	U41 replaced with diode
TE0803-03-3BE21-A	3eg_4gb	REV03	4GB	128MB	NA	NA	NA
TE0803-03-3AE10-A	3cg_0_2gb	REV03	2GB	NA	NA	NA	NA
TE0803-03-3AE10-A	3cg_0_2gb	REV03	2GB	NA	NA	NA	NA

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TEBF0808	Used as reference carrier. **Important**: CPLD Firmware REV07 or newer is recommended

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes
Cooler	It's recommended to use cooler on ZynqMP device
USB Cable	Connect to USB2 or better USB3 Hub for proper power supply over USB
DP Monitor	Optional HW Not all monitors are supported, also Adapter to other Standard can make drouble. Design was testet with **DELL U2412M**
Micro USB to USB A Adapter	Adapter for USB Hub
USB HUB	To connnect Mouse and Keyboard simultaneously
USB Keyboard	need for Ubuntu/Debian GUI
USB Mouse	need for Ubuntu/Debian GUI
DP Cable	--
Sata Disk	Optional HW
SATA Cable	Optional HW
PCIe Card	Optional HW
ETH Cable	Optional HW Ethernet works with DHCP, but can be setup also manually
SD Card	16GB

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

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
Debian SD-Image	*.img	Debian Image for SD-Card
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)
Image	---	Generic Linux kernel binary image file
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems
Device Tree Blob File	*.dtb	Contains a Device Tree Blob

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:

- TE0803 "SK DEMO1" 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.

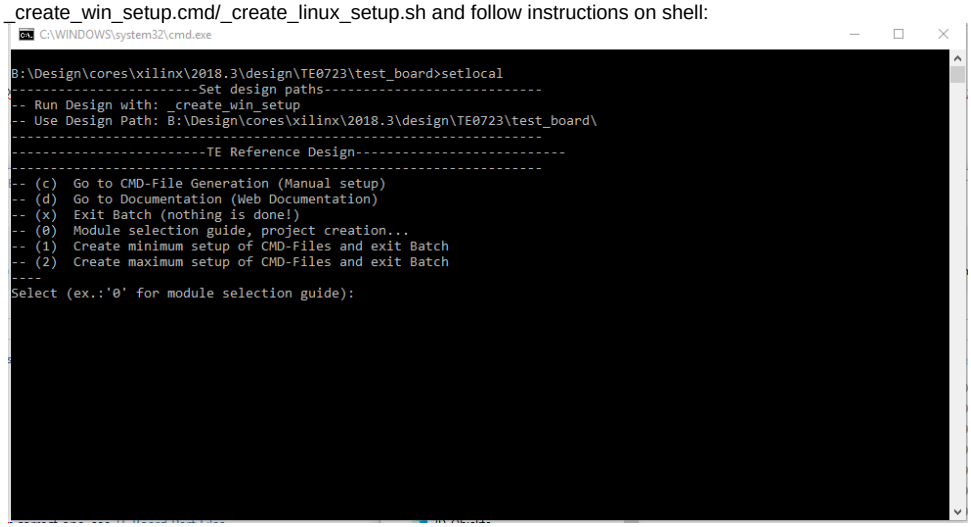
See also:

- 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
        i. Important: Use Board Part Files, which ends with *_tebf0808
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 , Image and system.dtb) 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/
        ii. Execute the script file for Debian/Ubuntu
7. Add Linux files (bl31.elf, uboot.elf , Image and system.dtb) 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
9. Preparing SD card for SD Filesystem and hard disk for HD Filesystem See Programming section

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

Not used in this example.

SD

1. Format the SD Card with SD Card Formatter or other tool
2. Write the Debian image or Ubuntu image file on SD Card with Win32DiskImager
3. It will automatically in BOOT directory two DTB file generated
    a. system_sd.dtb : This file ist used , if the root file system is located on SD card.
    b. system_harddisk.dtb : This file ist used , if the root file system is located on hard disk.
    c. Note: To use one of the DTB files, this file must be renamed to system.dtb
4. Rename the system_sd.dtb file in BOOT directory to system.dtb
5. Copy Petalinux Image (not use image.ub), system.dtb and Boot.bin files 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
6. Set Boot Mode to SD-Boot.
    - Depends on Carrier, see carrier TRM.
7. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TE0803 StarterKit#Programming
2. Connect UART USB (JTAG XMOD)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)
   Note: See TRM of the Carrier, which is used.
4. (Optional) Insert PCIe Card (detection depends on Linux driver. Only some basic drivers are installed)
5. (Optional) Connect Sata Disc
6. (Optional) Connect DisplayPort Monitor (List of usable Monitors: https://www.xilinx.com/support/answers/68671.html)
7. (Optional) Connect Network Cable
8. 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.
4. Debian Desktop
    a. Use connected mouse + keyboard for interaction with GUI
    b. Start the GUI with the command : startx
    c. Web Browser Dillo open console and type dillo or use browser
    d. open console and start video or audio with "mplayer <video or audio file>"
5. Ubuntu Desktop
    a. Use connected mouse + keyboard for interaction with GUI
    b. Start the GUI with the command : startx
    c. Web Browser Mozilla firefox can be used.
    d. Audio or Vider file can also be performed directly in GUI

Hard Disk (optional)

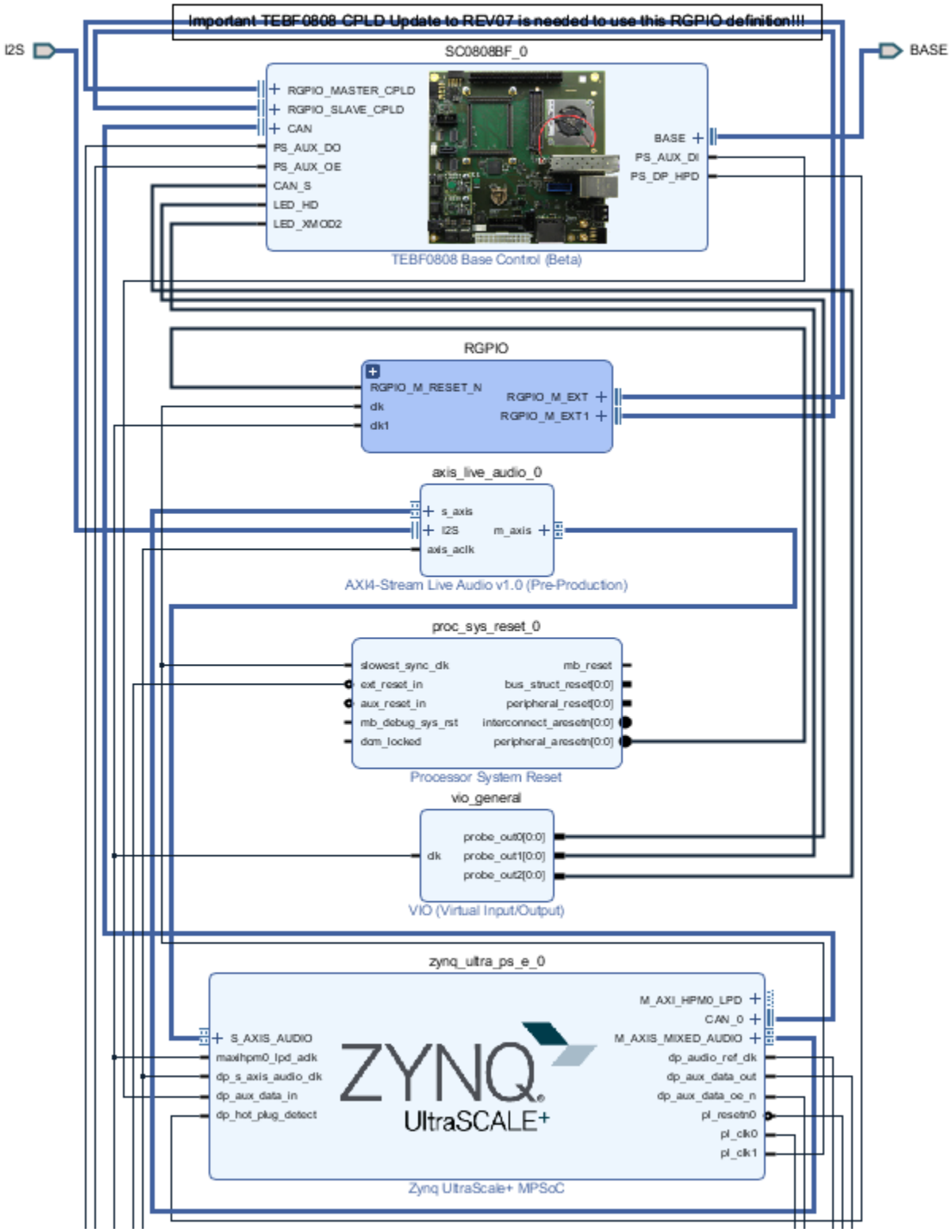
To locate root file system on Hard disk:

1. Plug in SD Card that you have prepared mit root file system
2. Plug in Hard Disk in Sata port on the carrier board
3. Format the hard disk by the following command:
    a. mkfs.ext4 /dev/sda
4. Edit the fstab file in directory /etc/ to mount hard disk by the following commands:
    a. mkdir /media/harddisk
    b. nano /etc/fstab
    c. Add this line to the fstab file and save it : /dev/sda /media/harddisk/   defaults 0 3
    d. Reboot
5. Copy entire root file system in direcroty ROOTFS from SD card to hard disk by the following commands:
    a. cd /media/ROOTFS
    b. cp -r ./ /media/harddisk
6. Edit the fstab file in directory /media/harddisk/etc/ by the following commands and save it:
    a. nano /media/harddisk/etc/fstab
    b. Edit this line to the fstab file : /dev/sda /media/harddisk/   defaults 0 1
    c. Comment this line: #/dev/mmcblk1p2   /media/ROOTFS   defaults 0 1
7. Shutdown the system
8. Format the SD card
9. Rename the Device Tree Blob file system_harddisk.dtb to system.dtb
10. Copy the following files to SD Card:
    a. Image
    b. BOOT.bin
    c. system.dtb
11. Plug in the SD Card and turn on the system

Vivado HW Manager

System Design - Vivado

Block Design


Important TEBF0808 CPLD Update to REV07 is needed to use this RGPIO definition!!!
I2S
BASE
SC0808BF_0
RGPIO_MASTER_CPLD
RGPIO_SLAVE_CPLD
CAN
PS_AUX_DO
PS_AUX_OE
CAN_S
LED_HD
LED_XMOD2
BASE
PS_AUX_DI
PS_DP_HPD
TEBF0808 Base Control (Beta)
RGPIO
RGPIO_M_RESET_N
clk
clk1
RGPIO_M_EXT
RGPIO_M_EXT1
axis_live_audio_0
s_axis
I2S
axis_aclk
m_axis
AXI4-Stream Live Audio v1.0 (Pre-Production)
proc_sys_reset_0
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
vio_general
clk
probe_out0[0:0]
probe_out1[0:0]
probe_out2[0:0]
VIO (Virtual Input/Output)
zynq_ultra_ps_e_0
S_AXIS_AUDIO
maxihpm0_lpd_aclk
dp_s_axis_audio_clk
dp_aux_data_in
dp_hot_plug_detect
ZYNQ
UltraSCALE+
M_AXI_HPM0_LPD
CAN_0
M_AXIS_MIXED_AUDIO
dp_audio_ref_clk
dp_aux_data_out
dp_aux_data_oe_n
pl_resetn0
pl_clk0
pl_clk1
Zynq UltraScale+ MPSoC

Block Design

PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
UART0	MIO
GPIO0	MIO
SWDT0..1	
TTC0..3	
GEM3	MIO
USB0	MIO/GTP
SATA	GTP
DisplayPort	EMIO/GTP
PCIe	MIO/GTP
CAN0	EMIO
PJTAG0	MIO

PS Interfaces

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

```
# system controller ip
#LED_HD SC0 J3:31
#LED_XMOD SC17 J3:48
#CAN RX SC19 J3:52 B26_L11_P
#CAN TX SC18 J3:50 B26_L11_N
#CAN S  SC16 J3:46 B26_L1_N

set_property PACKAGE_PIN G14 [get_ports BASE_sc0]
set_property PACKAGE_PIN D15 [get_ports BASE_sc5]
set_property PACKAGE_PIN H13 [get_ports BASE_sc6]
set_property PACKAGE_PIN H14 [get_ports BASE_sc7]
set_property PACKAGE_PIN A13 [get_ports BASE_sc10_io]
set_property PACKAGE_PIN B13 [get_ports BASE_sc11]
set_property PACKAGE_PIN A14 [get_ports BASE_sc12]
set_property PACKAGE_PIN B14 [get_ports BASE_sc13]
set_property PACKAGE_PIN F13 [get_ports BASE_sc14]
set_property PACKAGE_PIN G13 [get_ports BASE_sc15]
set_property PACKAGE_PIN A15 [get_ports BASE_sc16]
set_property PACKAGE_PIN B15 [get_ports BASE_sc17]
set_property PACKAGE_PIN J14 [get_ports BASE_sc18]
set_property PACKAGE_PIN K14 [get_ports BASE_sc19 ]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc0]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc5]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc6]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc7]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc10_io]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc11]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc12]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc13]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc14]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc15]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc16]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc17]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc18]
set_property IOSTANDARD LVCMOS18 [get_ports BASE_sc19]

# Audio Codec
#LRCLK                  J3:49
#BCLK                    J3:51
#DAC_SDATA        J3:53
#ADC_SDATA        J3:55
set_property PACKAGE_PIN L13 [get_ports I2S_lrclk ]
set_property PACKAGE_PIN L14 [get_ports I2S_bclk ]
set_property PACKAGE_PIN E15 [get_ports I2S_sdin ]
set_property PACKAGE_PIN F15 [get_ports I2S_sdout ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_lrclk ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_bclk ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_sdin ]
set_property IOSTANDARD LVCMOS18 [get_ports I2S_sdout ]
```

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
    - OTG+PCIe Reset over MIO
    - I2C MUX for EEPROM MAC

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_te0803

Hello TE0803 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**

Select Image Packaging Configuration ==> Root filesystem type ==> Select SD Card

Changes:

- CONFIG_SUBSYSTEM_PRIMARY_SD_PSU_SD_1_SELECT=y
- CONFIG_SUBSYSTEM_ETHERNET_PSU_ETHERNET_3_MAC=""
- # CONFIG_SUBSYSTEM_BOOTARGS_AUTO is not set
- CONFIG_SUBSYSTEM_USER_CMDLINE="console=ttyPS0,115200 earlycon clk_ignore_unused earlyprintk root=/dev/mmcblk1p2 rootfstype=ext4 rw rootwait cma=1024M"
- CONFIG_SUBSYSTEM_DEVICETREE_FLAGS=""
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_MEDIA_BOOTIMAGE_SELECT is not set
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_MEDIA_FLASH_SELECT is not set
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_MEDIA_SD_SELECT=y
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_MEDIA_ETHERNET_SELECT is not set
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_MEDIA_MANUAL_SELECT is not set
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_DTB_IMAGE_NAME="system.dtb"
- CONFIG_SUBSYSTEM_ENDIAN_LITTLE=y
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_MEDIA_FLASH_SELECT is not set
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_MEDIA_SD_SELECT=y
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_MEDIA_ETHERNET_SELECT is not set
- # CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_MEDIA_MANUAL_SELECT is not set
- CONFIG_SUBSYSTEM_IMAGES_ADVANCED_AUTOCONFIG_KERNEL_IMAGE_NAME="Image"

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=2
- 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

Device Tree

```
/include/ "system-conf.dtsi"
/ {
  chosen {
            xlnx,eeprom = &eeprom;
                bootargs= "console=ttyPS0,115200 earlycon clk_ignore_unused earlyprintk root=/dev/mmcblk1p2
rootfstype=ext4 rw rootwait cma=1024M";
                /* notes: root=/dev/mmcblk1p2 for SD and root=/dev/sda for hard disk will be changed
automatically  by executing debian/ubuntu script*/
  };
};

/* notes:
serdes: // PHY TYP see: dt-bindings/phy/phy.h
*/
```

```
/* default */

/* SD */

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

};




/* 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 2 100000000>;
    maximum-speed = "super-speed";
};

/* ETH PHY */

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

/* 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>;
    };
};

/* I2C */

&i2c0 {
    i2cswitch@73 { // u
        compatible = "nxp,pca9548";
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0x73>;
        i2c-mux-idle-disconnect;
        i2c@0 { // MCLK TEBF0808 SI5338A, 570FBB000290DG_unassembled
```

```
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0>;
    };
    i2c@1 { // SFP TEBF0808 PCF8574DWR
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <1>;
    };
    i2c@2 { // PCIe
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <2>;
    };
    i2c@3 { // SFP1 TEBF0808
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <3>;
    };
    i2c@4 {// SFP2 TEBF0808
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <4>;
    };
    i2c@5 { // TEBF0808 EEPROM
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <5>;
        eeprom: eeprom@50 {
                compatible = "atmel,24c08";
                reg = <0x50>;
              };
    };
    i2c@6 { // TEBF0808 FMC
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <6>;
    };
    i2c@7 { // TEBF0808 USB HUB
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <7>;
    };
};
i2cswitch@77 { // u
    compatible = "nxp,pca9548";
    #address-cells = <1>;
    #size-cells = <0>;
    reg = <0x77>;
    i2c-mux-idle-disconnect;
    i2c@0 { // TEBF0808 PMOD P1
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <0>;
    };
    i2c@1 { // i2c Audio Codec
        #address-cells = <1>;
        #size-cells = <0>;
        reg = <1>;
                    /*
        adau1761: adau1761@38 {
```

```
                compatible = "adi,adau1761";
                reg = <0x38>;
            };
                        */
        };
        i2c@2 { // TEBF0808 Firefly A
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <2>;
        };
        i2c@3 { // TEBF0808 Firefly B
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <3>;
        };
        i2c@4 { //Module PLL Si5338 or SI5345
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <4>;
        };
        i2c@5 { //TEBF0808 CPLD
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <5>;
        };
        i2c@6 { //TEBF0808 Firefly PCF8574DWR
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <6>;
        };
        i2c@7 { // TEBF0808 PMOD P3
            #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

Rootfs

File System will be generated with Debian script or Ubuntu script (mkdebian_stretch.sh/mkubuntu_BionicBeaver.sh)

Applications

Applications will be generated with Debian script or Ubuntu script (mkdebian_stretch.sh/mkubuntu_BionicBeaver.sh)

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

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.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$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]	**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]	• 2019.2 release

--	all	**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]	--

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.

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]