# TE0841 Test Board

Revision v.8

Exported on 2022-04-13

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0841+Test+Board

# 1 Table of Contents

# 2 Table of Figures

# 3 Table of Tables

# 4 Overview

Example show, how to reconfigure SI5338 with MCS and monitor CLK. Additional MicroBlaze with Linux example.

Refer to http://trenz.org/te0841-info for the current online version of this manual and other available documentation.

## 4.1 Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- MicroBlaze
- I2C
- UART
- Flash
- FMeter
- SI5338 initialisation with MCS

## 4.2 Revision History

| Date | Vivado | Project Built | Authors | Description |
|---|---|---|---|---|
| 2020-0513 | 2019.2 | TE0841-test_board-vivado_2019.2-build_11_20200513071943.zip<br>TE0841-test_board_noprebuilt-vivado_2019.2-build_11_20200513072026.zip | John Hartfiel | • new Assembly variants<br>• add Linux |
| 2018-06-21 | 2017.4 | TE0841-test_board_noprebuilt-vivado_2017.4-build_11_20180621164459.zip<br>TE0841-test_board-vivado_2017.4-build_11_20180621164432.zip | John Hartfiel | • REV02 Board parts<br>• new SI5338 configuration (default REV02)<br>• change xilisf_v5_9 for N25Q512A11G1240E support<br>• Some changes on block design |
| 2018-05-15 | 2017.4 | TE0841-test_board_noprebuilt-vivado_2017.4-build_08_201805151445 | John Hartfiel | • initial release |

| Date | Vivado | Project Built | Authors | Description |
|---|---|---|---|---|
| | | 42.zip TE0841-test_board-vivado_2017.4-build_08_20180515144523.zip | | |

**Table 1: Design Revision History**

## 4.3 Release Notes and Know Issues

| Issues | Description | Workaround | To be fixed version |
|---|---|---|---|
| --- | --- | --- | --- |

**Table 2: Known Issues**

## 4.4 Requirements

### 4.4.1 Software

| Software | Version | Note |
|---|---|---|
| Vitis | 2019.2 | • needed<br>• Vivado is included into Vitis installation |
| PetaLinux | 2019.2 | • needed |
| SI ClockBuilder Pro | --- | • optional |

**Table 3: Software**

### 4.4.2 Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.[1]

Complete List is available on <design name>/board_files/*_board_files.csv

Design supports following modules:

1 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

| Module Model | Board Part Short Name | PCB Revision Support | DDR | QSPI Flash | EMMC | Others | Notes |
|---|---|---|---|---|---|---|---|
| TE0841-01-035-1C | 01_35_1c_1gb | REV01 | 1GB | 32MB | NA | NA | NA |
| TE0841-01-035-1I | 01_35_1i_1gb | REV01 | 1GB | 32MB | NA | NA | NA |
| TE0841-01-035-2I | 01_35_2i_1gb | REV01 | 1GB | 32MB | NA | NA | NA |
| TE0841-01-040-1C | 01_40_1c_1gb | REV01 | 1GB | 32MB | NA | NA | Serial number 512479 up tp 512474 has same 64MB Flash like REV02 |
| TE0841-01-040-1I | 01_40_1i_1gb | REV01 | 1GB | 32MB | NA | NA | NA |
| TE0841-02-035-1C | 02_35_1c_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-035-1I | 02_35_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-035-2I | 02_35_2i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-040-1C | 02_40_1c_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-040-1I | 02_40_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-040-1IL | 02_40_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |

| Module Model | Board Part Short Name | PCB Revision Support | DDR | QSPI Flash | EMMC | Others | Notes |
|---|---|---|---|---|---|---|---|
| TE0841-02-31C21-A | 02_35_1c_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-31I21-A | 02_35_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-32I21-A | 02_35_2i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-41C21-A | 02_40_1c_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-41I21-A | 02_40_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |
| TE0841-02-41I21-L | 02_40_1i_2gb | REV02 | 2GB | 64MB | NA | NA | PLL programmed |

**Table 4: Hardware Modules**
Design supports following carriers:

| Carrier Model | Notes |
|---|---|
| TE0701 | |
| TE0703 | |
| TE0705 | |
| TE0706 | used as reference carrier |
| TEBA0841 | |

**Table 5: 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 |
| heat sink | Heat sink is recommended urgently |

**Table 6: Additional Hardware**

## 4.5 Content

For general structure and of the reference design, see Project Delivery - Xilinx devices[2]

### 4.5.1 Design Sources

| Type | Location | Notes |
|---|---|---|
| Vivado | <design name>/block_design<br><design name>/constraints<br><design name>/ip_lib | Vivado Project will be generated by TE Scripts |
| SDK/HSI | <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 |

**Table 7: Design sources**

### 4.5.2 Additional Sources

| Type | Location | Notes |
|---|---|---|
| SI5338 | <design name>/misc/Si5338 | SI5338 Project with current PLL Configuration |

**Table 8: Additional design sources**

### 4.5.3 Prebuilt

2 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices

| File | File-Extension | Description |
|---|---|---|
| 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 |
| MCS-File | *.mcs | Flash Configuration File with Boot-Image (MicroBlaze or FPGA part only) |
| MMI-File | *.mmi | File with BRAM-Location to generate MCS or BIT-File with *.elf content (MicroBlaze only) |
| 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 |
| SREC-File | *.srec | Converted Software Application for MicroBlaze Processor Systems |

**Table 9: Prebuilt files (only on ZIP with prebult content)**

### 4.5.4 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:

- TE0841 "Test Board" Reference Design[3]

3 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0841/Reference_Design/2019.2/test_board

# 5 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.

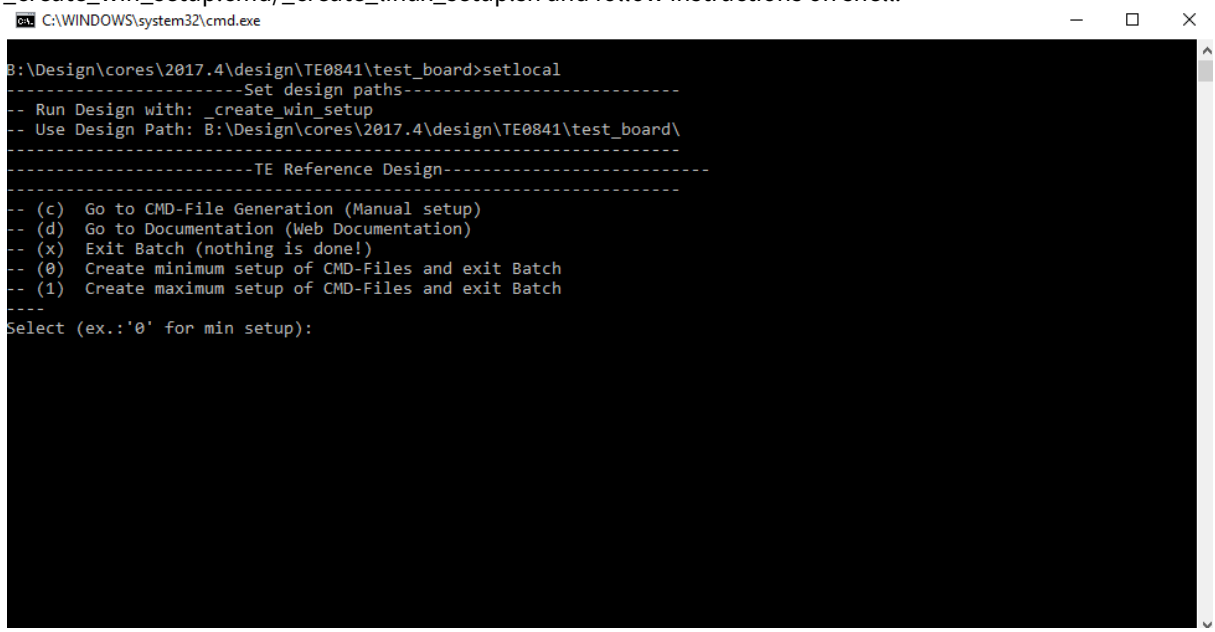
See also:Xilinx Development Tools[4]

- Xilinx Development Tools[5]
- Vivado Projects - TE Reference Design[6]
- Project Delivery.[7]

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[8]

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 TE Board Part Files[9]
5. Create XSA and export to prebuilt folder

4 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
5 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
6 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
7 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices
8 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDeliveryXilinxdevices-Currentlylimitationsoffunctionality
9 https://wiki.trenz-electronic.de/display/PD/TE+Board+Part+Files

   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[10]
      i. Use TE Template from /os/petalinux
         **Important Note**: Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings → Flash Settings, FPGA+Boot+bootenv=0xA00000 (increase automatically generate Boot partition), increase image size to A:, see TE0841 Test Board#Config(see page 22)
7. Add Linux files (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>" of the selected device
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[11]
9. Copy "\prebuilt\software\<short name>\srec_spi_bootloader.elf" into "\firmware\microblaze_0\"
10. (optional) Copy "\\workspace\sdk\scu\Release\scu.elf" into "\firmware\microblaze_mcs_0\"
11. Regenerate Vivado Project or Update Bitfile only with "srec_spi_bootloader.elf" and "scu_te0841.elf"

10 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
11 https://wiki.trenz-electronic.de/display/PD/Vitis

# 6 Launch

## 6.1 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[12]

### 6.1.1 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

### 6.1.2 QSPI

1. Connect JTAG and power on PCB
2. (if not done) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd" or open with "vivado_open_project_guimode.cmd", if generated.
3. Type on Vivado Console: TE::pr_program_flash -swapp u-boot
   Note: Alternative use SDK or setup Flash on Vivado manually
   optional "TE::pr_program_flash -swapp hello_te0841" possible
4. Reboot (if not done automatically)

### 6.1.3 SD

Not used on this Example.

### 6.1.4 JTAG

1. Connect JTAG and power on PCB
2. Open Vivado HW Manager
3. Program FPGA with Bitfile from "prebuilt\hardware\<short dir>"
   a. Note SREC Bootloader try to find application on flash, this will stop, if Flash is empty.

## 6.2 Usage

a. Prepare HW like described on section Programming(see page 14)
b. Connect UART USB (most cases same as JTAG)

12 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftwareProgrammingandDebugging

c. Power on PCB
   Note: FPGA Loads Bitfile from Flash,MCS Firmware configure SI5338 and starts Microblaze, SREC Bootloader from Bitfile Firmware loads U-Boot into DDR (This takes a while), U-boot loads Linux from QSPI Flash into DDR
d. Open Serial Console (e.g. putty)
   i. Speed: 9600
   ii. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

Boot process takes a while, please wait.

```
SREC SPI Bootloader (TE modified): Start initialization

SREC SPI Bootloader (TE modified): SPI driver Init passed

SREC SPI Bootloader (TE modified): Serial Flash Library Init passed

SREC SPI Bootloader (TE modified): Load Image
Loading SREC image from flash @ address: 005e0000
Please wait...
```

### 6.2.1 Linux

Note: Linux boot process is slower on Microblaze.

1. Open Serial Console (e.g. putty)
   a. Speed: 9600
   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

### 6.2.2 Vivado HW Manager:

1. Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
   a. Set radix from VIO signals (fm_si...) to unsigned integer.
      Note: Frequency Counter is inaccurate and displayed unit is Hz
   b. SI will be configured with MCS firmware, default all off on PCB REV01, PCB REV02 SI5338 will be preconfigured.
   c. LED control via VIO
   d. MGT CLK Freq can be changed over BUFG_GT control signals divider
   e. MCS Reset possible via VIO
   f. MIG Reset is possible over VIO
   g. MCS can be disabled over VIO (For PCB REV01 MCS is enabled, fpr PCB REV02 MCS is disabled by default VIO)

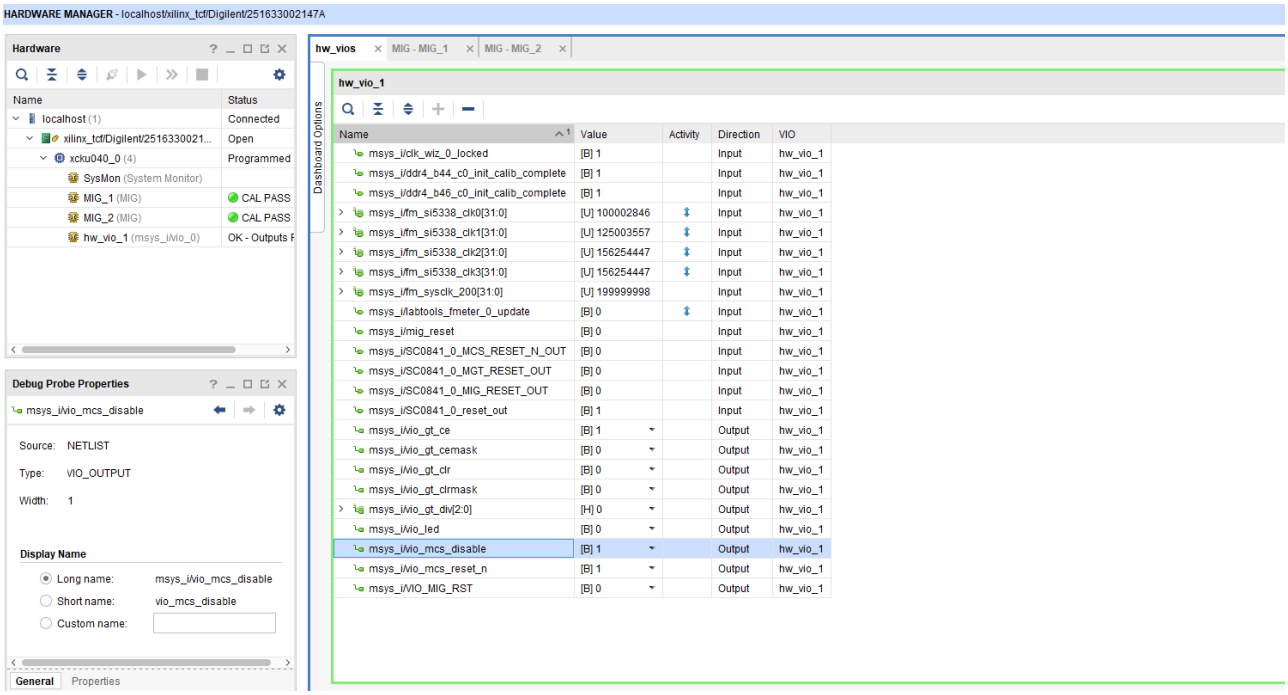



**Figure 1: Vivado Hardware Manager**

# 7 System Design - Vivado

## 7.1 Block Design

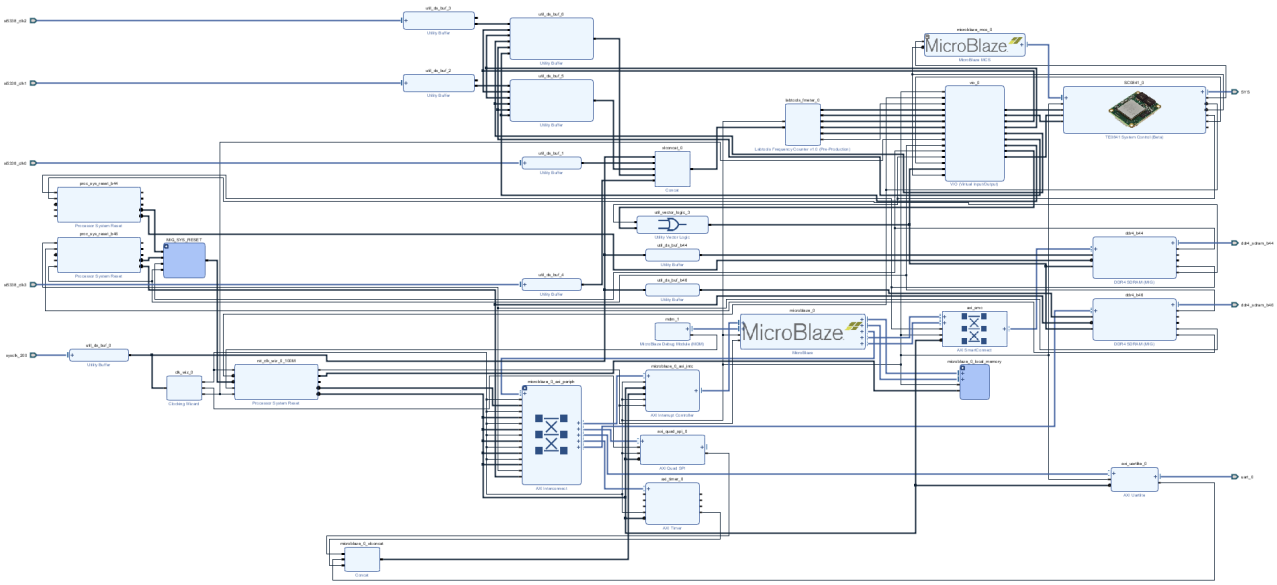



**Figure 2: Block Design PCB REV02**

*Note: REV01 has SI5338 programming default enabled and REV02 default disabled. SI5338 of REV02 is preprogrammed

## 7.2 Constrains

### 7.2.1 Basic module constrains

**_i_bitgen_common.xdc**

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 69 [current_design]
set_property CFGBVS GND [current_design]
set_property CONFIG_VOLTAGE 1.8 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
set_property BITSTREAM.CONFIG.SPI_32BIT_ADDR YES [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]
set_property BITSTREAM.CONFIG.M1PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M2PIN PULLNONE [current_design]
set_property BITSTREAM.CONFIG.M0PIN PULLNONE [current_design]
```

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

### 7.2.2 Design specific constrain

**_i_ddr4.xdc**

```
set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_pins -hier -filter {NAME =~ */u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1}]
create_clock -name ddr4_0_clk -period 4.95 [get_pins */ddr4_0/*/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1]
create_clock -name ddr4_1_clk -period 4.95 [get_pins */ddr4_1/*/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1]
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design
```

**_i_qspi.xdc**

```
# You must provide all the delay numbers
# CCLK delay is 0.1, 6.7 ns min/max for ultra-scale devices; refer Data sheet
# Consider the max delay for worst case analysis
set cclk_delay 6.7
create_generated_clock -name clk_sck -source [get_pins -hierarchical *axi_quad_spi_0/ext_spi_clk] -edges {3 5 7} -edge_shift [list $cclk_delay $cclk_delay $cclk_delay] [get_pins -hierarchical *USRCCLKO]
set_multicycle_path -setup -from clk_sck -to [get_clocks -of_objects [get_pins -hierarchical */ext_spi_clk]] 2
set_multicycle_path -hold -end -from clk_sck -to [get_clocks -of_objects [get_pins -hierarchical */ext_spi_clk]] 1
set_multicycle_path -setup -start -from [get_clocks -of_objects [get_pins -hierarchical */ext_spi_clk]] -to clk_sck 2
set_multicycle_path -hold -from [get_clocks -of_objects [get_pins -hierarchical */ext_spi_clk]] -to clk_sck 1
# Max delay constraints are used to instruct the tool to place IP near to STARTUPE3 primitive.
# If needed adjust the delays appropriately
set_max_delay -datapath_only -from [get_pins -hier {*STARTUP*_inst/DI[*]}] 1.000
set_max_delay -datapath_only -from [get_clocks clk_out2_msys_clk_wiz_0_0] -to [get_pins -hier *STARTUP*_inst/USRCCLKO] 1.000
#set_max_delay -datapath_only -from [get_clocks clk_out2_msys_clk_wiz_0_0] -to [get_pins -hier *STARTUP*_inst/DO[*] {*STARTUP*_inst/DTS[*]}] 1.000
set_max_delay -datapath_only -from [get_clocks clk_out2_msys_clk_wiz_0_0] -to [get_pins -hier *STARTUP*_inst/DO[*]] 1.000
set_max_delay -datapath_only -from [get_clocks clk_out2_msys_clk_wiz_0_0] -to [get_pins -hier *STARTUP*_inst/DTS[*]] 1.000
```

**_i_fm.xdc**

```
set_false_path -from [get_clocks {msys_i/util_ds_buf_5/U0/BUFG_GT_O[0]}] -to [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]]
set_false_path -from [get_clocks {msys_i/util_ds_buf_6/U0/BUFG_GT_O[0]}] -to [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks {msys_i/util_ds_buf_6/U0/BUFG_GT_O[0]}]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks {msys_i/util_ds_buf_5/U0/BUFG_GT_O[0]}]

set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks {msys_i/util_ds_buf_1/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks {msys_i/util_ds_buf_4/U0/IBUF_OUT[0]}]
set_false_path -from [get_clocks {msys_i/util_ds_buf_0/U0/IBUF_OUT[0]}] -to [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]]
set_false_path -from [get_clocks {msys_i/util_ds_buf_1/U0/IBUF_OUT[0]}] -to [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]]
set_false_path -from [get_clocks {msys_i/util_ds_buf_4/U0/IBUF_OUT[0]}] -to [get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst/CLKOUT0]]
```

# 8 Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis[13]

## 8.1 Application

Template location: ./sw_lib/sw_apps/

### 8.1.1 scu

MCS Firmware to configure SI5338 and Reset System.

### 8.1.2 srec_spi_bootloader

TE modified 2019.2 SREC

Bootloader to load app or second bootloader from flash into DDR

Descriptions:

- Modified Files: blconfig.h, bootloader.c
- Changes:
  - Add some console outputs and changed bootloader read address.
  - Add bugfix for 2018.2 qspi flash (some reinitialisation)

### 8.1.3 SREC SPI Bootloader

Modified Xilinx SREC Bootloader. Changes: Correct flash typ and SRec Start address, some additional console outputs, see source code

Changed xilisf_v5_9 to support N25Q512_1V8 for SREC (changes on xilisf.c and xilisf_intelstm.h)

Template location: \sw_lib\sw_apps\srec_spi_bootloader

\sw_lib\sw_services\xilisf_v5_9

### 8.1.4 xilisf_v5_14

TE modified 2019.2 xilisf_v5_14

- Changed default Flash type to 5.

### 8.1.5 hello_te0841

Hello TE0841 is a Xilinx Hello World example as endless loop instead of one console output.

13 https://wiki.trenz-electronic.de/display/PD/Vitis

### 8.1.6 u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate u-boot.srec. Vivado to generate *.mcs

# 9 Software Design - PetaLinux

- PetaLinux KICKstart[14]

Description currently not available.

## 9.1 Config

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

Changes:

- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = 0x6E0000
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = 0x300000
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = 0x20000
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = 0xA00000
    - (Set kernel flash Address to 0xA00000 and Kernel size to 0xA00000)

## 9.2 U-Boot

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

Changes:

- No changes.

Change platform-top.h:

## 9.3 Device Tree

## 9.4 Kernel

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

Changes:

- No changes.

## 9.5 Rootfs

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

Changes:

- # CONFIG_dropbear is not set
- # CONFIG_dropbear-dev is not set
- # CONFIG_dropbear-dbg is not set

14 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

- # CONFIG_packagegroup-core-ssh-dropbear is not set
- # CONFIG_packagegroup-core-ssh-dropbear-dev is not set
- # CONFIG_packagegroup-core-ssh-dropbear-dbg is not set
- # CONFIG_imagefeature-ssh-server-dropbear is not set

## 9.6 Applications

No additional application.

# 10 Additional Software

## 10.1 SI5338

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

General documentation how you work with these project will be available on Si5338[15]

15 https://wiki.trenz-electronic.de/display/PD/Si5338

# 11 Appx. A: Change History and Legal Notices

## 11.1 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 |
|---|---|---|---|
| 2020-05-13 | v.8(see page 6) | @ John Hartfiel[16] | • 19.2 release |
| 2018-08-07 | v.7 | John Hartfiel | • some notes |
| 2018-06-21 | v.5 | John Hartfiel | • Design update<br>• new assembly variants (PCB REV02 |
| 2018-06-21 | v.3 | John Hartfiel | • Release 2017.4 |
| 2018-04-16 | v.1 | @ John Hartfiel[17] | • Initial release |
| --- | All | @ John Hartfiel[18] | --- |

**Table 10: Document change history.**

## 11.2 Legal Notices

## 11.3 Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

## 11.4 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.

16 https://wiki.trenz-electronic.de/display/~j.hartfiel
17 https://wiki.trenz-electronic.de/display/~j.hartfiel
18 https://wiki.trenz-electronic.de/display/~j.hartfiel

## 11.5 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.

## 11.6 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.

## 11.7 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.

## 11.8 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.

## 11.9 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[19]. 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[20] 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)[21].

**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).

19 http://guidance.echa.europa.eu/
20 https://echa.europa.eu/candidate-list-table
21 http://www.echa.europa.eu/

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.

2019-06-07