TE0841 Test Board

Table of contents

- 1 Overview
  - 1.1 Key Features
  - 1.2 Revision History
  - 1.3 Release Notes and Known Issues
  - 1.4 Requirements
    - 1.4.1 Software
    - 1.4.2 Hardware
  - 1.5 Content
    - 1.5.1 Design Sources
    - 1.5.2 Additional Sources
    - 1.5.3 Prebuilt
    - 1.5.4 Download
- 2 Design Flow
- 3 Launch
  - 3.1 Programming
    - 3.1.1 Get prebuilt boot binaries
    - 3.1.2 QSPI-Boot mode
    - 3.1.3 SD-Boot mode
    - 3.1.4 JTAG
  - 3.2 Reset
    - 3.2.1 Linux
    - 3.2.2 Vivado HW Manager
- 4 System Design - Vivado
  - 4.1 Block Design
  - 4.2 Constraints
    - 4.2.1 Basic module constraints
    - 4.2.2 Design specific constraints
- 5 Software Design - Vitis
  - 5.1 Application
    - 5.1.1 scu
    - 5.1.2 spi_bootloader
    - 5.1.3 hello_te0841
    - 5.1.4 u-boot
- 6 Software Design - PetaLinux
  - 6.1 Config
  - 6.2 U-Boot
  - 6.3 Device Tree
  - 6.4 Kernel
  - 6.5 Rootfs
  - 6.6 Applications
- 7 Additional Software
  - 7.1 SI5338
- 8 App. A: Change History and Legal Notices
  - 8.1 Document Change History
  - 8.2 Legal Notices
  - 8.3 Data Privacy
  - 8.4 Document Warranty
  - 8.5 Limitation of Liability
  - 8.6 Copyright Notice
  - 8.7 Technology Licenses
  - 8.8 Environmental Protection
  - 8.9 REACH, RoHS and WEEE
- 9 Table of contents

Overview

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

Refer to http://trm.trenz.org/TE0841 for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2021.2
- PetaLinux
- MicroBlaze
- UART
- Flash
- FMeter
- SI5338 initialisation with MCS

Revision History

Date	Vivado	Project Built	Authors	Description
2022-05-06	2021.2	TE0841-test_board-vivado_2021.2-build_14_20220506142737.zip TE0841-test_board_noprebuilt-vivado_2021.2-build_14_20220506142737.zip	Waldemar Hanemann	• new spi bootloader by Henrik Brix Andersen • fixed SC0841 bugs
2020-05-13	2019.2	TE0841-test_board-vivado_2019.2-build_11_20200513071943.zip TE0841-test_board_noprebuilt-vivado_2019.2-build_11_20200513072026.zip	John Hartfiel	• new Assembly variants • add Linux
2018-06-21	2017.4	TE0841-test_board_noprebuilt-vivado_2017.4-build_11_20180621164459.zip TE0841-test_board-vivado_2017.4-build_11_20180621164432.zip	John Hartfiel	• REV02 Board parts • new SI5338 configuration (default REV02) • change xilisf_v5_9 for N25Q512A11G1240E support • Some changes on block design

2018-05-15	2017.4	TE0841-test_board_noprebuilt-vivado_2017.4-build_08_20180515144542.zip TE0841-test_board-vivado_2017.4-build_08_20180515144523.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	2021.2	needed, Vivado is included into Vitis installation
PetaLinux	2021.2	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 "<project folder>\board_files*_board_files.csv"

Design supports following modules:

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
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

*used as reference

Hardware Modules

Design supports following carriers:

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

*used as reference

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

*used as reference

Additional Hardware

Content

For general structure and usage of the reference design, see Project Delivery - AMD devices

Design Sources

Type	Location	Notes
Vivado	<project folder>\block_design <project folder>\constraints <project folder>\ip_lib <project folder>\board_files	Vivado Project will be generated by TE Scripts
Vitis	<project folder>\sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation
PetaLinux	<project folder>\os\petalinux	PetaLinux template with current configuration

Design sources

Additional Sources

Type	Location	Notes
SI5338	<project folder>\misc\Si5338	SI5338 Project with current PLL Configuration

Additional design sources

Prebuilt

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

Prebuilt files (only on ZIP with prebuilt 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:

- TE0841 "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 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/Vitis GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

Caution! Win OS has a 260 character limit for path lengths which can affect the Vivado tools. To avoid this issue, use Virtual Drive or the shortest possible names and directory locations for the reference design (for example "x:\<project folder>")

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:

_create_win_setup.cmd/_create_linux_setup.sh

```
-----------------------Set design paths---------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
------------------------------------------------------------------
-----------------------TE Reference
Design-------------------------
------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. Create project and follow instructions of the product selection guide, settings file will be configured automatically during this process.
    - 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 Vivado Board Part Flow

4. Create hardware description file (.xsa file) for PetaLinux project and export to prebuilt folder

run on Vivado TCL (Script generates design and export files into "<project folder>\prebuilt\hardware\<short name>")

```
TE::hw_build_design -export_prebuilt
```

Using Vivado GUI is the same, except file export to prebuilt folder.

5. Create and configure your PetaLinux project with exported .xsa-file, see PetaLinux KICKstart
    - use TE Template from "<project folder>\os\petalinux"
    - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.
    - The build images are located in the "<plnx-proj-root>/images/linux" directory

Important Note: Select correct Flash partition offset on petalinux-config: Subsystem Auto HW Settings Flash Settings, FPGA+Boot+bootenv=0xB00000 (increase automatically generate Boot partition), see TE0841 Test Board#Config

6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr
7. Copy PetaLinux build image files(uboot.elf and image.ub) to prebuilt folder
    - copy **u-boot.elf** and **image.ub** "<plnx-proj-root>/images/linux" to prebuilt folder

    > "<project folder>\prebuilt\os\petalinux\<ddr size>" or "<project folder>\prebuilt\os\petalinux\<short name>"

8. Generate Programming Files with Vitis(Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv" and open Vitis)

    run on Vivado TCL (Script generates applications and bootable files, which are defined in "test_board\sw_lib\apps_list.csv")

    ```
    TE::sw_run_vitis -all
    TE::sw_run_vitis (optional; Start Vitis from Vivado GUI or start with TE Scripts on Vivado TCL)
    ```

    > TCL scripts generate also platform project, this must be done manually in case GUI is used. See Vitis

9. (optional) Update spi_bootloader.elf and/or scu_te084.elf
    a. Copy "\prebuilt\software\<short name>\spi_bootloader.elf" into "\firmware\microblaze_0\"
    b. Copy "\\workspace\sdk\scu\Release\scu.elf" into "\firmware\microblaze_mcs_0\"
    c. Regenerate Vivado Project or Update Bitfile only with "spi_bootloader.elf" and "scu_te0841.elf"

Launch

Programming

> Check Module and Carrier TRMs for proper HW configuration before you try any design.
>
> Reference Design is also available with prebuilt files. It's recommended to use TE prebuilt files for first launch.

Xilinx documentation for programming and debugging: Vivado/Vitis/SDSoC-Xilinx Software Programming and Debugging

Get prebuilt boot binaries

1. Run _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
    a. Select assembly version
    b. Validate selection
    c. Select create and open delivery binary folder

    > Note: Folder "<project folder>_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

QSPI-Boot mode

Option for **u-boot.mcs** on QSPI Flash.
(u-boot.mcs contains all files necessary to boot up linux)

1. Connect the **USB cable**(JTAG) and **power supply** 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"
   Enter the following TCL-Command into the TCL-Console inside Vivado to program the QSPI Flash.

run on Vivado TCL (Script programs BOOT.bin on QSPI flash)
`TE::pr_program_flash -swapp u-boot`

To program with Vitis/Vivado GUI, use special FSBL (fsbl_flash) on setup

3. Reboot (if not done automatically)

SD-Boot mode

Not used on this Example.

JTAG

Not used on this example.

Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Select QSPI as Boot Mode

Note: See TRM of the Carrier, which is used.

Starting with Petalinux version 2020.1, the industry standard "Distro-Boot" boot flow for U-Boot was introduced, which significantly expands the possibilities of the boot process and has the primary goal of making booting much more standardised and predictable.
The boot options described above describe the common boot processes for this hardware; other boot options are possible.
For more information see Distro Boot with Boot.scr

4. Power On PCB
   1. FPGA Loads Bitfile from Flash
   2. MCS Firmware configure SI5338 and starts Microblaze
   3. SREC Bootloader from Bitfile Firmware loads U-Boot into DDR
   4. U-boot loads Linux from QSPI Flash into DDR

Linux

1. Open Serial Console (e.g. putty)
    - Speed: 9600
    - select COM Port

    > Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Boot process takes a while, please wait...

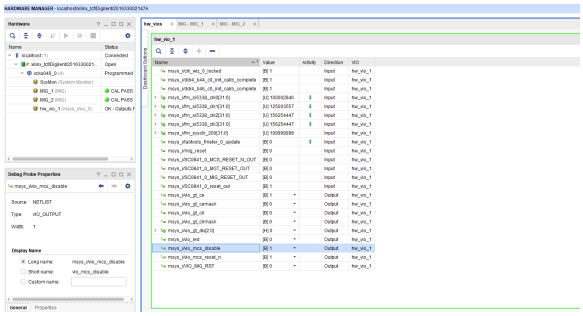
```
SPI ELF Bootloader
Copying ELF image from SPI flash @ 0x006e0000 to RAM
................................................................................
................................................................................
................................................................................
................................................................................
.....................................................................
Transferring execution to program @ 0x80100000


U-Boot 2021.01 (Oct 12 2021 - 09:28:42 +0000)

Model: Xilinx MicroBlaze
DRAM:  512 MiB
WDT:   Not found!
In:    serial
Out:   serial
Err:   serial
Model: Xilinx MicroBlaze
U-BOOT for microblaze-generic

Hit any key to stop autoboot:  2
```

3. You can use Linux shell now.

Vivado HW Manager

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

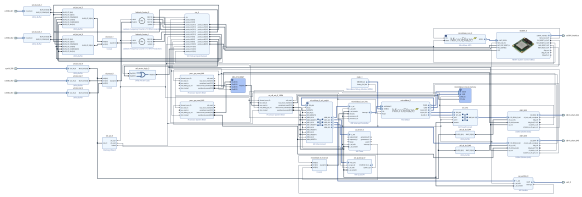
- Control:
    - SI will be configured with MCS firmware, default all off on PCB REV01, PCB REV02 SI5338 will be preconfigured.
    - LED control via VIO
    - MGT CLK Freq can be changed over BUFG_GT control signals divider
    - MCS Reset possible via VIO
    - MIG Reset is possible over VIO
    - MCS can be disabled over VIO (For PCB REV01 MCS is enabled, fpr PCB REV02 MCS is disabled by default VIO)
- Monitoring:
    - Set radix from VIO signals (fm_si...) to unsigned integer.
      Note: Frequency Counter is inaccurate and displayed unit is Hz



Vivado Hardware Manager

System Design - Vivado

Block Design



Block Design PCB

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

Constraints

Basic module constraints

_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]
```

Design specific constraints

_i_io.xdc

```
set_property PACKAGE_PIN AD28 [get_ports sc0841_interface_ddr4_par_44]
set_property PACKAGE_PIN C28 [get_ports sc0841_interface_ddr4_par_46]
set_property PACKAGE_PIN AD20 [get_ports sc0841_interface_en_ddr4pwr]
set_property PACKAGE_PIN AH23 [get_ports sc0841_interface_en_gtpwr]
set_property PACKAGE_PIN AF24 [get_ports sc0841_interface_en_osc]
set_property PACKAGE_PIN AB20 [get_ports sc0841_interface_pll_scl_io]
set_property PACKAGE_PIN P28 [get_ports sc0841_interface_xio_io]
set_property PACKAGE_PIN AE20 [get_ports sc0841_interface_pg_ddr]
set_property PACKAGE_PIN AH22 [get_ports sc0841_interface_pg_gt]
set_property PACKAGE_PIN AB19 [get_ports sc0841_interface_pll_sda_io]

set_property IOSTANDARD SSTL12_DCI [get_ports sc0841_interface_ddr4_par_44]
set_property IOSTANDARD SSTL12_DCI [get_ports sc0841_interface_ddr4_par_46]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_en_ddr4pwr]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_en_gtpwr]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_en_osc]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_pll_scl_io]
set_property IOSTANDARD LVCMOS18 [get_ports sc0841_interface_xio_io]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_pg_ddr]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_pg_gt]
set_property IOSTANDARD LVCMOS33 [get_ports sc0841_interface_pll_sda_io]
```

_i_ddr4.xdc

```
set_property CLOCK_DEDICATED_ROUTE BACKBONE [get_pins -hier -filter {NAME
=~ */u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1}]
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]

create_clock -period 4.950 -name ddr4_0_clk [get_pins */ddr4_b44/*
/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1]
create_clock -period 4.950 -name ddr4_1_clk [get_pins */ddr4_b46/*
/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst/CLKIN1]
```

_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
# 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_clocks clk_out2_msys_clk_wiz_0_0]
-to [get_pins -hier *STARTUP*_inst/DO[*] {*STARTUP*_inst/DTS[*]}] 1.000


create_generated_clock -name clk_sck -source [get_pins -hierarchical
*axi_quad_spi_0/ext_spi_clk] -edges {3 5 7} -edge_shift {6.700 6.700
6.700} [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
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[*]}] 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

```
current_instance msys_i/ddr4_b46/inst
set_property LOC MMCME3_ADV_X0Y2 [get_cells -hier -filter {NAME =~ *
/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst}]
current_instance -quiet
current_instance msys_i/ddr4_b44/inst
set_property LOC MMCME3_ADV_X0Y0 [get_cells -hier -filter {NAME =~ *
/u_ddr4_infrastructure/gen_mmcme*.u_mmcme_adv_inst}]

current_instance -quiet
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks si5338_clk0_clk_p]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks si5338_clk3_clk_p]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/clk_wiz_0
/inst/mmcme3_adv_inst/CLKOUT0]] -to [get_clocks -of_objects [get_pins
{msys_i/util_ds_buf_6/U0/USE_BUFG_GT.GEN_BUFG_GT[0].BUFG_GT_U/O}]]
set_false_path -from [get_clocks si5338_clk3_clk_p] -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/ddr4_b44/inst
/u_ddr4_infrastructure/gen_mmcme3.u_mmcme_adv_inst/CLKOUT0]] -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/ddr4_b46/inst
/u_ddr4_infrastructure/gen_mmcme3.u_mmcme_adv_inst/CLKOUT0]] -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 -of_objects [get_pins
{msys_i/util_ds_buf_5/U0/USE_BUFG_GT.GEN_BUFG_GT[0].BUFG_GT_U/O}]]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/ddr4_b44/inst
/u_ddr4_infrastructure/gen_mmcme3.u_mmcme_adv_inst/CLKOUT0]] -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/ddr4_b46/inst
/u_ddr4_infrastructure/gen_mmcme3.u_mmcme_adv_inst/CLKOUT0]] -to
[get_clocks -of_objects [get_pins msys_i/clk_wiz_0/inst/mmcme3_adv_inst
/CLKOUT0]]
```

Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

Application

Template location: "<project folder>\sw_lib\sw_apps\"

scu

MCS Firmware to configure SI5338 and Reset System.

spi_bootloader

TE modified SPI Bootloader from Henrik Brix Andersen.

Bootloader to load app or second bootloader from flash into DDR.

Here it loads the u-boot.elf from QSPI-Flash to RAM. Hence u-boot.srec becomes redundant.

Descriptions:

- Modified Files: bootloader.c
- Changes:
  - Change the SPI defines in the header
  - Add some reiteration in the frist spi read call

hello_te0841

Hello TE0841 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 u-boot.srec. Vivado to generate *.mcs

Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

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

Changes:

- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = **0x6E0000** (fpga)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = **0x400000** (boot)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = **0x20000** (bootenv)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = **0xB00000** (kernel)
  - (with this kernel flash address is 0xB00000 (fpga+boot+bootenv) and Kernel size 0xB00000)

U-Boot

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

Changes:

- No changes.

Content of **platform-top.h** located in <plnx-proj-root>\project-spec\meta-user\recipes-bsp\u-boot\files:

```
#include <configs/microblaze-generic.h>
#include <configs/platform-auto.h>

#define CONFIG_SYS_BOOTM_LEN 0xF000000
```

Device Tree

Content of **system-user.dtsi** located in <petalinux project directory>\project-spec\meta-user\recipes-bsp\device-tree\files:

```
/include/ "system-conf.dtsi"
/ {
};
```

Kernel

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

Changes:

- No changes.

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
- # 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

Applications

No additional application.

Additional Software

SI5338

File location "<project folder>\misc\Si5338\Si5338-*.slabtimeproj"

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

App. A: Change History and Legal Notices

Document Change History

To get content of older revision go 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.proxy24.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy24.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy24.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.	• new document style • 21.2 release

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

lang.
String,
class
com.
atlassian
.
confluen
ce.
pages.
Page]
due to
overlapp
ing
prototyp
es
between
:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac

e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	
2020-05-13	v.8	John Hartfiel	• 19.2 release
2018-08-07	v.7	John Hartfiel	• some notes
2018-06-21	v.5	John Hartfiel	• Design update • new assembly variants (PCB REV02
2018-06-21	v.3	John Hartfiel	• Release 2017.4
2018-04-16	v.1	John Hartfiel	• Initial release
---	All	**Error renderi ng macro 'page-info'**	---

Ambiguous method overloading for method jdk.proxy24.$Proxy3589#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between

:
[interfac
e com.
atlassian
.
confluen
ce.user.
Conflue
nceUser
, class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb
ject]
[interfac
e com.
atlassian
.user.
User,
class
java.
lang.
String,
class
com.
atlassian
.
confluen
ce.core.
Content
EntityOb

		ject]	

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.proxy244.$Proxy3589#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]