1 Overview

Microblaze Design with linux example.
Refer to http://trenz.org/te0712-info for the current online version of this manual and other available documentation.

1.1 Key Features

- Vitis/Vivado 2020.2
- PetaLinux
- MicroBlaze
- SREC
- I2C
- Flash
- MIG
- FMeter
- SI5338 initialisation with MCS
- ETH

1.2 Revision History

Date	Vivado	Project Built	Authors	Description
2021-06-28	2020.2	TE0712-test_board_noprebuilt-vivado_2020.2-build_5_20210628072407.zip TE0712-test_board-vivado_2020.2-build_5_20210628072421.zip	Manuela Strücker	• 2020.2 update • document style update • update TE Board Part List
2020-03-25	2019.2	TE0712-test_board_noprebuilt-vivado_2019.2-build_8_20200325074937.zip TE0712-test_board-vivado_2019.2-build_8_20200325074915.zip	John Hartfiel	• Script update
2020-01-22	2019.2	TE0712-test_board_noprebuilt-vivado_2019.2-build_3_20200122155446.zip TE0712-test_board-vivado_2019.2-build_3_20200122155355.zip	John Hartfiel	• update for linux user • new script features

Date	Vivado	Project Built	Authors	Description
2020-01-08	2019.2	TE0712-test_board_noprebuilt-vivado_2019.2-build_2_20200108161124.zip TE0712-test_board-vivado_2019.2-build_2_20200108155510.zip	John Hartfiel	• 2019.2 update • Vitis support
2019-04-18	2018.3	TE0712-test_board_noprebuilt-vivado_2018.3-build_05_20190418082456.zip TE0712-test_board-vivado_2018.3-build_05_20190418082240.zip	John Hartfiel	• MCU depends on EOS now
2019-02-22	2018.3	TE0712-test_board_noprebuilt-vivado_2018.3-build_01_20190222073819.zip TE0712-test_board-vivado_2018.3-build_01_20190222073754.zip	John Hartfiel	• TE Script update • linux changes • SCU rework • SI5338 CLKBuilder Pro Project
2018-09-05	2018.2	te0712-test_board-vivado_2018.2-build_03_20180906071356.zip te0712-test_board_noprebuilt-vivado_2018.2-build_03_20180906071434.zip	John Hartfiel	• chance block design: qspi clks, clock wizard(REV01 only) • change timing constrains • add hello_te0712 application • new SREC bootloader version • change linux device tree
2018-05-25	2017.4	te0712-test_board-vivado_2017.4-build_10_20180525155402.zip te0712-test_board_noprebuilt-vivado_2017.4-build_10_20180525155555.zip	John Hartfiel	• solved eth issue for REV01 • changed design + second design for REV01
2018-04-12	2017.4	te0712-test_board-vivado_2017.4-build_07_20180412081225.zip te0712-test_board_noprebuilt-vivado_2017.4-build_07_20180412081253.zip	John Hartfiel	• bugfix constrain file - ETH REFCLK, timing

Date	Vivado	Project Built	Authors	Description
2018-03-28	2017.4	te0712-test_board-vivado_2017.4-build_07_20180328145151.zip te0712-test_board_noprebuilt-vivado_2017.4-build_07_20180328145135.zip	John Hartfiel	• new assembly variant
2018-01-08	2017.4	te0712-test_board-vivado_2017.4-build_02_20180108155712.zip te0712-test_board_noprebuilt-vivado_2017.4-build_02_20180108155735.zip	John Hartfiel	• no design changes • small constrain changes
2017-12-15	2017.2	te0712-test_board-vivado_2017.2-build_07_20171215172447.zip te0712-test_board_noprebuilt-vivado_2017.2-build_07_20171215172514.zip	John Hartfiel	• add SI5338 initialisation with MCS • add Ethernet IP
2017-11-07	2017.2	te0712-test_board-vivado_2017.2-build_05_20171107172917.zip te0712-test_board_noprebuilt-vivado_2017.2-build_05_20171107172939.zip	John Hartfiel	• add wiki link in Boart Part Files • set correct short link for te0712-02-200-2c
2017-10-05	2017.2	te0712-test_board-vivado_2017.2-build_03_20171005082148.zip te0712-test_board_noprebuilt-vivado_2017.2-build_03_20171005082225.zip	John Hartfiel	• initial release

Table 1: Design Revision History

1.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
For PCB REV01 only: prebuilt does not boot	There is a Pullup missing on REV01 I2C SCL, so SI5338	Remove MCS	**solved** with 20180528 update

Issues	Description	Workaround	To be fixed version
	configuration over MCS fails		
For PCB REV01 only: CLK1B is not available on	additional clk is not connected on PCB	use other internal generated CLK, maybe more effort is needed to get ETH running	**solved** with 20180528 update
SREC SPI BootLoader default Offset	Default load offset is set to 0x400000	Change manually on SDK to 0x5E0000	**solved** with 20180412 update

Table 2: Known Issues

1.4 Requirements

1.4.1 Software

Software	Version	Note
Vitis	2020.2	needed, Vivado is included into Vitis installation
PetaLinux	2020.2	needed
SI ClockBuilder Pro	---	optional

Table 3: Software

1.4.2 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
TE0712-01-100-1I*	01_100_1i_1gb	REV01	1GB	32MB	NA	NA	NA
TE0712-01-100-2C	01_100_2c_1gb	REV01	1GB	32MB	NA	NA	NA
TE0712-01-100-2C3	01_100_2c_1gb	REV01	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-01-200-1I	01_200_1i_1gb	REV01	1GB	32MB	NA	NA	NA
TE0712-01-200-2C	01_200_2c_1gb	REV01	1GB	32MB	NA	NA	NA
TE0712-01-200-2C3	01_200_2c_1gb	REV01	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-100-1I	100_1i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-100-2C	100_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-100-2C3	100_2c_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-100-2CA	100_2ca_1gb	REV02	1GB	32MB	NA	NA	Micron QSPI Flash
TE0712-02-200-1I	200_1i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-200-1I3	200_1i_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-200-2C	200_2c_1gb	REV02	1GB	32MB	NA	NA	NA

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0712-02-200-2C3	200_2c_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-200-2I	200_2i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-35-2I	35_2i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-42I36-A	35_2i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-71I06-M	100_1i_1gb	REV02	0GB	32MB	NA	NA	Without DDR
TE0712-02-71I36-A	100_1i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-72C03-M	100_2ca_1gb	REV02	0GB	32MB	NA	NA	Without DDR
TE0712-02-72C06-M	100_2c_1gb	REV02	0GB	32MB	NA	NA	Without DDR
TE0712-02-72C36-A	100_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-72C36-C	100_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-72C36-L	100_2c_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-81I36-A	200_1i_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-81I36-AC	200_1i_1gb	REV02	1GB	32MB	NA	NA	NA

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0712-02-81I36-L	200_1i_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-81I36-X	200_1i_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-82C11-P	200_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-82C36-A	200_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-82C36-AW	200_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-82C36-L	200_2c_1gb	REV02	1GB	32MB	NA	2.5 mm Samtec connectors	NA
TE0712-02-82C36-P	200_2c_1gb	REV02	1GB	32MB	NA	NA	NA
TE0712-02-82I36-A	200_2i_1gb	REV02	1GB	32MB	NA	NA	NA

Table 4: Hardware Modules

*used as reference

Design supports following carriers:

Carrier Model	Notes
TE0701	
TE0703*	
TE0705	
TE0706	

Carrier Model	Notes
TEBA0841	

Table 5: Hardware Carrier
*used as reference

Additional HW Requirements:

Additional Hardware	Notes
USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct type
XMOD Programmer	Carrier Board dependent, only if carrier has no own FTDI

Table 6: Additional Hardware

1.5 Content

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

1.5.1 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

Table 7: Design sources

1.5.2 Additional Sources

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

Table 8: Additional design sources

1.5.3 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 -Description-File	*.xsa	Exported Vivado hardware description file 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

File	File-Extension	Description
SREC-File	*.srec	Converted Software Application for MicroBlaze Processor Systems

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

1.5.4 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:

- TE0712 "Test Board" Reference Design

2 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:Xilinx Development Tools

- Xilinx Development Tools
- 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 "\prebuilt\hardware\")

```
\prebuilt\hardware\")">
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=0x900000 (increase automatically generate Boot partition), increase image size to A:, see Config

6. Copy PetaLinux build image files to prebuilt folder
   - copy **u-boot.elf** and **image.ub** from "<plnx-proj-root>/images/linux" to prebuilt folder

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

7. Generate Programming Files with 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

8. (Optional) BlockRam Firmware Update
   a. Copy "<project folder>\prebuilt\software\<short name>\srec_spi_bootloader.elf" into "<project folder>\firmware\microblaze_0\"
   b. Copy "<project folder>\workspace\sdk\scu_te0712\Release\scu_te0712.elf" into "\firmware\microblaze_mcs_0\"
   c. Regenerate Vivado Project or Update Bitfile only with "srec_spi_bootloader.elf" and "scu_te0712.elf"

```
TE::hw_build_design -export_prebuilt
TE::sw_run_vitis -all
```

3 Launch

3.1 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

3.1.1 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

3.1.2 QSPI-Boot mode

1. Connect **JTAG** and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"

   run on Vivado TCL (Script programs BOOT.bin on QSPI flash)

   ```
   TE::pr_program_flash -swapp u-boot
   ```

3. Reboot (if not done automatically)

3.1.3 SD-Boot mode

Not used on this Example.

3.1.4 JTAG

Not used on this Example.

3.2 Usage

1. Prepare HW like described on section Programming
2. Connect UART USB (most cases same as JTAG)
3. Power on PCB
   boot process

   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 (This takes a while),
   4. U-boot loads Linux from QSPI Flash into DDR

3.2.1 Linux

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

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

2. 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...
```

3. Linux Console:

```
petalinux login: root
Password: root
```

> Note: Wait until Linux boot finished.
> Linux boot process is slower on Microblaze.

4. You can use Linux shell now.

```
udhcpc              (ETH0 check)
```

3.2.2 Vivado HW Manager

- Open Vivado HW-Manager and add VIO signal to dashboard (*.ltx located on prebuilt folder).
  a. Set radix from VIO signals (MGT REF, MIG_OUT, CLK1B, CLK0) to unsigned integer.
     Note: Frequency Counter is inaccurate and displayed unit is Hz
- Monitoring:
  - MGT REF~125MHz, MIG_50MHZ~50MHz., CLK1B ~50MHz, CLK0~100MHz
  - System reset from MCS and GIO outputs

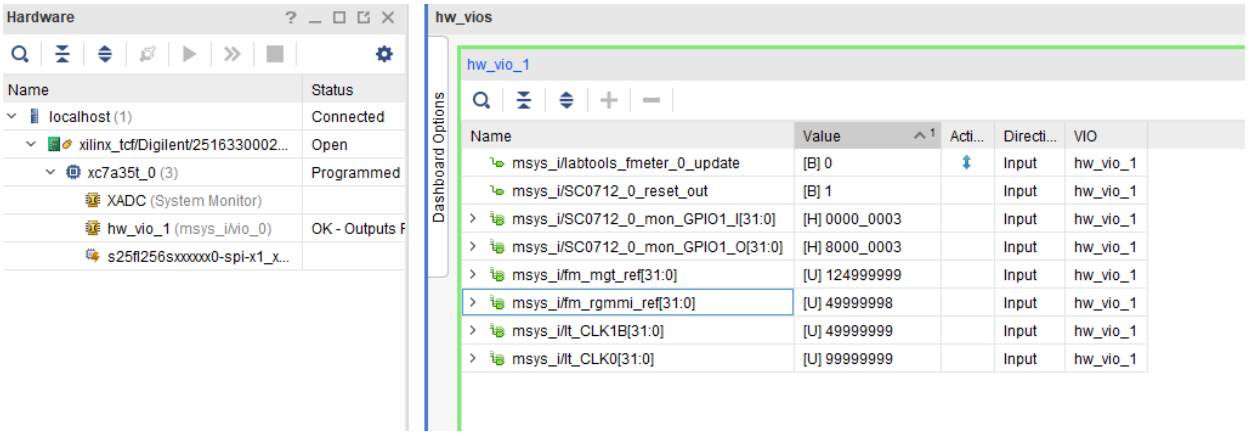

Figure 1: Vivado Hardware Manager

4 System Design - Vivado

4.1 Block Design

4.1.1 REV02

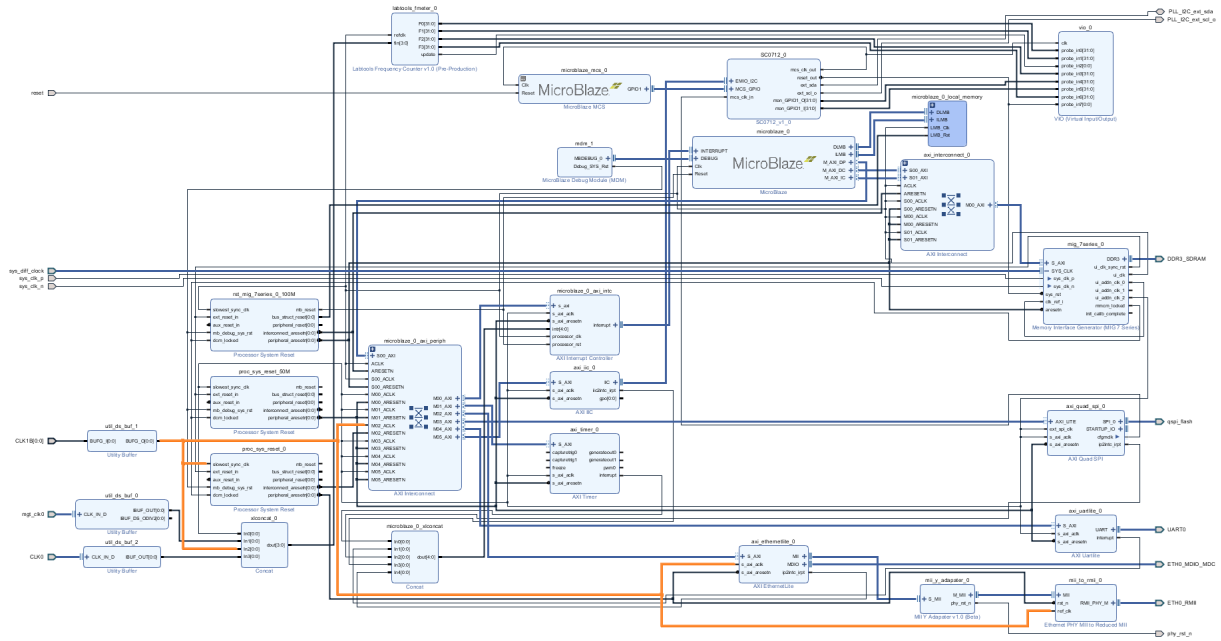

Figure 2: Block Design PCB REV02

4.1.2 REV01

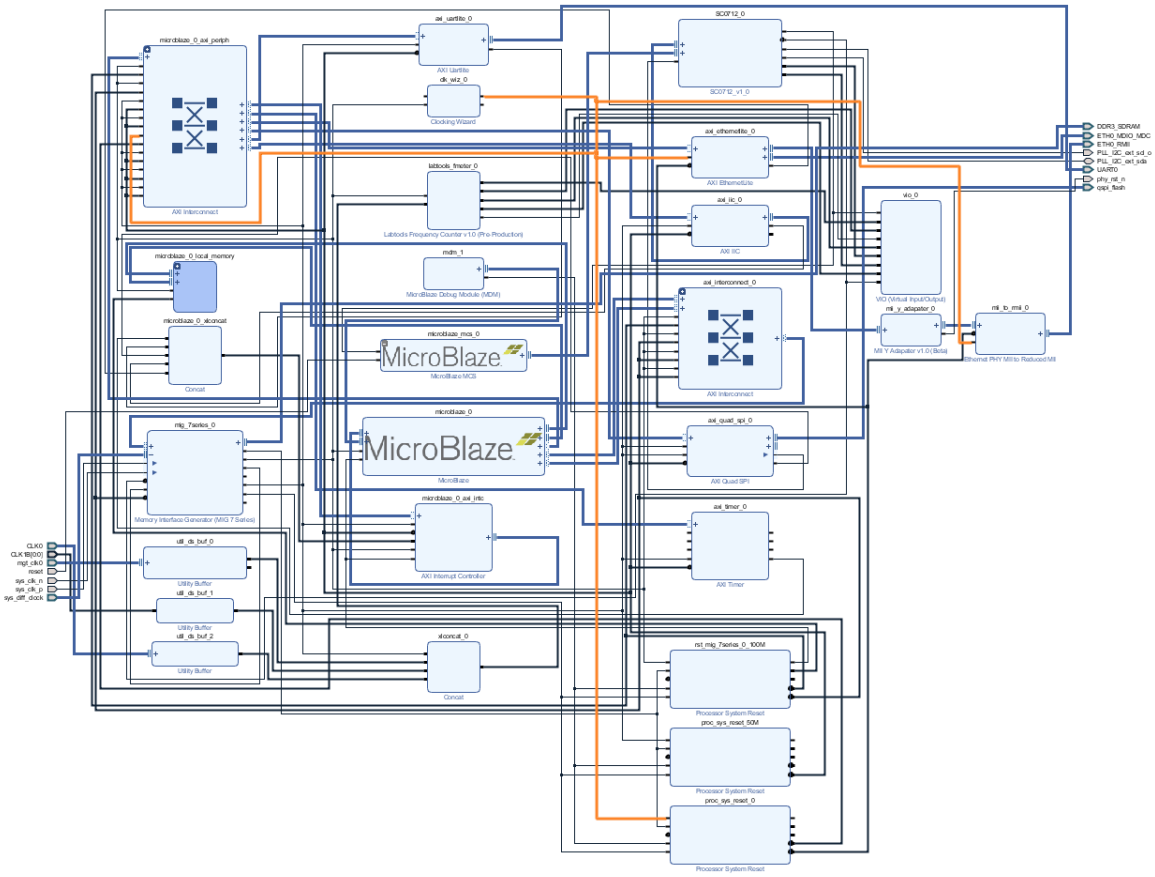

Figure 3: Block Design PCB REV01 (Same as REV02 but 50 MHz ETH REV CLK is generated from MIG output with 180° Phase shift.)

4.2 Constrains

4.2.1 Basic module constrains

_i_bitgen_common.xdc

```
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 66 [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]
set_property 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]
```

_i_bitgen.xdc

```
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLDOWN [current_design]
```

4.2.2 Design specific constrain

_i_reset.xdc

```
set_property PULLDOWN true [get_ports reset]
```

_i_io.xdc

```
#I2C
#set_property PACKAGE_PIN W21 [get_ports PLL_I2C_scl_io]
#set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_scl_io]
#set_property PACKAGE_PIN T20 [get_ports PLL_I2C_sda_io]
#set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_sda_io]
set_property PACKAGE_PIN W21 [get_ports PLL_I2C_ext_scl_o]
set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_ext_scl_o]
set_property PACKAGE_PIN T20 [get_ports PLL_I2C_ext_sda]
set_property IOSTANDARD LVCMOS33 [get_ports PLL_I2C_ext_sda]

#Reset
set_property PACKAGE_PIN T3 [get_ports reset]
set_property IOSTANDARD LVCMOS15 [get_ports reset]
#CLKS
set_property PACKAGE_PIN R4 [get_ports {CLK1B[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {CLK1B[0]}]
set_property PACKAGE_PIN K4 [get_ports {CLK0_clk_p[0]}]
set_property IOSTANDARD DIFF_SSTL15 [get_ports {CLK0_clk_p[0]}]

#ETH PHY
set_property PACKAGE_PIN N17 [get_ports phy_rst_n]
set_property IOSTANDARD LVCMOS33 [get_ports phy_rst_n]
```

_i_timing.xdc

```
create_clock -period 8.000 -name mgt_clk0_clk_p -waveform {0.000 4.000} [get_ports
mgt_clk0_clk_p]


create_clock -period 10.000 -name {CLK0_clk_p[0]} -waveform {0.000 5.000}
[get_ports {CLK0_clk_p[0]}]
create_clock -period 20.000 -name {CLK1B[0]} -waveform {0.000 10.000} [get_ports
{CLK1B[0]}]
```

```
create_clock -period 15.152 -name CFGMCLK -waveform {0.000 7.576} [get_pins
-hierarchical -filter {NAME =~*NO_DUAL_QUAD_MODE.QSPI_NORMAL/*STARTUP_7SERIES_GEN.
STARTUP2_7SERIES_inst/CFGMCLK}]


set_false_path -from [get_clocks {CLK0_clk_p[0]}] -to [get_clocks clk_pll_i]
set_false_path -from [get_clocks mgt_clk0_clk_p] -to [get_clocks clk_pll_i]
set_false_path -from [get_pins {msys_i/SC0712_0/U0/rst_delay_i_reg[3]/C}] -to
[get_pins -hierarchical -filter {NAME =~*u_msys_mig_7series_0_0_mig/
u_ddr3_infrastructure/rstdiv0*/PRE}]
set_false_path -from [get_clocks -of_objects [get_pins msys_i/mig_7series_0/
u_msys_mig_7series_0_0_mig/u_ddr3_infrastructure/gen_ui_extra_clocks.mmcm_i/
CLKFBOUT]] -to [get_clocks mgt_clk0_clk_p]
set_false_path -from [get_clocks clk_pll_i] -to [get_clocks {msys_i/util_ds_buf_0/
U0/IBUF_OUT[0]}]
set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/F_reg[*]/C}] -to
[get_pins {msys_i/vio_0/inst/PROBE_IN_INST/probe_in_reg_reg[*]/D}]
set_false_path -from [get_pins msys_i/labtools_fmeter_0/U0/COUNTER_REFCLK_inst/
bl.DSP48E_2/CLK] -to [get_pins {msys_i/vio_0/inst/PROBE_IN_INST/
probe_in_reg_reg[*]/D}]
set_false_path -from [get_pins {msys_i/labtools_fmeter_0/U0/
FMETER_gen[*].COUNTER_F_inst/bl.DSP48E_2/CLK}] -to [get_pins {msys_i/
labtools_fmeter_0/U0/F_reg[*]/D}]
```

5 Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis

5.1 Application

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

5.1.1 scu_te0712

MCS Firmware to configure SI5338 and Reset System.

5.1.2 srec_spi_bootloader

TE modified 2020.2 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)

5.1.3 hello_te0712

Hello TE0712 is a Xilinx Hello World example as endless loop instead of one console output.

5.1.4 u-boot

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

6 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

6.1 Config

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

Changes:

- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART0_SIZE = 0x5E0000 (fpga)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART1_SIZE = 0x300000 (boot)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART2_SIZE = 0x20000 (bootenv)
- SUBSYSTEM_FLASH_AXI_QUAD_SPI_0_BANKLESS_PART3_SIZE = 0xB00000 (kernel)
    - (Set kernel flash Address to 0x900000 (fpga+boot+bootenv) and Kernel size to 0xB00000)

6.2 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_PHY_ATHEROS is not set
- # CONFIG_PHY_BROADCOM is not set
- # CONFIG_PHY_DAVICOM is not set
- # CONFIG_PHY_LXT is not set
- # CONFIG_PHY_MICREL_KSZ90X1 is not set
- # CONFIG_PHY_MICREL is not set
- # CONFIG_PHY_NATSEMI is not set
- # CONFIG_PHY_REALTEK is not set
- CONFIG_RGMII=y

Change platform-top.h:

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

#define CONFIG_SYS_BOOTM_LEN 0xF000000
```

```
/* Extra U-Boot Env settings */
#define CONFIG_EXTRA_ENV_SETTINGS \
    SERIAL_MULTI \
    CONSOLE_ARG \
    TTYUL0 \
    "nc=setenv stdout nc;setenv stdin nc;\0" \
    "ethaddr=00:0a:35:00:22:01\0" \
    "autoload=no\0" \
    "sdbootdev=0\0" \
    "clobstart=0x80000000\0" \
    "netstart=0x80000000\0" \
    "dtbnetstart=0x81e00000\0" \
    "netstartaddr=0x81000000\0" \
  "bootcmd=sf probe 0 0 0 && sf read ${imageub_addr} ${imageub_flash_addr} $
{imageub_flash_size} && echo QSPI: Trying to boot image.ub at ${imageub_addr} &&
bootm ${imageub_addr}; \0" \
  "imageub_addr=0x81000000\0" \
  "imageub_flash_addr=0x0900000\0" \
  "imageub_flash_size=0x00b00000\0" \
  "loadaddr=0x81000000\0" \
    "initrd_high=0x0\0" \
    "bootsize=0x300000\0" \
    "bootstart=0x5e0000\0" \
    "boot_img=u-boot-s.bin\0" \
    "load_boot=tftpboot ${clobstart} ${boot_img}\0" \
    "update_boot=setenv img boot; setenv psize ${bootsize}; setenv installcmd
\"install_boot\"; run load_boot test_img; setenv img; setenv psize; setenv
installcmd\0" \
    "install_boot=sf probe 0 && sf erase ${bootstart} ${bootsize} && " \
        "sf write ${clobstart} ${bootstart} ${filesize}\0" \
    "bootenvsize=0x20000\0" \
    "bootenvstart=0x8e0000\0" \
    "eraseenv=sf probe 0 && sf erase ${bootenvstart} ${bootenvsize}\0" \
    "kernelsize=0xb00000\0" \
    "kernelstart=0x900000\0" \
    "kernel_img=image.ub\0" \
    "load_kernel=tftpboot ${clobstart} ${kernel_img}\0" \
    "update_kernel=setenv img kernel; setenv psize ${kernelsize}; setenv
installcmd \"install_kernel\"; run load_kernel test_crc; setenv img; setenv psize;
setenv installcmd\0" \
    "install_kernel=sf probe 0 && sf erase ${kernelstart} ${kernelsize} && " \
        "sf write ${clobstart} ${kernelstart} ${filesize}\0" \
    "cp_kernel2ram=sf probe 0 && sf read ${netstart} ${kernelstart} ${kernelsize}
\0" \
    "fpgasize=0x5e0000\0" \
    "fpgastart=0x0\0" \
    "fpga_img=system.bit.bin\0" \
    "load_fpga=tftpboot ${clobstart} ${fpga_img}\0" \
    "update_fpga=setenv img fpga; setenv psize ${fpgasize}; setenv installcmd
\"install_fpga\"; run load_fpga test_img; setenv img; setenv psize; setenv
installcmd\0" \
    "install_fpga=sf probe 0 && sf erase ${fpgastart} ${fpgasize} && " \
        "sf write ${clobstart} ${fpgastart} ${filesize}\0" \
    "fault=echo ${img} image size is greater than allocated place - partition $
{img} is NOT UPDATED\0" \
```

```
    "test_crc=if imi ${clobstart}; then run test_img; else echo ${img} Bad CRC - $
{img} is NOT UPDATED; fi\0" \
    "test_img=setenv var \"if test ${filesize} -gt ${psize}\\; then run fault\\;
else run ${installcmd}\\; fi\"; run var; setenv var\0" \
    "netboot=tftpboot ${netstartaddr} ${kernel_img} && bootm\0" \
    "default_bootcmd=bootcmd\0" \
""
```

6.3 Device Tree

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

/* QSPI PHY */

&axi_quad_spi_0 {
    #address-cells = <1>;
    #size-cells = <0>;
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        spi-tx-bus-width=<1>;
        spi-rx-bus-width=<4>;
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
        spi-max-frequency = <25000000>;
    };
};


/* ETH PHY */
&axi_ethernetlite_0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            device_type = "ethernet-phy";
            reg = <1>;
        };
    };
};
```

6.4 Kernel

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

Changes:

- No changes.

6.5 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
- # CONFIG_package-group-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

"Dropbear" is part of the "petalinux-image-minimal" configuration, so changes in the petalinux rootfs will not be applied. To remove "dropbear" anyway, enter the following line in petalinuxbsp.conf:

```
PACKAGE_EXCLUDE += " dropbear dropbear-openssh-sftp-server dropbear-dev dropbear-dbg dropbear-openssh-sftp-server packagegroup-core-ssh-dropbear packagegroup-core-ssh-dropbear-dbg packagegroup-core-ssh-dropbear-dev"
```

6.6 Applications

No additional application.

7 Additional Software

7.1 SI5338

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

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

8 Appx. A: Change History and Legal Notices

8.1 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
2021-06-28	v.38	@ Manuela Strücker	• 2020.2 update • document style update • update TE Board Part List
2021-06-28	v.37	John Hartfiel	• typo correction
2020-03-25	c.35	John Hartfiel	• update scripts
2020-01-21	v.34	John Hartfiel	• update scripts, new features and linux support
2020-01-08	v.33	John Hartfiel	• 2019.2 release
2019-04-18	v.32	John Hartfiel	• small design changes
2019-02-22	v.31	John Hartfiel	• 2018.3 release (include design reworks)
2018-09-06	v.30	John Hartfiel	• 2018.2 release
2018-05-25	v.28	John Hartfiel	• Design update
2018-05-08	v.27	John Hartfiel	• Know Issues • Documentation
2018-04-12	v.23	John Hartfiel	• Design Update
2018-03-28	v.22	John Hartfiel	• Know Issue for PCB REV01 only • Fix typo • New assembly variant
2018-02-13	v.19	John Hartfiel	• Release 2017.4
2018-01-08	v.16	John Hartfiel	• Add SCU source path
2017-12-15	v.15	John Hartfiel	• Update Design and Description
2017-11-07	v.11	John Hartfiel	• Update Design Files

Date	Document Revision	Authors	Description
2017-10-06	v.10	John Hartfiel	• small Document Update
2017-10-05	v.8	John Hartfiel	• Release 2017.2
2017-09-11	v.1	@ John Hartfiel	• Initial release
---	All	@ John Hartfiel , Manuela Strücker	---

Table 10: Document change history.

8.2 Legal Notices

8.3 Data Privacy

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

8.4 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.

8.5 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.

8.6 Copyright Notice

8.7 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.

8.8 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.

8.9 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. 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.

2019-06-07