TE0720 Test Board

Revision v.49

Exported on 2022-01-17

Online version of this document:

https://wiki.trenz-electronic.de/display/PD/TE0720+Test+Board

1 Table of Contents

2 Table of Figures

3 Table of Tables

4 Overview

Zynq PS Design with Linux Example and PHY status LED on Vivado HW-Manager.
Refer to http://trenz.org/te0720-info for the current online version of this manual and other available documentation.

4.1 Key Features

- Vitis/Vivado 2020.2
- PetaLinux
- SD
- ETH (use EEPROM MAC)
- USB
- I2C
- RTC
- VIO PHY LED
- FSBL for EEPROM MAC and CPLD access / petalinux patch
- Special FSBL for QSPI Programming

4.2 Revision History

Date	Vivado	Project Built	Authors	Description
2021-12-15	2020.2	TE0720-test_board-vivado_2020.2-build_9_20211215123235.zip TE0720-test_board-vivado_2020.2-build_9_20211215123235_production.zip	Manuela Strücker	• new Assembly variants
2021-11-29	2020.2	TE0720-test_board-vivado_2020.2-build_9_20211129062154.zip TE0720-test_board_noprebuilt-vivado_2020.2-build_9_20211129062716.zip	Manuela Strücker	• new Assembly variants
2021-07-19	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_6_202107191318	Manuela Strücker	• boot.scr file updated for 256 MB QSPI flash size variants

Date	Vivado	Project Built	Authors	Description
		00.zip TE0720-test_board-vivado_2020.2-build_6_20210719131744.zip		
2021-04-30	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_5_20210430085624.zip TE0720-test_board-vivado_2020.2-build_5_20210430085609.zip	Manuela Strücker	• update board files • update boot.scr file
2021-04-01	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_4_20210401140444.zip TE0720-test_board-vivado_2020.2-build_4_20210401140432.zip	John Hartfiel	• bugfix missing binaries+ boot.scr file(supports now QSPI and SD boot with image.ub on SD)
2021-02-17	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_2_20210217064925.zip TE0720-test_board-vivado_2020.2-build_2_20210217064913.zip	John Hartfiel	• 2020.2 update • add boot.scr file • petalinux fsbl patch (beta-version)
2020-03-25	2019.2	TE0720-test_board_noprebuilt-vivado_2019.2-build_8_20200325075220.zip TE0720-test_board-vivado_2019.2-build_8_20200325075301.zip	John Hartfiel	• script update

Date	Vivado	Project Built	Authors	Description
2020-01-22	2019.2	TE0720-test_board-vivado_2019.2-build_3_20200122154933.zip TE0720-test_board_noprebuilt-vivado_2019.2-build_3_20200122154951.zip	John Hartfiel	• script update for linux user
2020-01-14	2019.2	TE0720-test_board-vivado_2019.2-build_3_20200114090828.zip TE0720-test_board_noprebuilt-vivado_2019.2-build_3_20200114090837.zip	John Hartfiel	• Vitis script updates (include linux domain and prebuilt linux files for vitis) • prebuilt binary export on selection guide
2019-12-18	2019.2	TE0720-test_board-vivado_2019.2-build_1_20191218151902.zip TE0720-test_board_noprebuilt-vivado_2019.2-build_1_20191218152732.zip	John Hartfiel	• 2019.2 update • Vitis support
2019-03-04	2018.3	TE0720-test_board-vivado_2018.3-build_01_20190304100745.zip TE0720-test_board_noprebuilt-vivado_2018.3-build_01_20190304100755.zip	John Hartfiel	• update for -1CR version only (256MB DDR3)
2019-02-21	2018.3	TE0720-test_board-vivado_2018.3-build_01_20190221125123.zip TE0720-	John Hartfiel	• TE Script update • rework of the FSBLs • some additional Linux features

Date	Vivado	Project Built	Authors	Description
		test_board_noprebuilt-vivado_2018.3-build_01_20190221125133.zip		
2018-08-23	2018.2	te0720-test_board-vivado_2018.2-build_03_20180823185142.zip te0720-test_board_noprebuilt-vivado_2018.2-build_03_20180823185158.zip	John Hartfiel	• DDR setup bugfix for l1if only
2018-08-13	2018.2	te0720-test_board-vivado_2018.2-build_02_20180810162024.zip te0720-test_board_noprebuilt-vivado_2018.2-build_02_20180810162040.zip	John Hartfiel	• 2018.2 update • Board Part Files rework
2018-04-26	2017.4	te0720-test_board-vivado_2017.4-build_07_20180426144351.zip te0720-test_board_noprebuilt-vivado_2017.4-build_07_20180426144405.zip	John Hartfiel	• new assembly variant
2018-03-12	2017.4	te0720-test_board_noprebuilt-vivado_2017.4-build_06_20180312152408.zip te0720-test_board-vivado_2017.4-build_06_20180312152419.zip	John Hartfiel	• add assembly variant • script update

Date	Vivado	Project Built	Authors	Description
2018-01-09	2017.4	te0720-test_board_noprebuilt-vivado_2017.4-build_02_20180109121313.zip te0720-test_board-vivado_2017.4-build_02_20180109121300.zip	John Hartfiel	• no design changes • set EEPROM MAC with FSBL+u-boot • FSBL for QSPI Programming
2017-11-27	2017.2	te0720-test_board_noprebuilt-vivado_2017.2-build_05_20171127153028.zip te0720-test_board-vivado_2017.2-build_05_20171127153006.zip	John Hartfiel	• remove duplicated content
2017-11-20	2017.2	te0720-test_board_noprebuilt-vivado_2017.2-build_05_20171122074701.zip te0720-test_board-vivado_2017.2-build_05_20171122074646.zip	John Hartfiel	• initial release

Table 1: Design Revision History

4.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
TE0720-test_board_noprebuilt-vivado_2020.2-build_2_20210217064925.zip TE0720-test_board-vivado_2020.2-build_2_20210217064913.zip	Linux binaries are missing boot.scr are only prepared for SD Boot	create and modify by yourself or use 2019.2 design	solved with 2020-04-01 update
Variant with 256MB DDR only(TE0720-03-1CR)	wrong netboot offset	recreate u-boot on petalinux with reduced netboot offset only	solved with 2019-03-04 update

Table 2: Known Issues

4.4 Requirements

4.4.1 Software

Software	Version	Note
Vitis	2020.2	needed, Vivado is included into Vitis installation
PetaLinux	2020.2	needed

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 "<project folder>\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
TE0720-03-2IF	2if_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-2IFC3	2if_1gb	REV03| REV02	1GB	32MB	4GB	2.5 mm connectors	low profile
TE0720-03-2IFC8	2if_1gb	REV03| REV02	1GB	32MB	32GB	NA	NA
TE0720-03-1QF	1qf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CF*	1cf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CFA	1cf_1gb	REV03| REV02	1GB	32MB	8GB	NA	NA
TE0720-03-1CR	1cr_256mb	REV03| REV02	256 MB	32MB	NA	NA	NA
TE0720-03-L1IF	l1if_512mb	REV03| REV02	512 MB	32MB	4GB	NA	LP DDR3
TE0720-03-14S-1C	14s_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1QFA	1qf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-2IFA	2if_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1QFL	1qf_1gb	REV03| REV02	1GB	32MB	4GB	2.5 mm connectors	low profile
TE0720-03-31C33FA	14s_1gb	REV03	1GB	32MB	8GB	NA	NA

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-61C33FA	1cf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61C530A	1cr_256mb	REV03	256MB	32MB	NA	NA	NA
TE0720-03-61Q33FA	1qf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61Q33FL	1qf_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	low profile
TE0720-03-61Q42GA	1qf_256mb	REV03	256MB	32MB	32GB	NA	NA
TE0720-03-61Q43FA	1qf_256mb	REV03	256MB	32MB	8GB	NA	NA
TE0720-03-61Q43GA	1qf_256mb	REV03	256MB	32MB	32GB	NA	NA
TE0720-03-61Q86KL	1qf_1gb	REV03	1GB	32MB	8GB	NA	Automotive DDR and QSPI
TE0720-03-62I33GA	2if_1gb	REV03	1GB	32MB	32GB	NA	NA
TE0720-03-62I12GA	2if_1gb	REV03	1GB	32MB	32GB	NA	NA
TE0720-03-62I320M	2if_1gb	REV03	1GB	32MB	NA	NA	CAO: no Eth USB RTC VBAT CryptoKey

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-62I330M	2if_1gb	REV03	1GB	32MB	NA	NA	CAO: no Eth USB RTC VBAT CryptoKey
TE0720-03-62I33FA	2if_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-62I33FL	2if_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	low profile
TE0720-03-64I63FA	l1if_512mb	REV03	512MB	32MB	8GB	NA	LP DDR3
TE0720-03-1QFY	1qf_1gb	REV03	1GB	32MB	4GB	NA	no RTC
TE0720-03-31C33MA	14s_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61C33MAS	1cf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61Q33MA	1qf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61Q33MAY	1qf_1gb	REV03	1GB	32MB	8GB	NA	no RTC
TE0720-03-61Q33ML	1qf_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	low profile
TE0720-03-61Q42GAY	1qf_256mb	REV03	256MB	32MB	32GB	NA	no RTC

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-61Q43MA	1qf_256mb	REV03	256 MB	32MB	8GB	NA	automotive Zynq mit IME1G16D3EEBG-15:EI
TE0720-03-62I33MA	2if_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-62I33MAN	2if_1gb	REV03	1GB	32MB	8GB	NA	industrieller Temperaturbereich; coated
TE0720-03-62I33MAY	2if_1gb	REV03	1GB	32MB	8GB	NA	no RTC
TE0720-03-62I33ML	2if_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	low profile
TE0720-03-62I33NA	2if_1gb	REV03	1GB	32MB	32GB	NA	NA
TE0720-03-S006C1	1qf_1gb	REV03	1GB	32MB	8GB	NA	custom variant
TE0720-03-S007C1	1qf_1gb	REV03	1GB	32MB	8GB	NA	custom variant
TE0720-03-S011	2if_1gb	REV03	1GB	32MB	8GB	NA	custom variant, no ETH
TE0720-03-S012	2if_1gb	REV03	1GB	32MB	8GB	NA	custom variant
TE0720-03-S014	2if_1gb	REV03	1GB	32MB	8GB	NA	custom variant
TE0720-03-S016	1cr_256mb	REV03	256 MB	32MB	NA	NA	custom variant, no RTC

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-S017	2if_1gb	REV03	1GB	32MB	8GB	NA	custom variant, no RTC
TE0720-03-61C33MA	1cf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61C33MAY	1cf_1gb	REV03	1GB	32MB	8GB	NA	no RTC
TE0720-03-62I33-V1	2if_1gb	REV03	1GB	32MB	NA	NA	NA
TE0720-03-S013	1cf_1gb	REV03	1GB	32MB	8GB	NA	custom variant

Table 4: Hardware Modules

*used as reference

Design supports following carriers:

Carrier Model	Notes
TE0701	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[2]
TE0703*	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[3] • Used as reference carrier.
TE0705	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[4]
TE0706	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[5]
TEBA0841	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[6] • No SD Slot available, pins goes to Pin Header • For TEBA0841 REV01, please contact TE support

Table 5: Hardware Carrier

2 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
3 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
4 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
5 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
6 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers

*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

4.5 Content

For general structure and usage of the reference design, see Project Delivery - Xilinx devices[7]

4.5.1 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

4.5.2 Additional Sources

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

Type	Location	Notes
init.sh	<project folder>\misc\sd\	Additional Initialization Script for Linux

Table 8: Additional design sources

4.5.3 Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIN-File	*.bin	Flash Configuration File with Boot-Image (Zynq-FPGAs)
BIT-File	*.bit	FPGA (PL Part) Configuration File
Boot Source	*.scr	Distro Boot 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
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

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

4.5.4 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:

- TE0720 "Test Board" Reference Design[8]

8 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0720/Reference_Design/2020.2/test_board

5 Design Flow

⚠ Reference Design is available with and without prebuilt files. It's recommended to use TE prebuilt files for first launch.

Trenz Electronic provides a tcl based built environment based on Xilinx Design Flow.
See also:

- Xilinx Development Tools[9]
- Vivado Projects - TE Reference Design[10]
- Project Delivery.[11]

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[12]

⚠ **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.

9 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
10 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
11 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices
12 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDeliveryXilinxdevices-Currentlylimitationsoffunctionality

- 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[13]

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[14]
   - 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
6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr[15]
7. Copy PetaLinux build image files to prebuilt folder
   - copy **u-boot.elf**, **image.ub** and **boot.scr** from "<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

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[16]

13 https://wiki.trenz-electronic.de/display/PD/Vivado+Board+Part+Flow
14 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
15 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr
16 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/Vitis/SDSoC-Xilinx Software Programming and Debugging[17]

6.1.1 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

6.1.2 QSPI-Boot mode

Option for **Boot.bin** on QSPI Flash and **image.ub** and **boot.scr** on **SD** or **USB**.

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
   TE::pr_program_flash -swapp hello_te0720 (optional)
   ```

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

3. Copy **image.ub** and **boot.scr** on **SD** or **USB**
   - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder,see: Get prebuilt boot binaries(see page 24)
   - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
4. Set Boot Mode to **QSPI-Boot** and insert **SD** or **USB**.
   - Depends on Carrier, see carrier TRM.

6.1.3 SD-Boot mode

1. Copy **image.ub**, **boot.src** and **Boot.bin** on **SD**

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

- use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder, see: Get prebuilt boot binaries(see page 24)
- or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"

2. Set Boot Mode to SD-Boot.
   - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

6.1.4 JTAG

Not used on this Example.

6.2 Usage

1. Prepare HW like described on section Programming(see page 24)
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)

   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[18]

4. Power On PCB
   boot process

   1. Zynq Boot ROM loads FSBL from SD/QSPI into OCM,

   2. FSBL init PS, programs PL using the bitstream and loads U-boot from SD into DDR,

   3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

6.2.1 Linux

1. Open Serial Console (e.g. putty)
   - Speed: 115200
   - select COM Port

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

2. Linux Console:

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

18 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr

> Note: Wait until Linux boot finished

3. You can use Linux shell now.

```
i2cdetect -y -r 0   (check I2C 0 Bus)
i2cdetect -y -r 1   (check I2C 1 Bus)
dmesg | grep rtc    (RTC check)
udhcpc              (ETH0 check)
lsusb               (USB check)
```

4. Option Features
    - Webserver to get access to Zynq
        - insert IP on web browser to start web interface
    - init.sh scripts
        - add init.sh script on SD, content will be load automatically on startup (template included in "<project folder>\misc\SD")

6.2.2 Vivado HW Manager

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

- Monitoring: PHY LED

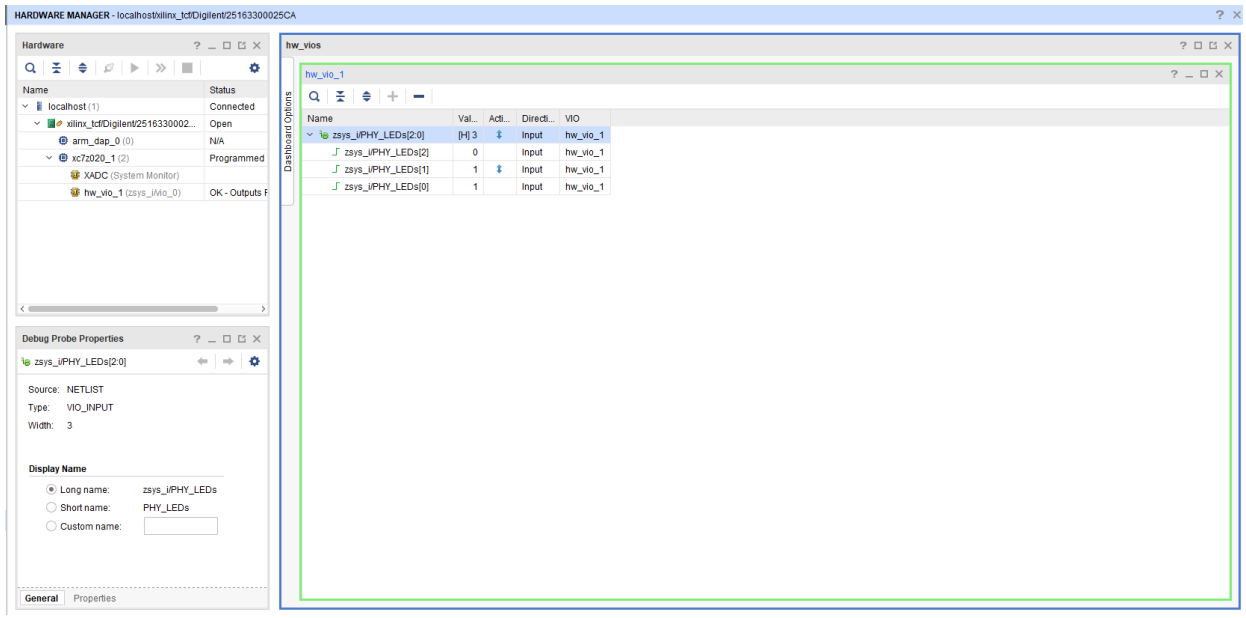

Figure 1: Vivado Hardware Manager

7 System Design - Vivado

7.1 Block Design

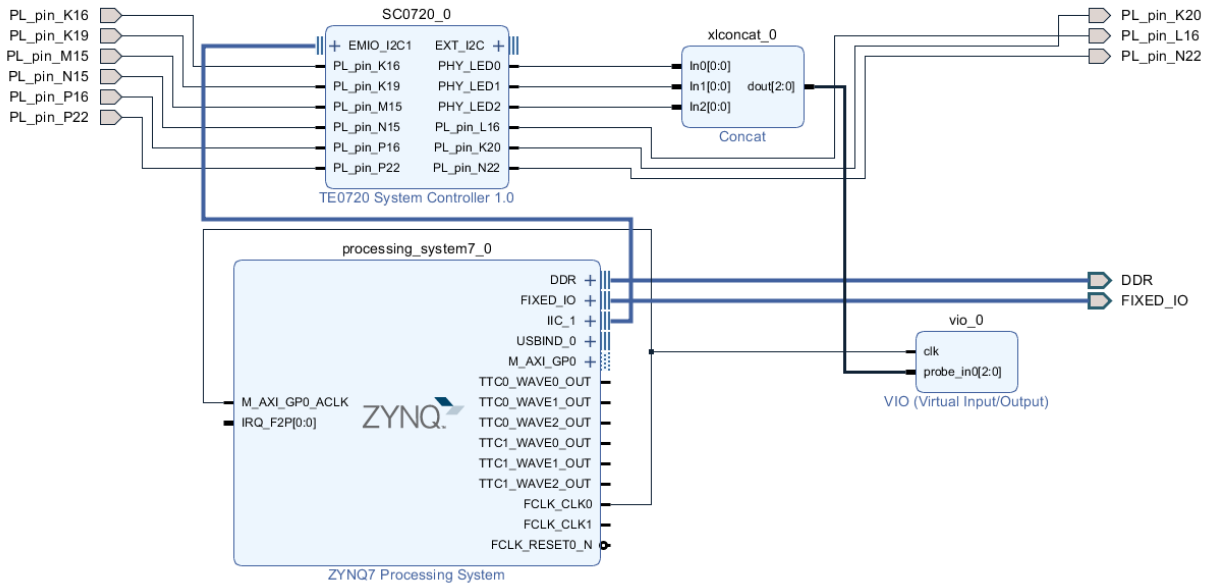

Figure 2: Block Design

7.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	---
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
I2C1	EMIO
UART0	MIO
UART1	MIO

Type	Note
GPIO	MIO
SWDT	EMIO
TTC0..1	EMIO
ETH0	MIO
USB0	MIO

Table 10: PS Interfaces

7.2 Constrains

7.2.1 Basic module constrains

_i_bitgen_common.xdc

```
#
# Common BITGEN related settings for TE0720 SoM
#
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design
```

_i_common.xdc

```
#
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
```

7.2.2 Design specific constrain

_i_TE0720-SC.xdc

```
#
# Constraints for System controller support logic
#
set_property PACKAGE_PIN K16 [get_ports PL_pin_K16]
set_property PACKAGE_PIN K19 [get_ports PL_pin_K19]
set_property PACKAGE_PIN K20 [get_ports PL_pin_K20]
set_property PACKAGE_PIN L16 [get_ports PL_pin_L16]
set_property PACKAGE_PIN M15 [get_ports PL_pin_M15]
```

```
set_property PACKAGE_PIN N15 [get_ports PL_pin_N15]
set_property PACKAGE_PIN N22 [get_ports PL_pin_N22]
set_property PACKAGE_PIN P16 [get_ports PL_pin_P16]
set_property PACKAGE_PIN P22 [get_ports PL_pin_P22]

#
# If Bank 34 is not 3.3V Powered need change the IOSTANDARD
#
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_P22]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_P16]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_N22]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_N15]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_M15]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_L16]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K20]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K19]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K16]
```

8 Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis[19]

8.1 Application

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

8.1.1 fsbl

TE modified 2020.2 FSBL

General:

- Modified Files: main.c, fsbl_hooks.h/.c (search for 'TE Mod' on source code)
- Add Files: te_fsbl_hooks.h/.c (for hooks and board)
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- Add Files: all TE Files start with te_*
    - READ MAC from EEPROM and make Address accessible by UBOOT (need copy defines on uboot platform-top.h)
    - CPLD access
    - Read CPLD Firmware and SoC Type
    - Configure Marvell PHY
    - USB PHY Reset
    - Configure LED usage

8.1.2 fsbl_flash

TE modified 2020.2 FSBL

General:

- Modified Files: main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

8.1.3 hello_te0720

Hello World App in Endless loop.

8.1.4 u-boot

U-Boot.elf is generated with PetaLinux. Vitis is used to generate Boot.bin.

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

9 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[20]

9.1 Config

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

Changes:

- CONFIG_SUBSYSTEM_SERIAL_PS7_UART_0_SELECT=y
- CONFIG_SUBSYSTEM_SERIAL_IP_NAME="ps7_uart_0"

Note: for variants with 256MB DDR only, change NET Boot Address to 0x8000000 on **boot.src** file

9.2 U-Boot

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

Changes:

- CONFIG_QSPI_BOOT=y
- CONFIG_SD_BOOT=y
- CONFIG_ENV_IS_NOWHERE=y
- # CONFIG_ENV_IS_IN_SPI_FLASH is not set

Change platform-top.h:

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

#define CONFIG_PREBOOT "echo U-BOOT for petalinux;echo importing env from FSBL
shared area at 0xFFFFFC00; if itest *0xFFFFFC00 == 0xCAFEBABE; then echo Found
valid magic; env import -t 0xFFFFFC04; fi;setenv preboot; echo; dhcp"
```

9.3 Device Tree

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

/* default */

/* QSPI PHY */
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
```

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

```
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
    };
};


/* ETH PHY */
&gem0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <0>;
        };
    };
};

/* USB PHY */

/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        //compatible = "usb-nop-xceiv";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb0 {
    dr_mode = "host";
    //dr_mode = "peripheral";
    usb-phy = <&usb_phy0>;
};

/* I2C need I2C1 connected to te0720 system controller ip */
&i2c1 {

    iexp@20 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x20>;
        gpio-controller;
    };

    iexp@21 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x21>;
```

```
        gpio-controller;
    };

    rtc@6F {        // Real Time Clock
        compatible = "isl12022";
        reg = <0x6F>;
    };
};
```

9.4 FSBL patch

Must be add manually, see template

9.5 Kernel

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

Changes:

- CONFIG_RTC_DRV_ISL12022=y

9.6 Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_packagegroup-petalinux-utils(util-linux,cpufrequtils,bridge-utils,mtd-utils,usbutils,pciutils,canutils,i2c-tools,smartmontools,e2fsprogs)

9.7 Applications

See "<project folder>\os\petalinux\project-spec\meta-user\recipes-apps\"

9.7.1 startup

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

9.7.2 webfwu

Webserver application suitable for Zynq access. Need busybox-httpd

10 Additional Software

No additional software is needed.

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
2021-12-16	v.49(see page 7)	@ Manuela Strücker[21]	• boot.scr file updated for 256 MB QSPI flash size variants
2021-05-25	v.44	Manuela Strücker	• update board files • update boot.scr file
2021-04-01	v.42	@John Hartfiel	• Design update
2021-02-26	v.41	John Hartfiel	• add issue notes
2021-02-17	v.40	John Hartfiel	• 2020.2 release
2020-03-25	v.39	John Hartfiel	• script update
2020-01-22	v.38	John Hartfiel	• script update for linux user
2020-01-14	v.37	John Hartfiel	• Vitis script updates (include linux domain and prebuilt linux files for vitis) • prebuilt binary export on selection guide
2019-12-19	v.36	John Hartfiel	• 2019.2 release
2019-12-03	v.34	John Hartfiel	• bugfix document link
2019-10-28	v.33	John Hartfiel	• removed remove instructions that are no longer used

21 https://wiki.trenz-electronic.de/display/~m.struecker

Date	Document Revision	Authors	Description
2019-05-07	v.31	John Hartfiel	• Some FSBL notes • wrong link
2019-03-06	v.28	John Hartfiel	• Fixed prebuilt issue for TE0720-03-1CR
2019-03-01	v.27	John Hartfiel	• Known issue for TE0720-03-1CR linux design
2019-02-21	v.26	John Hartfiel	• 2018.3 release finished (include design reworks)
2018-08-30	v.25	John Hartfiel	• update documentation PS configuration
2018-08-23	v.24	John Hartfiel	• update l1if board parts
2018-08-13	v.23	John Hartfiel	• 2018.4 release
2018-04-26	v.22	John Hartfiel	• add assembly variant
2018-02-20	v.20	John Hartfiel	• small documentation update
2018-01-09	v.16	John Hartfiel	• Release 2017.4 • Documentation update
2017-11-27	v.14	John Hartfiel	• Typo correction • Design Files update
2017-11-22	v.12	John Hartfiel	• Update HW list
2017-11-22	v.11	John Hartfiel	• Release 2017.2
2017-11-20	v.1	@ John Hartfiel[22]	• Initial release

22 https://wiki.trenz-electronic.de/display/~j.hartfiel

Date	Document Revision	Authors	Description
--	All	@ Antonio Luppi[23], John Hartfiel[24], Manuela Strücker[25]	--

Table 11: 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.

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.

23 https://wiki.trenz-electronic.de/display/~a.luppi
24 https://wiki.trenz-electronic.de/display/~j.hartfiel
25 https://wiki.trenz-electronic.de/display/~m.struecker

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[26]. 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[27] 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)[28].

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.

26 http://guidance.echa.europa.eu/
27 https://echa.europa.eu/candidate-list-table
28 http://www.echa.europa.eu/

2019-06-07

12 Overview

Zynq PS Design with Linux Example and PHY status LED on Vivado HW-Manager.
Refer to http://trenz.org/te0720-info for the current online version of this manual and other available documentation.

12.1 Key Features

- Vitis/Vivado 2020.2
- PetaLinux
- SD
- ETH (use EEPROM MAC)
- USB
- I2C
- RTC
- VIO PHY LED
- FSBL for EEPROM MAC and CPLD access / petalinux patch
- Special FSBL for QSPI Programming

12.2 Revision History

Date	Vivado	Project Built	Authors	Description
2021-12-15	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_9_20211215123252.zip TE0720-test_board-vivado_2020.2-build_9_20211215123235.zip	Manuela Strücker	• new assembly variants
2021-07-19	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_6_20210719131800.zip TE0720-test_board-vivado_2020.2-build_6_20210719131744.zip	Manuela Strücker	• boot.scr file updated for 256 MB QSPI flash size variants
2021-04-30	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-	Manuela Strücker	• update board files • update boot.scr file

Date	Vivado	Project Built	Authors	Description
		build_5_20210430085624.zip TE0720-test_board-vivado_2020.2-build_5_20210430085609.zip		
2021-04-01	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_4_20210401140444.zip TE0720-test_board-vivado_2020.2-build_4_20210401140432.zip	John Hartfiel	• bugfix missing binaries+ boot.scr file(supports now QSPI and SD boot with image.ub on SD)
2021-02-17	2020.2	TE0720-test_board_noprebuilt-vivado_2020.2-build_2_20210217064925.zip TE0720-test_board-vivado_2020.2-build_2_20210217064913.zip	John Hartfiel	• 2020.2 update • add boot.scr file • petalinux fsbl patch (beta-version)
2020-03-25	2019.2	TE0720-test_board_noprebuilt-vivado_2019.2-build_8_20200325075220.zip TE0720-test_board-vivado_2019.2-build_8_20200325075301.zip	John Hartfiel	• script update
2020-01-22	2019.2	TE0720-test_board-vivado_2019.2-build_3_20200122154933.zip TE0720-test_board_noprebuilt-vivado_2019.2-	John Hartfiel	• script update for linux user

Date	Vivado	Project Built	Authors	Description
		build_3_20200122154951.zip		
2020-01-14	2019.2	TE0720-test_board-vivado_2019.2-build_3_20200114090828.zip TE0720-test_board_noprebuilt-vivado_2019.2-build_3_20200114090837.zip	John Hartfiel	• Vitis script updates (include linux domain and prebuilt linux files for vitis) • prebuilt binary export on selection guide
2019-12-18	2019.2	TE0720-test_board-vivado_2019.2-build_1_20191218151902.zip TE0720-test_board_noprebuilt-vivado_2019.2-build_1_20191218152732.zip	John Hartfiel	• 2019.2 update • Vitis support
2019-03-04	2018.3	TE0720-test_board-vivado_2018.3-build_01_20190304100745.zip TE0720-test_board_noprebuilt-vivado_2018.3-build_01_20190304100755.zip	John Hartfiel	• update for -1CR version only (256MB DDR3)
2019-02-21	2018.3	TE0720-test_board-vivado_2018.3-build_01_20190221125123.zip TE0720-test_board_noprebuilt-vivado_2018.3-build_01_20190221125133.zip	John Hartfiel	• TE Script update • rework of the FSBLs • some additional Linux features

Date	Vivado	Project Built	Authors	Description
2018-08-23	2018.2	te0720-test_board-vivado_2018.2-build_03_20180823185142.zip te0720-test_board_noprebuilt-vivado_2018.2-build_03_20180823185158.zip	John Hartfiel	• DDR setup bugfix for l1if only
2018-08-13	2018.2	te0720-test_board-vivado_2018.2-build_02_20180810162024.zip te0720-test_board_noprebuilt-vivado_2018.2-build_02_20180810162040.zip	John Hartfiel	• 2018.2 update • Board Part Files rework
2018-04-26	2017.4	te0720-test_board-vivado_2017.4-build_07_20180426144351.zip te0720-test_board_noprebuilt-vivado_2017.4-build_07_20180426144405.zip	John Hartfiel	• new assembly variant
2018-03-12	2017.4	te0720-test_board_noprebuilt-vivado_2017.4-build_06_20180312152408.zip te0720-test_board-vivado_2017.4-build_06_20180312152419.zip	John Hartfiel	• add assembly variant • script update
2018-01-09	2017.4	te0720-test_board_noprebuilt-vivado_2017.4-build_02_20180109121	John Hartfiel	• no design changes • set EEPROM MAC with FSBL+u-boot

Date	Vivado	Project Built	Authors	Description
		313.zip te0720-test_board-vivado_2017.4-build_02_20180109121300.zip		• FSBL for QSPI Programming
2017-11-27	2017.2	te0720-test_board_noprebuilt-vivado_2017.2-build_05_20171127153028.zip te0720-test_board-vivado_2017.2-build_05_20171127153006.zip	John Hartfiel	• remove duplicated content
2017-11-20	2017.2	te0720-test_board_noprebuilt-vivado_2017.2-build_05_20171122074701.zip te0720-test_board-vivado_2017.2-build_05_20171122074646.zip	John Hartfiel	• initial release

Table 12: Design Revision History

12.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
TE0720-test_board_noprebuilt-vivado_2020.2-build_2_20210217064925.zip TE0720-test_board-vivado_2020.2-build_2_20210217064913.zip	Linux binaries are missing boot.scr are only prepared for SD Boot	create and modify by yourself or use 2019.2 design	solved with 2020-04-01 update

Issues	Description	Workaround	To be fixed version
Variant with 256MB DDR only(TE0720-03-1CR)	wrong netboot offset	recreate u-boot on petalinux with reduced netboot offset only	solved with 2019-03-04 update

Table 13: Known Issues

12.4 Requirements

12.4.1 Software

Software	Version	Note
Vitis	2020.2	needed, Vivado is included into Vitis installation
PetaLinux	2020.2	needed

Table 14: Software

12.4.2 Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.[29]
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
TE0720-03-2IF	2if_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-2IFC3	2if_1gb	REV03| REV02	1GB	32MB	4GB	2.5 mm connectors	NA
TE0720-03-2IFC8	2if_1gb	REV03| REV02	1GB	32MB	32GB	NA	NA

29 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
TE0720-03-1QF	1qf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CF*	1cf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CFA	1cf_1gb	REV03| REV02	1GB	32MB	8GB	NA	NA
TE0720-03-1CR	1cr_256mb	REV03| REV02	256MB	32MB	NA	NA	NA
TE0720-03-L1IF	1if_512mb	REV03| REV02	512MB	32MB	4GB	NA	LP DDR3
TE0720-03-14S-1C	14s_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1QFA	1qf_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-2IFA	2if_1gb	REV03| REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1QFL	1qf_1gb	REV03| REV02	1GB	32MB	4GB	2.5 mm connectors	NA
TE0720-03-31C33FA	14s_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61C33FA	1cf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61C530A	1cr_256mb	REV03	256MB	32MB	NA	NA	NA
TE0720-03-61Q33FA	1qf_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-61Q33FL	1qf_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	NA

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0720-03-61Q86KL	1qf_1gb	REV03	1GB	32MB	8GB	NA	Automotive DDR and QSPI
TE0720-03-62I33GA	2if_1gb	REV03	1GB	32MB	32GB	NA	NA
TE0720-03-62I12GA	2if_1gb	REV03	1GB	32MB	32GB	NA	NA
TE0720-03-62I320M	2if_1gb	REV03	1GB	32MB	NA	NA	CAO: no Ethernet USB RTC VBAT CryptoKey
TE0720-03-62I330M	2if_1gb	REV03	1GB	32MB	NA	NA	CAO: no Ethernet USB RTC VBAT CryptoKey
TE0720-03-62I33FA	2if_1gb	REV03	1GB	32MB	8GB	NA	NA
TE0720-03-62I33FL	2if_1gb	REV03	1GB	32MB	8GB	2.5 mm connectors	NA
TE0720-03-64I63FA	1if_512mb	REV03	512MB	32MB	8GB	NA	LP DDR3

Table 15: Hardware Modules

*used as reference

Design supports following carriers:

Carrier Model	Notes
TE0701	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[30]

30 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers

Carrier Model	Notes
TE0703*	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[31] • Used as reference carrier.
TE0705	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[32]
TE0706	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[33]
TEBA0841	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers[34] • No SD Slot available, pins goes to Pin Header • For TEBA0841 REV01, please contact TE support

Table 16: 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 17: Additional Hardware

12.5 Content

For general structure and usage of the reference design, see Project Delivery - Xilinx devices[35]

12.5.1 Design Sources

Type	Location	Notes
Vivado	<project folder>\block_design <project folder>\constraints <project folder>\ip_lib	Vivado Project will be generated by TE Scripts

31 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
32 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
33 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
34 https://wiki.trenz-electronic.de/display/PD/4+x+5+SoM+Carriers
35 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices

Type	Location	Notes
	<project folder>\board_files	
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 18: Design sources

12.5.2 Additional Sources

Type	Location	Notes
init.sh	<project folder>\misc\sd\	Additional Initialization Script for Linux

Table 19: Additional design sources

12.5.3 Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIN-File	*.bin	Flash Configuration File with Boot-Image (Zynq-FPGAs)
BIT-File	*.bit	FPGA (PL Part) Configuration File
Boot Source	*.scr	Distro Boot 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

File	File-Extension	Description
LabTools Project-File	*.lpr	Vivado Labtools Project File
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

Table 20: Prebuilt files (only on ZIP with prebult content)

12.5.4 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:

- TE0720 "Test Board" Reference Design[36]

36 https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0720/Reference_Design/2020.2/test_board

13 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[37]
- Vivado Projects - TE Reference Design[38]
- Project Delivery.[39]

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[40]

⚠ **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.

37 https://wiki.trenz-electronic.de/display/PD/Xilinx+Development+Tools#XilinxDevelopmentTools-XilinxSoftware-BasicUserGuides
38 https://wiki.trenz-electronic.de/display/PD/Vivado+Projects+-+TE+Reference+Design
39 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices
40 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+Xilinx+devices#ProjectDeliveryXilinxdevices-Currentlylimitationsoffunctionality

- 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[41]

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[42]
   - 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
6. Configure the **boot.scr** file as needed, see Distro Boot with Boot.scr[43]
7. Copy PetaLinux build image files to prebuilt folder
   - copy **u-boot.elf**, **image.ub** and **boot.scr** from "<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

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[44]

41 https://wiki.trenz-electronic.de/display/PD/Vivado+Board+Part+Flow
42 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart
43 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr
44 https://wiki.trenz-electronic.de/display/PD/Vitis

14 Launch

14.1 Programming

⚠ Check Module and Carrier TRMs for proper HW configuration before you try any design.

Xilinx documentation for programming and debugging: Vivado/Vitis/SDSoC-Xilinx Software Programming and Debugging[45]

14.1.1 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

14.1.2 QSPI-Boot mode

Option for **Boot.bin** on QSPI Flash and **image.ub** and **boot.scr** on **SD** or **USB**.

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
   TE::pr_program_flash -swapp hello_te0720 (optional)
   ```

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

3. Copy **image.ub** and **boot.scr** on **SD** or **USB**
   - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder,see: Get prebuilt boot binaries(see page 24)
   - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
4. Set Boot Mode to **QSPI-Boot** and insert **SD** or **USB**.
   - Depends on Carrier, see carrier TRM.

14.1.3 SD-Boot mode

1. Copy **image.ub**, **boot.src** and **Boot.bin** on **SD**

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

- use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder, see: Get prebuilt boot binaries(see page 24)
- or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"

2. Set Boot Mode to SD-Boot.
   - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

14.1.4 JTAG

Not used on this Example.

14.2 Usage

1. Prepare HW like described on section Programming(see page 24)
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)

   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[46]

4. Power On PCB
   boot process

   1. Zynq Boot ROM loads FSBL from SD/QSPI into OCM,
   2. FSBL init PS, programs PL using the bitstream and loads U-boot from SD into DDR,
   3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

14.2.1 Linux

1. Open Serial Console (e.g. putty)
   - Speed: 115200
   - select COM Port

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

2. Linux Console:

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

46 https://wiki.trenz-electronic.de/display/PD/Distro+Boot+with+Boot.scr

> Note: Wait until Linux boot finished

3. You can use Linux shell now.

```
i2cdetect -y -r 0   (check I2C 0 Bus)
i2cdetect -y -r 1   (check I2C 1 Bus)
dmesg | grep rtc    (RTC check)
udhcpc              (ETH0 check)
lsusb               (USB check)
```

4. Option Features
    - Webserver to get access to Zynq
        - insert IP on web browser to start web interface
    - init.sh scripts
        - add init.sh script on SD, content will be load automatically on startup (template included in "<project folder>\misc\SD")

14.2.2 Vivado HW Manager

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

- Monitoring: PHY LED

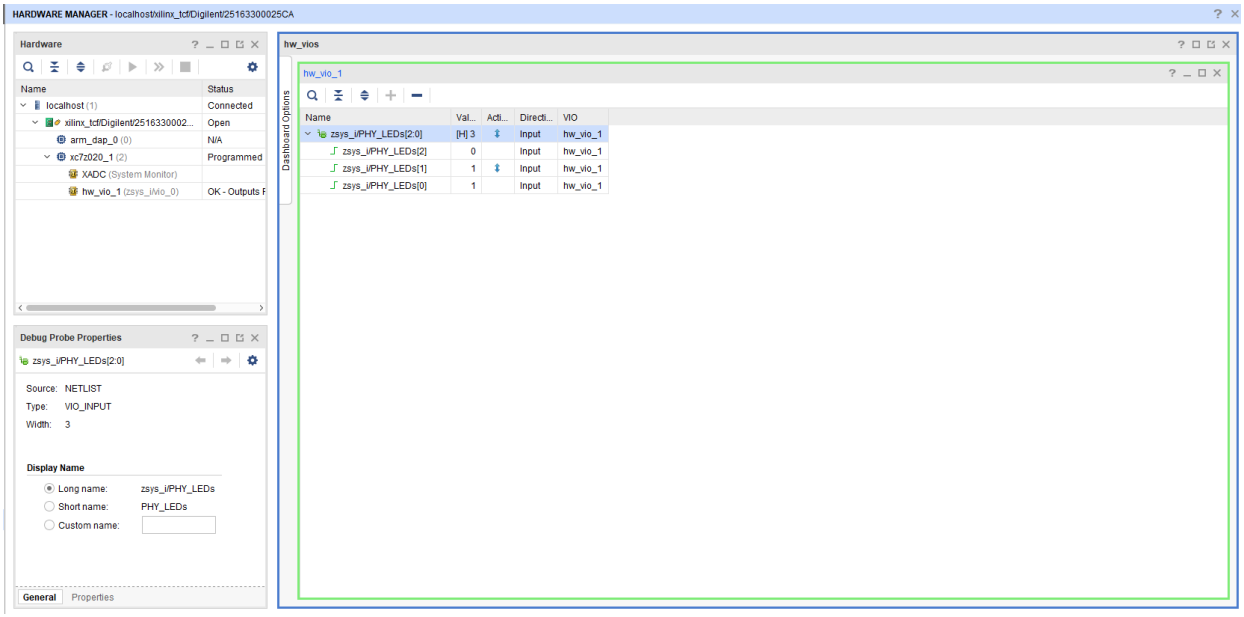

Figure 3: Vivado Hardware Manager

15 System Design - Vivado

15.1 Block Design

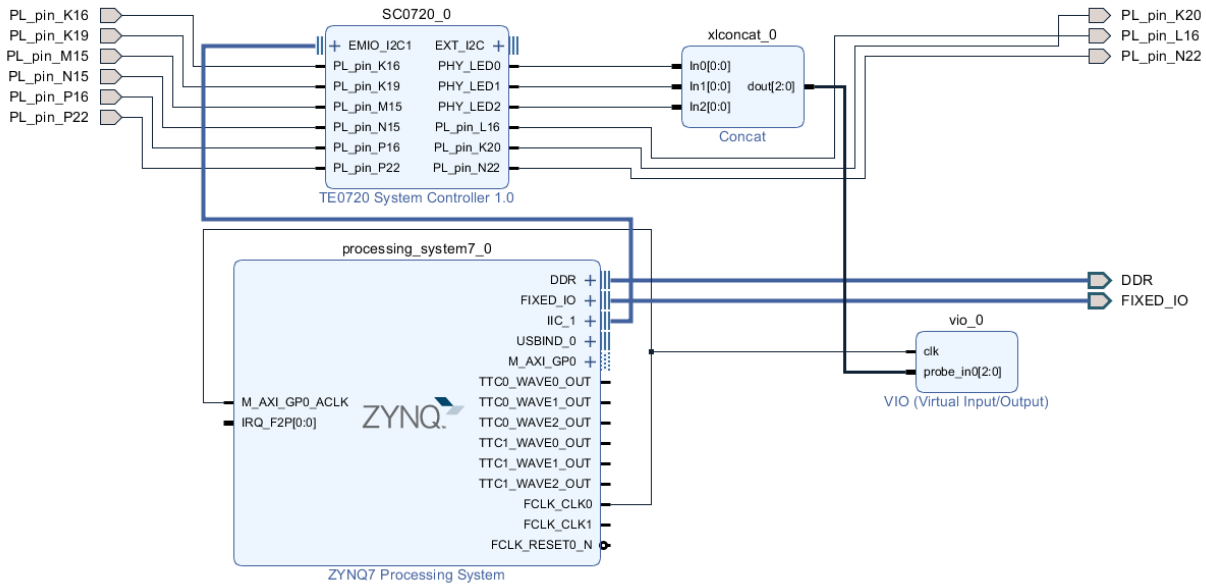

Figure 4: Block Design

15.1.1 PS Interfaces

Activated interfaces:

Type	Note
DDR	---
QSPI	MIO
SD0	MIO
SD1	MIO
I2C0	MIO
I2C1	EMIO
UART0	MIO
UART1	MIO

Type	Note
GPIO	MIO
SWDT	EMIO
TTC0..1	EMIO
ETH0	MIO
USB0	MIO

Table 21: PS Interfaces

15.2 Constrains

15.2.1 Basic module constrains

_i_bitgen_common.xdc

```
#
# Common BITGEN related settings for TE0720 SoM
#
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design
```

_i_common.xdc

```
#
set_property BITSTREAM.CONFIG.UNUSEDPIN PULLUP [current_design]
```

15.2.2 Design specific constrain

_i_TE0720-SC.xdc

```
#
# Constraints for System controller support logic
#
set_property PACKAGE_PIN K16 [get_ports PL_pin_K16]
set_property PACKAGE_PIN K19 [get_ports PL_pin_K19]
set_property PACKAGE_PIN K20 [get_ports PL_pin_K20]
set_property PACKAGE_PIN L16 [get_ports PL_pin_L16]
set_property PACKAGE_PIN M15 [get_ports PL_pin_M15]
```

```
set_property PACKAGE_PIN N15 [get_ports PL_pin_N15]
set_property PACKAGE_PIN N22 [get_ports PL_pin_N22]
set_property PACKAGE_PIN P16 [get_ports PL_pin_P16]
set_property PACKAGE_PIN P22 [get_ports PL_pin_P22]

#
# If Bank 34 is not 3.3V Powered need change the IOSTANDARD
#
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_P22]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_P16]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_N22]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_N15]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_M15]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_L16]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K20]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K19]
set_property IOSTANDARD LVCMOS33 [get_ports PL_pin_K16]
```

16 Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis[47]

16.1 Application

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

16.1.1 fsbl

TE modified 2020.2 FSBL

General:

- Modified Files: main.c, fsbl_hooks.h/.c (search for 'TE Mod' on source code)
- Add Files: te_fsbl_hooks.h/.c (for hooks and board)
- General Changes:
    - Display FSBL Banner and Device ID

Module Specific:

- Add Files: all TE Files start with te_*
    - READ MAC from EEPROM and make Address accessible by UBOOT (need copy defines on uboot platform-top.h)
    - CPLD access
    - Read CPLD Firmware and SoC Type
    - Configure Marvell PHY
    - USB PHY Reset
    - Configure LED usage

16.1.2 fsbl_flash

TE modified 2020.2 FSBL

General:

- Modified Files: main.c
- General Changes:
    - Display FSBL Banner
    - Set FSBL Boot Mode to JTAG
    - Disable Memory initialisation

16.1.3 hello_te0720

Hello World App in Endless loop.

16.1.4 u-boot

U-Boot.elf is generated with PetaLinux. Vitis is used to generate Boot.bin.

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

17 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[48]

17.1 Config

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

Changes:

- CONFIG_SUBSYSTEM_SERIAL_PS7_UART_0_SELECT=y
- CONFIG_SUBSYSTEM_SERIAL_IP_NAME="ps7_uart_0"

Note: for variants with 256MB DDR only, change NET Boot Address to 0x8000000 on **boot.src** file

17.2 U-Boot

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

Changes:

- CONFIG_QSPI_BOOT=y
- CONFIG_SD_BOOT=y
- CONFIG_ENV_IS_NOWHERE=y
- # CONFIG_ENV_IS_IN_SPI_FLASH is not set

Change platform-top.h:

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

#define CONFIG_PREBOOT "echo U-BOOT for petalinux;echo importing env from FSBL
shared area at 0xFFFFFC00; if itest *0xFFFFFC00 == 0xCAFEBABE; then echo Found
valid magic; env import -t 0xFFFFFC04; fi;setenv preboot; echo; dhcp"
```

17.3 Device Tree

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

/* default */

/* QSPI PHY */
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
```

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

```
    flash0: flash@0 {
        compatible = "jedec,spi-nor";
        reg = <0x0>;
        #address-cells = <1>;
        #size-cells = <1>;
    };
};


/* ETH PHY */
&gem0 {
    phy-handle = <&phy0>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy0: phy@0 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <0>;
        };
    };
};

/* USB PHY */

/{
    usb_phy0: usb_phy@0 {
        compatible = "ulpi-phy";
        //compatible = "usb-nop-xceiv";
        #phy-cells = <0>;
        reg = <0xe0002000 0x1000>;
        view-port = <0x0170>;
        drv-vbus;
    };
};

&usb0 {
    dr_mode = "host";
    //dr_mode = "peripheral";
    usb-phy = <&usb_phy0>;
};

/* I2C need I2C1 connected to te0720 system controller ip */
&i2c1 {

    iexp@20 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x20>;
        gpio-controller;
    };

    iexp@21 {       // GPIO in CPLD
        #gpio-cells = <2>;
        compatible = "ti,pcf8574";
        reg = <0x21>;
```

```
        gpio-controller;
    };

    rtc@6F {        // Real Time Clock
        compatible = "isl12022";
        reg = <0x6F>;
    };
};
```

17.4 FSBL patch

Must be add manually, see template

17.5 Kernel

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

Changes:

- CONFIG_RTC_DRV_ISL12022=y

17.6 Rootfs

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

Changes:

- CONFIG_i2c-tools=y
- CONFIG_busybox-httpd=y (for web server app)
- CONFIG_packagegroup-petalinux-utils(util-linux,cpufrequtils,bridge-utils,mtd-utils,usbutils,pciutils,canutils,i2c-tools,smartmontools,e2fsprogs)

17.7 Applications

See "<project folder>\os\petalinux\project-spec\meta-user\recipes-apps\"

17.7.1 startup

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

17.7.2 webfwu

Webserver application suitable for Zynq access. Need busybox-httpd

18 Additional Software

No additional software is needed.

19 Appx. A: Change History and Legal Notices

19.1 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
2021-12-16	v.49(see page 7)	@ Manuela Strücker[49]	• new assembly variants
2021-11-29	v.46	@John Hartfiel	• new assembly variants
2021-07-19	v.45	Manuela Strücker	• boot.scr file updated for 256 MB QSPI flash size variants
2021-05-25	v.44	Manuela Strücker	• update board files • update boot.scr file
2021-04-01	v.42	@John Hartfiel	• Design update
2021-02-26	v.41	John Hartfiel	• add issue notes
2021-02-17	v.40	John Hartfiel	• 2020.2 release
2020-03-25	v.39	John Hartfiel	• script update
2020-01-22	v.38	John Hartfiel	• script update for linux user
2020-01-14	v.37	John Hartfiel	• Vitis script updates (include linux domain and prebuilt linux files for vitis) • prebuilt binary export on selection guide
2019-12-19	v.36	John Hartfiel	• 2019.2 release

49 https://wiki.trenz-electronic.de/display/~m.struecker

Date	Document Revision	Authors	Description
2019-12-03	v.34	John Hartfiel	• bugfix document link
2019-10-28	v.33	John Hartfiel	• removed remove instructions that are no longer used
2019-05-07	v.31	John Hartfiel	• Some FSBL notes • wrong link
2019-03-06	v.28	John Hartfiel	• Fixed prebuilt issue for TE0720-03-1CR
2019-03-01	v.27	John Hartfiel	• Known issue for TE0720-03-1CR linux design
2019-02-21	v.26	John Hartfiel	• 2018.3 release finished (include design reworks)
2018-08-30	v.25	John Hartfiel	• update documentation PS configuration
2018-08-23	v.24	John Hartfiel	• update l1if board parts
2018-08-13	v.23	John Hartfiel	• 2018.4 release
2018-04-26	v.22	John Hartfiel	• add assembly variant
2018-02-20	v.20	John Hartfiel	• small documentation update
2018-01-09	v.16	John Hartfiel	• Release 2017.4 • Documentation update
2017-11-27	v.14	John Hartfiel	• Typo correction • Design Files update
2017-11-22	v.12	John Hartfiel	• Update HW list
2017-11-22	v.11	John Hartfiel	• Release 2017.2

Date	Document Revision	Authors	Description
2017-11-20	v.1	@ John Hartfiel[50]	• Initial release
--	All	@ Antonio Luppi[51], John Hartfiel[52], Manuela Strücker[53]	--

Table 22: Document change history.

19.2 Legal Notices

19.3 Data Privacy

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

19.4 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.

19.5 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.

50 https://wiki.trenz-electronic.de/display/~j.hartfiel
51 https://wiki.trenz-electronic.de/display/~a.luppi
52 https://wiki.trenz-electronic.de/display/~j.hartfiel
53 https://wiki.trenz-electronic.de/display/~m.struecker

19.6 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.

19.7 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.

19.8 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.

19.9 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[54]. 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[55] 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)[56].

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

54 http://guidance.echa.europa.eu/
55 https://echa.europa.eu/candidate-list-table
56 http://www.echa.europa.eu/

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