TE0720 ETH0706

Table of contents

Overview

Zynq PS Design with Linux Example and 2 ETH (On board and TE0706 ETH PHY).

Refer to http://trenz.org/te0720-info for the current online version of this manual and other available documentation.

Key Features

- Vitis/Vivado 2019.2
- PetaLinux
- SD
- ETH (use EEPROM MAC for on board PHY)
- **second ETH from TE0706 carrier**
- USB
- I2C
- RTC
- VIO PHY LED
- FSBL for EEPROM MAC and CPLD access
- Special FSBL for QSPI Programming

Revision History

Date	Vivado	Project Built	Authors	Description
2021-04-20	2019.2	TE0720-ETH706-vivado_2019.2-build_15_20210420080413.zip TE0720-ETH706_noprebuilt-vivado_2019.2-build_15_20210420080424.zip	John Hartfiel	• bugfix contrain files for 1.8V Bankpower
2020-12-11	2019.2	TE0720-ETH706_noprebuilt-vivado_2019.2-build_15_20201211074527.zip TE0720-ETH706-vivado_2019.2-build_15_20201211074146.zip	John Hartfiel	• 2019.2
2017-11-20	2016.2	te0720-TE0706_ETH_noprebuilt-vivado_2016.2-build_08_20161206155126.zip te0720-TE0706_ETH-vivado_2016.2-build_08_20161206155111.zip	John Hartfiel	• initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
Syntax error on device tree for Marvel register configuration	Register configuration for GMII/Cupper is "marvell, reg-init = <0x12 0x14 0x0 0x0200>;", current value was set to "marvell,reg-init = <0x12 0x14 0x0200>;" and will be ignored by drivers. This Bug will not influence functionality and will be fixed on next design update	remove entry from device tree or change to correct syntax.	---

Known Issues

Requirements

Software

Software	Version	Note
Vitis	2019.2	needed, Vivado is included into Vitis installation
PetaLinux	2019.2	needed

Software

Hardware

Basic description of TE Board Part Files is available on TE Board Part Files.

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

Design supports following modules:

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
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
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
TE0720-03-2EF	2ef_1gb	REV03|REV02	1GB	32MB	4GB	NA	NA
TE0720-03-1CR	1cr_256mb	REV03|REV02	256MB	32MB	NA	NA	NA

TE0720-03-L1IF	l1if_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	Micron Flash
TE0720-03-2IFA	2if_1gb	REV03|REV02	1GB	32MB	4GB	NA	Micron Flash
TE0720-03-1QFL	1qf_1gb	REV03|REV02	1GB	32MB	4GB	2.5 mm connectors	NA

Hardware Modules

Design supports following carriers:

Carrier Model	Notes
TE0706	• See restrictions on usage with 7 Series Carriers: 4 x 5 SoM Carriers

Hardware Carrier

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

Additional Hardware

Content

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

Design Sources

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

Design sources

Additional Sources

Type	Location	Notes
init.sh	<design name>/sd/	Additional Initialization Script for Linux

Additional design sources

Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File
BIT-File	*.bit	FPGA (PL Part) Configuration File
DebugProbes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Debian SD-Image	*.img	Debian Image for SD-Card
Diverse Reports	---	Report files in different formats
Hardware-Platform-Specification-Files	*.xsa	Exported Vivado Hardware Specification for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

Prebuilt files (only on ZIP with prebult content)

Download

Reference Design is only usable with the specified Vivado/SDK/PetaLinux/SDx version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0720 "ETH0706" Reference Design

Design Flow

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

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

See also:

- AMD Development Tools#XilinxSoftware-BasicUserGuides
- Vivado Projects - TE Reference Design
- Project Delivery.

The Trenz Electronic FPGA Reference Designs are TCL-script based project. Command files for execution will be generated with "_create_win_setup.cmd" on Windows OS and "_create_linux_setup.sh" on Linux OS.

TE Scripts are only needed to generate the vivado project, all other additional steps are optional and can also executed by Xilinx Vivado/SDK GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality

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

2. Press 0 and enter to start "Module Selection Guide"
3. (optional Win OS) Generate Virtual Drive or use short directory for the reference design (for example x:\<design name>)
4. Create Project (follow instruction of the product selection guide), settings file will be configured automatically during this process
    a. (optional for manual changes) Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"
    Note: Select correct one, see also TE Board Part Files
5. Create XSA and export to prebuilt folder
    a. Run on Vivado TCL: TE::hw_build_design -export_prebuilt
    Note: Script generate design and export files into \prebuilt\hardware\<short dir>. Use GUI is the same, except file export to prebuilt folder
6. Create Linux (uboot.elf and image.ub) with exported XSA
    a. XSA is exported to "prebuilt\hardware\<short name>"
    Note: HW Export from Vivado GUI create another path as default workspace.
    Create Linux images on VM, see PetaLinux KICKstart
        i. Use TE Template from /os/petalinux
7. Add Linux files (uboot.elf and image.ub) to prebuilt folder
    a. "prebuilt\os\petalinux\<ddr size>" or "prebuilt\os\petalinux\<short name>"
    Notes: Scripts select "prebuilt\os\petalinux\<short name>", if exist, otherwise "prebuilt\os\petalinux\<DDR size>" of the selected device
8. Generate Programming Files with Vitis
    - Run on Vivado TCL: TE::sw_run_vitis -all
    Note: Depending of PC performance this can take several minutes. Scripts generate applications and bootable files, which are defined in "sw_lib\apps_list.csv" and open Vitis
    - (alternative) Start Vitis with Vivado GUI or start with TE Scripts on Vivado TCL: TE::sw_run_vitis
    Note: TCL scripts generate also platform project, this must be done manuelly in case GUI is used. See Vitis

Launch

Programming

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

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

Get prebuilt boot binaries

1. _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell
2. Press 0 and enter to start "Module Selection Guide"
    a. Select assembly version
    b. Validate selection
    c. Select Create and open delivery binary folder
    Note: Folder (<project foler>/_binaries_<Artikel Name>) with subfolder (boot_<app name>) for different applications will be generated

QSPI

Optional for Boot.bin on QSPI Flash and image.ub on SD.

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"
3. Type on Vivado TCL Console: TE::pr_program_flash_binfile -swapp u-boot
   Note: To program with SDK/Vivado GUI, use special FSBL (zynq_fsbl_flash) on setup optional "TE::pr_program_flash_binfile -swapp hello_te0720" possible
4. Copy image.ub on SD-Card
    a. use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries
    b. or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
5. Insert SD-Card

SD

1. Copy image.ub and Boot.bin on SD-Card.
    - use files from (<project foler>/_binaries_<Articel Name>)/boot_linux from generated binary folder,see: Get prebuilt boot binaries
    - or use prebuilt file location, see <design_name>/prebuilt/readme_file_location.txt
2. Set Boot Mode to SD-Boot.
    - Depends on Carrier, see carrier TRM.
3. Insert SD-Card in SD-Slot.

JTAG

Not used on this Example.

Usage

1. Prepare HW like described on section TE0720 Test Board#Programming
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.
4. Power On PCB
   Note: 1. Zynq Boot ROM loads FSBL from SD into OCM, 2. FSBL loads U-boot from SD into DDR, 3. U-boot load Linux from SD into DDR

Linux

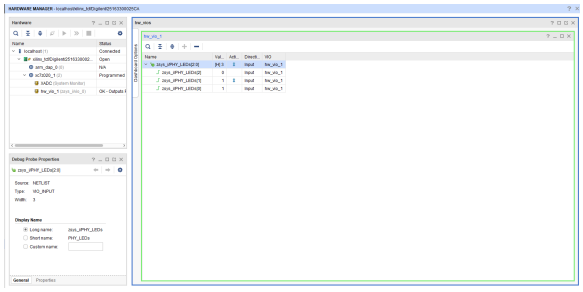
1. Open Serial Console (e.g. putty)
    a. Speed: 115200
    b. COM Port: Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console:
   Note: Wait until Linux boot finished For Linux Login use:
   a. User Name: root
   b. Password: root
3. You can use Linux shell now.
   a. I2C 0 Bus type: i2cdetect -y -r 0
   b. I2C 1 Bus type: i2cdetect -y -r 1
   c. RTC check: dmesg | grep rtc
   d. ETH0 works with udhcpc
   e. USB: insert USB device
4. Option Features
   a. Webserver to get access to Zynq
      i. insert IP on web browser to start web interface
   b. init.sh scripts
      i. add init.sh script on SD, content will be load automatically on startup (template included in ./misc/SD)

Vivado HW Manager

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

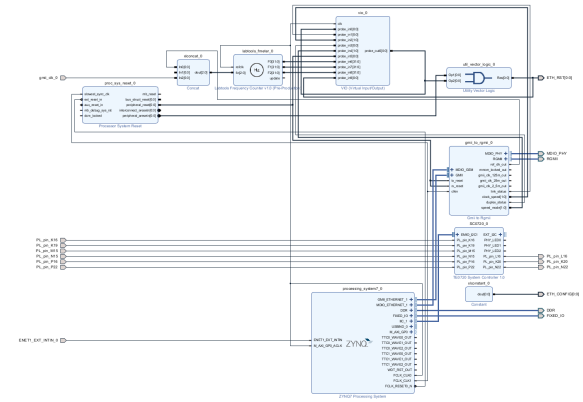
- Monitoring: PHY LED



Vivado Hardware Manager

System Design - Vivado

Block Design



Block Design

PS Interfaces

Type	Note
DDR	---
QSPI	MIO
ETH0	MIO
ETH1	EMIO
USB0	MIO
SD0	MIO
SD1	MIO
UART0	MIO
UART1	MIO
I2C0	MIO
I2C1	EMIO
GPIO	MIO
TTC0..1	EMIO
WDT	EMIO

Constrains

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]
```

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]
```

_i_eth1.xdc

```
############################################################
# 34_L13_N JB3:33 PHY:IN
set_property PACKAGE_PIN M20 [get_ports {ENET1_EXT_INTIN_0}]
set_property IOSTANDARD LVCMOS33 [get_ports ENET1_EXT_INTIN_0]
############################################################
# B34_L8_P JB3:38 RGMII_td[3]
set_property PACKAGE_PIN J21 [get_ports {RGMII_td[3]}]
# B34_L8_N JB3:40 RGMII_td[2]
set_property PACKAGE_PIN J22 [get_ports {RGMII_td[2]}]
# B34_L9_P JB3:42 RGMII_td[1]
set_property PACKAGE_PIN J20 [get_ports {RGMII_td[1]}]
# B34_L9_N JB3:44 RGMII_td[0]
set_property PACKAGE_PIN K21 [get_ports {RGMII_td[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_td*]
## B34_L22_P JB3:58 RGMII_rxc
#set_property PACKAGE_PIN R19 [get_ports RGMII_rxc]
#set_property IOSTANDARD LVCMOS33 [get_ports RGMII_rxc]
## B34_L22_N JB3:60 ETH_CONFIG
set_property PACKAGE_PIN T19 [get_ports {ETH_CONFIG[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports ETH_CONFIG*]
## B34_L12_N JB3:32 CLK_R_125
set_property PACKAGE_PIN L18 [get_ports {gmii_clk_0}]
set_property IOSTANDARD LVCMOS33 [get_ports gmii_clk_0]
#set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {gmii_clk_0_IBUF}]
############################################################
# B34_L13_P JB3:31 RGMII_rxc
set_property PACKAGE_PIN M19 [get_ports RGMII_rxc]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_rxc]
# B34_L21_P JB3:37 RGMII_rd[3]
set_property PACKAGE_PIN T16 [get_ports {RGMII_rd[3]}]
# B34_L21_N JB3:39 RGMII_rd[2]
```

```
set_property PACKAGE_PIN T17 [get_ports {RGMII_rd[2]}]
# B34_L15_P JB3:41 RGMII_rd[1]
set_property PACKAGE_PIN M21 [get_ports {RGMII_rd[1]}]
# B34_L15_N JB3:43 RGMII_rd[0]
set_property PACKAGE_PIN M22 [get_ports {RGMII_rd[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_rd*]
# B34_L17_P JB3:47 RGMII_rx_ctl
set_property PACKAGE_PIN R20 [get_ports RGMII_rx_ctl]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_rx_ctl]
# B34_L17_N JB3:49 MDIO_PHY_mdc
set_property PACKAGE_PIN R21 [get_ports MDIO_PHY_mdc]
set_property IOSTANDARD LVCMOS33 [get_ports MDIO_PHY_mdc]
# B34_L23_P JB3:51 mdio_phy_mdio_io
set_property PACKAGE_PIN R18 [get_ports MDIO_PHY_mdio_io]
set_property IOSTANDARD LVCMOS33 [get_ports MDIO_PHY_mdio_io]
# B34_L23_N JB3:53 ETH_RST
set_property PACKAGE_PIN T18 [get_ports {ETH_RST[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports ETH_RST*]
# B34_L14_P JB3:57 RGMII_txc
set_property PACKAGE_PIN N19 [get_ports RGMII_txc]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_txc]
# B34_L14_N JB3:59 RGMII_tx_ctl
set_property PACKAGE_PIN N20 [get_ports RGMII_tx_ctl]
set_property IOSTANDARD LVCMOS33 [get_ports RGMII_tx_ctl]
############################################################

#timing
#create_clock -period 8.000 -name RGMII_rxc -waveform {0.000 4.000}
[get_ports RGMII_rxc]
#set_clock_groups -logically_exclusive -group [get_clocks -
include_generated_clocks {*gmii_clk_25m_out *gmii_clk_2_5m_out}] -group
[get_clocks -include_generated_clocks *gmii_clk_125m_out]


#ignore vio
set_false_path -from [get_pins -hier -filter {name =~ */i_gmii_to_rgmii
/i_gmii_to_rgmii/link_status_reg/C}] -to [get_pins -hier -filter {name =~ *
/PROBE_IN_INST/probe_in_reg_reg*/D}]
set_false_path -from [get_pins -hier -filter {name =~ */i_gmii_to_rgmii
/i_gmii_to_rgmii/clock_speed_reg[*]/C}] -to [get_pins -hier -filter {name
=~ */PROBE_IN_INST/probe_in_reg_reg*/D}]
set_false_path -from [get_pins -hier -filter {name =~ */i_gmii_to_rgmii
/i_gmii_to_rgmii/duplex_status_reg/C}] -to [get_pins -hier -filter {name
=~ */PROBE_IN_INST/probe_in_reg_reg*/D}]


## Clock Period Constraints
#create_clock -period 5.000 -name clkin -add [get_nets clkin]
create_clock -period 8.000 -name rgmii_rxc -add [get_ports RGMII_rxc]

## Clock constraint if parameter C_EXTERNAL_CLOCK = 1
#create_clock -add -name gmii_clk  -period 8.000 [get_ports gmii_clk]
## Clock constraint if parameter C_EXTERNAL_CLOCK = 1 and clock skew on
TXC is through MMCM
#create_clock -add -name gmii_clk_90  -period 8.000 -waveform {2 6}
[get_ports gmii_clk_90]

##False path constraints to async inputs coming directly to synchronizer
set_false_path -to [get_pins -hier -filter {name =~ *idelayctrl_reset_gen
/*reset_sync*/PRE }]
set_false_path -to [get_pins -of [get_cells -hier -filter { name =~
```

```
*i_MANAGEMENT/SYNC_*/data_sync* }] -filter { name =~ *D }]
set_false_path -to [get_pins -hier -filter {name =~ *reset_sync*/PRE }]

##False path constraints from Control Register outputs
set_false_path -from [get_pins -hier -filter {name =~ *i_MANAGEMENT
/DUPLEX_MODE_REG*/C }]
set_false_path -from [get_pins -hier -filter {name =~ *i_MANAGEMENT
/SPEED_SELECTION_REG*/C }]

## constraint valid if parameter C_EXTERNAL_CLOCK = 0
set_case_analysis 0 [get_pins -hier -filter {name =~ *i_bufgmux_gmii_clk
/CE0}]
set_case_analysis 0 [get_pins -hier -filter {name =~ *i_bufgmux_gmii_clk
/S0}]
set_case_analysis 1 [get_pins -hier -filter {name =~ *i_bufgmux_gmii_clk
/CE1}]
set_case_analysis 1 [get_pins -hier -filter {name =~ *i_bufgmux_gmii_clk
/S1}]

## constraint valid if parameter C_EXTERNAL_CLOCK = 0 and clock skew on
TXC is through MMCM
#set_case_analysis 0 [get_pins -hier -filter {name =~
*i_bufgmux_gmii_90_clk/CE0}]
#set_case_analysis 0 [get_pins -hier -filter {name =~
*i_bufgmux_gmii_90_clk/S0}]
#set_case_analysis 1 [get_pins -hier -filter {name =~
*i_bufgmux_gmii_90_clk/CE1}]
#set_case_analysis 1 [get_pins -hier -filter {name =~
*i_bufgmux_gmii_90_clk/S1}]

#...

set_property slew FAST [get_ports [list {RGMII_td[3]} {RGMII_td[2]}
{RGMII_td[1]} {RGMII_td[0]} RGMII_txc RGMII_tx_ctl]]
```

Software Design - Vitis

For SDK project creation, follow instructions from:

Vitis

Application

Template location: ./sw_lib/sw_apps/

zynq_fsbl

TE modified 2019.2 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)\n\
- 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

zynq_fsbl_flash

TE modified 2019.2 FSBL

General:

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

hello_te0720

Hello World App in Endless loop.

u-boot

U-Boot.elf is generated with PetaLinux. SDK/HSI is used to generate Boot.bin.

Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart

Config

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

Changes:

- CONFIG_SUBSYSTEM_SERIAL_PS7_UART_0_SELECT=y
- CONFIG_SUBSYSTEM_SERIAL_IP_NAME="ps7_uart_0"
- CONFIG_SUBSYSTEM_NETBOOT_OFFSET=0x8000000 ! Must be done manually for 256MB DDR only not done on with HDF import from the template!

U-Boot

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

Changes:

- CONFIG_ENV_IS_NOWHERE=y

- # CONFIG_ENV_IS_IN_SPI_FLASH is not set

Change platform-top.h:

```
#include <configs/platform-auto.h>
#define CONFIG_SYS_BOOTM_LEN 0xF000000
#define DFU_ALT_INFO_RAM \
                "dfu_ram_info=" \
        "setenv dfu_alt_info " \
        "image.ub ram $netstart 0x1e00000\0" \
        "dfu_ram=run dfu_ram_info && dfu 0 ram 0\0" \
        "thor_ram=run dfu_ram_info && thordown 0 ram 0\0"

#define DFU_ALT_INFO_MMC \
        "dfu_mmc_info=" \
        "set dfu_alt_info " \
        "${kernel_image} fat 0 1\\\\;" \
        "dfu_mmc=run dfu_mmc_info && dfu 0 mmc 0\0" \
        "thor_mmc=run dfu_mmc_info && thordown 0 mmc 0\0"


/*Required for uartless designs */
#ifndef CONFIG_BAUDRATE
#define CONFIG_BAUDRATE 115200
#ifdef CONFIG_DEBUG_UART
#undef CONFIG_DEBUG_UART
#endif
#endif

/*Dependencies for ENV to be stored in EEPROM. Ensure environment fits in
eeprom size*/
#ifdef CONFIG_ENV_IS_IN_EEPROM
#define CONFIG_SYS_I2C_EEPROM_ADDR_LEN         1
#define CONFIG_SYS_I2C_EEPROM_ADDR             0x54
#define CONFIG_SYS_EEPROM_PAGE_WRITE_BITS      4
#define CONFIG_SYS_EEPROM_PAGE_WRITE_DELAY_MS  5
#define CONFIG_SYS_EEPROM_SIZE                 1024 /* Bytes */
#define CONFIG_SYS_I2C_MUX_ADDR                0x74
#define CONFIG_SYS_I2C_MUX_EEPROM_SEL          0x4
#endif


#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"
```

Device Tree

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


/* default */
```

```
/* QSPI PHY */
&qspi {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";
    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>;
        };
    };
};
/* ETH PHY from TE0706*/
&gem1 {
  //  local-mac-address = [00 0a 35 00 db b2];
        phy-handle = <&phy1>;
    mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        phy1: phy@1 {
            compatible = "marvell,88e1510";
            device_type = "ethernet-phy";
            reg = <1>;
            //marvell,reg-init = <0x3 0x10 0x0000 0x0501 0x3 0x11 0x0000
0x4415>;
            marvell,reg-init = <0x12 0x14 0x0200>;
        };
    };
};
/* 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>;
    };
};
```

Kernel

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

Changes:

- CONFIG_RTC_DRV_ISL12022=y
- CONFIG_XILINX_GMII2RGMII=y

Core

Add manually

Changes:

- init-ifupdown eth interface configuration

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)

Applications

startup

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

See: \os\petalinux\project-spec\meta-user\recipes-apps\startup\files

webfwu

Webserver application accemble for Zynq access. Need busybox-httpd

Additional Software

No additional software is needed.

Appx. A: Change History and Legal Notices

Document Change History

To get content of older revision got to "Change History" of this page and select older document revision number.

Date	Document Revision	Authors	Description
Error renderi ng macro 'page-info' Ambiguo us method overload ing for method jdk. proxy24 1.$Proxy 3496#ha sConten tLevelPe	**Error renderi ng macro 'page-info'** Ambiguo us method overload ing for method jdk. proxy24 1.$Proxy 3496#ha sConten tLevelPe	**Error renderi ng macro 'page-info'** Ambiguo us method overload ing for method jdk. proxy24 1.$Proxy 3496#ha sConten tLevelPe	• Note to "Known Issues"

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

rmission . Cannot resolve which method to invoke for [null, class java. lang. String, class com. atlassian . confluen ce. pages. Page] due to overlapp ing prototyp es between : [interfac e com. atlassian . confluen ce.user. Conflue nceUser , class java. lang.

String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	String, class com. atlassian . confluen ce.core. Content EntityOb ject] [interfac e com. atlassian .user. User, class java. lang. String, class com. atlassian . confluen ce.core. Content EntityOb ject]	
2021-04-20	v.6	John Hartfiel	• update design files
2020-12-11	v.5	John Hartfiel	• 2019.2 release
--	All	**Error renderi ng**	--

macro 'page-info'

Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping

prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluen

		ce.core. Content EntityOb ject]	

Document change history.

Legal Notices

Data Privacy

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

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA).

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]