TE0720 SDSoC

Table of contents

Overview

Basic project to generate SDSoC Platform project and petalinux. Various SDSoC examples are included.

Refer to http://trenz.org/te0xyz-info for the current online version of this manual and other available documentation.

Key Features

- SDSoC platform export
- SDSoC examples (baremetal and linux)

Revision History

Date	Vivado	Project Built	Authors	Description
2018-11-06	2018.2	TE0720-TE0720_zsys_SDSoC-vivado_2018.2-build_03_20181106093337.zip	Zdenek Pohl, Jiri Kadlec	initial release

Design Revision History

Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version
No known issues	---	---	---

Known Issues

Requirements

Software

Software	Version	Note
SDx(SDSoC)	2018.2	needed
PetaLinux	2018.2	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	Others	Notes
te0720-03-2if	2if	REV02, REV03	1GB	32		
te0720-03-2ifc3	2if	REV02, REV03	1GB	32	2.5 mm connector	
te0720-03-2ifc8	2if	REV02, REV03	1GB	32	32GB eMMC	
te0720-03-1qf	1qf	REV02, REV03	1GB	32		
te0720-03-1qfa	1qf	REV03	1GB	32		Micron instead of Spansion Flash
te0720-03-1cf	1cf	REV02, REV03	1GB	32		
te0720-03-1cfa	1cf	REV02, REV03	1GB	32	8GB eMMC	
te0720-03-2ef	2ef	REV02, REV03	1GB	32		
te0720-03-14s-1c	14s	REV02, REV03	1GB (L)	32		
te0720-03-2ifa	2if	REV03	1GB	32		Micron instead of Spansion Flash

Hardware Modules

Design supports following carriers:

Carrier Model	Notes

Hardware Carrier

Additional HW Requirements:

Additional Hardware	Notes

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
SDK/HSI	<design name>/sw_lib	Additional Software Template for SDK/HSI and apps_list.csv with settings for HSI
PetaLinux	<design name>/os/petalinux	PetaLinux template with current configuration
SDSoC	<design name>/../SDSoC_PFM	SDSoC Platform will be generated by TE Scripts or as separate download

Design sources

Additional Sources

Type	Location	Notes

Additional design sources

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.

Basic Reference Design and SDSoC Platform projects for Win OS are available on:

- TE0720 "SDSoC" Design

Create SDSoC Platform Project

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

See also:

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

This cheaper describes how you can create SDSoC platform project from special Trenz Electronik Vivado projects.

> This chapter is optional: Prebuilt SDSoC platform projects for Win OS are also available on the download area.

Vivado SDSoc Platform export

1. Download and unpack package (use short path or 'subst' command to make path as short as possible).
2. Run '*_create_win_setup.cmd*' script and choose option '1'
3. Edit file '*design_basic_settings.cmd*' and fill correctly all variables, set ENABLE_SDSOC=1.
4. Launch Vivado by command '*vivado_create_project_guimode.cmd*'
5. In Vivado:
    a. run script in TCL console
    TE::hw_build_design -export_prebuilt
6. (Optional) Build Petalinux:
    a. Copy OS folder from downloaded package to Linux PC.
    b. Copy hdf file located in prebuilt/hardware/<shortname> subfolder to Linux PC
    c. Switch to Linux PC and run in terminal:
        i. Initialize Petalinux SDK:
        source <petalinux_SDK_install_path>/settings.sh
        ii. Use hdf file to configure linux project:
        petalinux-config -p <path_to_petalinux_project> --get-hw-description
        iii. Build petalinux image from within the project:
        petalinux-build
        iv. Repeat previous step until images are sucessfully created. It may take from 1 to 5 attempts.
        v. Find linux images and copy them back to the Trenz package
        Files to be copied are located in <petalinux_project_path>/images/linux. Filenames are:
        image.ub, u-boot.elf, bl31.elf
        Target folder in Trenz package is:
        prebuilt/os/petalinux/default for all 1GB DDR modules.
        or
        prebuilt/os/petalinux/<shortname> for both supported 2GB DDR modules
        vi. Return back to Vivado
7. In Vivado:
    a. run script in TCL console
    TE::sw_run_hsi
    b. run script in TCL console
    TE::ADV::beta_util_sdsoc_project
8. Custom platform for SDx tool is now exported to SDSoC_PFM folder sitting next to Trenz package folder

SDK/HSI Application

Source location: \sw_lib\sw_apps

zynqmp_fsbl

TE modified 2018.2 FSBL

Changes:

- Si5338 Configuration, ETH+OTG Reset over GPIO
    - see xfsbl_board.c, xfsbl_board.h, xfsbl_main.c
    - Add register_map.h, si5338.c, si5338.h

Note: Remove compiler flags "-Os -flto -ffat-lto-objects" on 2018.2 SDK to generate FSBL

zynqmp_fsbl_flash

Changes:

- Set FSBL Boot Mode to JTAG
- Disable Memory initialisation
- see xfsbl_initialisation.c, xfsbl_hw.h, xfsbl_handoff.c, xfsbl_main.c

Note: Remove compiler flags "-Os -flto -ffat-lto-objects" on 2018.2 SDK to generate FSBL

Petalinux Configuration

UConfig

U-boot

Device Tree

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

Kernel

Rootfs

Application

Use SDSoC Platform Project

Included Example

Example	Comment
Array partition	This example shows how to use array partitioning to improve performance of a hardware function
Burst rw	This is simple example of using AXI4-master interface for burst read and write
Custom data type	This is a simple example of RGB to HSV conversion to demonstrate Custom Data Type usage in hardware accelerator. Xilinx HLS compiler supports custom data type to operate within the hardware function and also it acts as a memory interface between PL to DDR

Data access random	This is a simple example of matrix multiplication (Row x Col) to demonstrate random data access pattern
Dependence inter	This is a simple example to demonstrate inter dependence attribute. Using inter dependence attribute user can provide additional dependency details to compiler which allow compiler to perform unrolling/pipelining to get better performance.
Direct connect	This is a simple example of matrix multiplication with matrix addition (Out = (A x B) + C) to demonstrate direct connection which helps to achieve increasing in system parallelism and concurrency
Dma sg	This example demonstrates how to use Scatter-Gather DMAs for data transfer to/from hardware accelerator
Dma simple	This example demonstrates how to insert Simple DMAs for data transfer between User program and hardware accelerator
File IO Dense Optical Flow	Linux video processing application that reads input video from a file and writes out the output video to a file. Video processing performs LK Dense Optical Flow over two Full HD frames video file. You can run it by supplying a 1080p YUV422 file route85_1920x1080.yuv as input.
File IO Stereo Block Matching	Linux video processing application that reads input video from a file and writes out the output video to a file. Video processing performs Stereo Block Matching to calculate depth in a single sample stereo video file desk_1280x720.yuv as input.
File IO Video Processing	Linux video processing application that reads input video from a file and writes out the output video to a file. Video processing includes Motion Adaptive Noise Reduction (MANR) followed by a Sobel filter for edge detection. You can run it by supplying a 1080p YUV422 file as input with limiting number of frames to a maximum of 20 frames.
Full array 2d	This is a simple example of accessing full data from 2D array
Hello vadd	This is a basic hello world kind of example which demonstrates how to achieve vector addition using hardware function
Lmem 2rw	This is a simple example of vector addition to demonstrate how to utilize both ports of Local Memory
Loop fusion	This example will demonstrate how to fuse two loops into one to improve the performance of a C/C++ hardware function.
Loop perfect	This nearest neighbor example is to demonstrate how to achieve better performance using perfect loop.
Loop pipeline	This example demonstrates how loop pipelining can be used to improve the performance of a hardware function.
Loop reorder	This is a simple example of matrix multiplication (Row x Col) to demonstrate how to achieve better pipeline II factor by loop reordering.
Row array 2D	This is a simple example of accessing each row of data from 2D array
Shift register	This example demonstrates how to shift values in each clock cycle
Sys port	This is a simple example which demonstrates sys_port usage
Systolic array	Matrix multiplication implemented as systolic array
Wide memory rw	Wide memory read write 128 bit wide
Window array 2d	This is a simple example of accessing window of data from 2D array

Table 8: List of sample applications

Create Example

1. Start SDx 2018.2
2. Select Workspace folder, short path is recommended. Use '*subst*' command if needed.
   Click "Create SDx Project"
   a. Choose 'Application' project type
   b. Set Project Name (example: mmult)
   c. Set Platform:
      i. Click 'Add Custom Platform ...'
      ii. Add SDSoC_PFM folder
      iii. Click next
   d. Select 'System configuration': Linux
   e. Click "Next"
   f. Select Template Application, example:'Array partition'
   g. Click "Finish"
   h. Right click project <project name>, example:'mmult' in the Project Explorer window and choose 'Build'
   i. The SDSoC project is compiled (ca 30 min) hw accelerated matrix multiplication.
      SDCard image is created

Launch

1. Copy created files to the SD card.
2. Connect serial terminal via the USB cable.
3. Power ON carrier.
4. On PC, open serial terminal.
5. Reset carrier.
6. Boot of Linux starts up to login stage. Login as 'root' with password 'root'.
   Use serial terminal:
   a. cd to /run/media/mmcblk0p1
   b. execute mmult.elf or other example application
   c. Observe the result of accelerated matrix multiplication example.

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 rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy241.$Proxy3496#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy241.$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]	**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]	• change list

--	all	**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]	--

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]