

Application Note



Xilinx Vitis Al *facedetect* Demo on Trenz Electronic TE0820-4EV SoM with TE0701-06 Carrier Board and Avnet HDMI In/Out FMC Card

Lukáš Kohout, Zdeněk Pohl and Jiří Kadlec <u>kohoutl@utia.cas.cz</u>, <u>zdenek.pohl@utia.cas.cz</u> and <u>kadlec@utia.cas.cz</u>

Revision history

Rev.	Date	Author	Description
0	07.02.2023	L.Kohout	Document creation
1			
2			

Contents

1	Introduction	. 1
2	Requirements	. 1 . 1 . 1
3	Create Extensible Vitis Platform	1 4 4 10
	Facadatast Vitia Al Dama	11
4		14
4 5	VCU Decoder	20
4 5 6	VCU Decoder Automations and Optimizations 6.1 QoS 6.2 Monitor Optimizations 6.3 DPU Firmware	20 22 22 22 22
4 5 6 7	VCU Decoder Automations and Optimizations	20 22 22 22 22 22 22 22
4 5 6 7 8	VCU Decoder Automations and Optimizations	20 22 22 22 22 22 22 22 22 22 22 22 22 2
4 5 6 7 8 9	VCU Decoder Automations and Optimizations	20 22 22 22 22 22 22 22 22

Acknowledgement

Acknowledgement to the StorAlge project and the corresponding Czech institutional support project No. 8A21009.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007321. The JU receives support from the European Union's Horizon 2020 research and innovation program and France, Belgium, Czech Republic, Germany, Italy, Sweden, Switzerland, Turkey.



1 Introduction

This document provides a tutorial describing setup and run Vitis AI 2.0 *facedetect* demonstrator on Trenz TE0820-4EV SoM [1] with Trenz TE0701-06 carrier board [2] and Avnet HDMI In/Out FMC card [3]. The system uses a Xilinx DPU unit to accelerate calculations. An input video data are taken from the HDMI input (FMC), processed data are displayed via HDMI output (FMC). The system desktop is shown via HDMI output on TE0701 carrier board. As the TE0820-4EV provides H.264/H.265 Video Codec Unit (VCU), the demonstrator is configured to use the decoder of the VCU.

2 Requirements

2.1 Hardware

- Trenz Electronic TE0820-4EV SoM with Xilinx Zynq UltraScale+ ZU4EV-1land 2GB DDR4 [1].
- Trenz Electronic TE0701-06 carrier board [2].
- AVNET HDMI In/Out FMC card [3] (AES-FMC-HDMI-CAM-G).
- HDMI camera, tested device is LAMAX X10 TAURUS. This camera provides HDMI output in 1920x1080 resolution at 60 frames per second. The camera can be replaced by the output from the PC.
- HDMI monitor capable to display 1920x1080 resolution at 60 frames per second.

2.2 Software

• This tutorial is an extension of the *TE0820 test board Vitis AI Tutorial* [4] <u>https://wiki.trenz-electronic.de/display/PD/TE0820+test+board+Vitis+AI+Tutorial</u>. Before following this guide, it is required to go through the *TE0820 test board Vitis AI Tutorial* from the beginning up to step 3.1.1 *Create Extensible Vitis platform*. As many of the next steps are slightly different from the original tutorial, this document will describe all of them again. In Section select the *Fast Track* branch.

3 Create Extensible Vitis Platform

This section continues with the description of the extensible Vitis platform creation. It is assumed that the Vivado project has been already created and it is opened now. The working directory is ~/work/te0820_15_240/test_board.

3.1 Build HW

To create a HW of the extensible Vitis platform follow the steps bellow.

1. Add HDMI IP cores to the Vivado project. The IPs are the content of the package attached with this document. Copy content of the folder

ip_lib to ~/work/te0820_15_240/test_board/ip_lib

The *ip_lib* folder should contain these IP cores (folders):

axis_vid_det_1_0
hblank_det_1_0
im_hdmi_in
im_hdmi_out
labtools_fmeter
video io to hdmi



https://sp.utia.cas.cz

NOTE: The project IP catalog will be updated automatically in the next step. Otherwise, it can be updated by the user with command "update_ip_catalog -rebuild" executed from the Vivado TCL console.

Block Design of the Vivado project must be opened for this step. The package attached 2. with this application note contains a TCL file containing all commands to create a Vivado block design of the extensible Vitis platform. To create the platform copy the file

vivado/te0820_EV_fast_track_vcu_vdma.tcl

into the folder

ŪTİA

~/work/te0820_15_240/test_board/vivado

and execute from the Vivado TCL console

source te0820 EV fast track vcu vdma.tcl

Expected block design is shown in figure bellow. To get an overview of the generated extensible platform, explore the *Platform Setup* tab.



In Diagram window, validate design by clicking on Validate Design icon. Received 3. Critical Messages window indicates that input intr[0:0] of axi intc 0 is not connected. This is expected. The Vitis extensible design flow will connect this input to interrupt outputs from generated HW IPs. Kick OK.



4. Compile the created block design with Trenz scripts. In Vivado Tcl Console, type following script and execute it. It will take some time to compile HW.

TE::hw_build_design -export_prebuilt

The resulting XSA file is created in the location:

~/work/te0820_15_240/test_board/vivado/test_board_4ev_1e_2gb.xsa

5. Compile custom SW with Trenz scripts (FSBL,). In Vivado Tcl Console, type the following script and execute it by Enter. It will take some time to compile.

TE::sw_run_vitis -all

After the script controlling SW compilation is finished, the Vitis SDK GUI is opened. Close the Vitis *Welcome* page. Compile the two included SW projects. Standalone custom Vitis platform **TE0820-03-04EV-1EA** has been created and compiled.



The **TE0820-03-04EV-1EA** Vitis platform includes Trenz Electronic custom first stage boot loader. It includes SW extension specific for the Trenz SoM initialization. This custom *zynqmp_fsbl* project has been compiled into executable file *fsbl.elf*. It is located in folder:

~/work/te0820_15_240/test_board/prebuilt/software/4ev_1e_2gb/fsbl.elf

- 6. Exit the opened Vitis SDK project.
- 7. Exit Vivado.



https://sp.utia.cas.cz

All disclosure and/or

3.2 Get Vitis Al Libraries

 Download the Vitis-AI 2.0 repository. In web browser, open page: <u>https://github.com/Xilinx/Vitis-AI/tree/2.0</u>

Click on green Code button and download Vitis-Al-2.0.zip file.

2. Unzip

Vitis-Al-2.0.zip

file to directory

~/vitis_ai_2_0.

3.3 Build OS and SDK

To configure the default Trenz Petalinux for the Vitis extensible platform follow the steps described in the list below.

1. Go to the Petalinux working directory:

cd ~/work/te0820_15_240/test_board/os/petalinux

2. Set the path to Petalinux tool:

source ~/petalinux/2021.2/settings.sh

3. Configure Petalinux with the current HW specification, which is part of the file test_board_4ev_1e_2gb.xsa.

petalinux-config --get-hw-description=../../vivado

Select Exit and then Yes to close this window.

4. Customize main device-tree of the Petalinux project. Open file

~/work/te0820_15_240/test_board/os/petalinux/project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi

and append these sections into the file

```
&amba {
 zyxclmm drm {
   compatible = "xlnx, zocl";
   status = "okay";
   reg = <0x0 0xA0000000 0x0 0x10000>;
    interrupt-parent = <&axi_intc_0>;
    interrupts = <0 4>, <1 4>, <2 4>, <3 4>,
                <4 4>, <5 4>, <6 4>, <7 4>,
                <8 4>, <9 4>, <10 4>, <11 4>,
                <12 4>, <13 4>, <14 4>, <15 4>,
                <16 4>, <17 4>, <18 4>, <19 4>,
                <20 4>, <21 4>, <22 4>, <23 4>,
                <24 4>, <25 4>, <26 4>, <27 4>,
                <28 4>, <29 4>, <30 4>, <31 4>;
 };
};
/ {
 reserved-memory {
   #address-cells = <2>;
   #size-cells = <2>;
   ranges;
   reserved1: buffer@0x28000000 {
     no-map;
    reg = <0x0 0x28000000 0x0 (1920 * 1080 * 4)>;
```

signal processing

4/25



ŪTĬA

```
};
      };
      framebuffer1: fbxserver {
                                      // HDMI out
         #address-cells = <2>;
         #size-cells = <2>;
        compatible = "simple-framebuffer";
        reg = <0x0 0x28000000 0x0 (1920 * 1080 * 4)>; // 1080p
        width = <1920>;
        height = <1080>;
        stride = <(1920 * 4)>;
        format = "a8b8g8r8";
      };
    };
    /delete-node/ &hdmi in v proc ss 0;
    /delete-node/ &hdmi out v proc ss 0;
    /delete-node/ &axi vdma 0;
    /delete-node/ &axi vdma 1;
    &amba {
      hdmi in v proc ss 0: v proc ss 0@80000000 {
        clock-names = "aclk";
        clocks = <&misc clk 1>;
        compatible = "xlnx,v-proc-ss-2.3", "xlnx,vpss-csc", "xlnx,v-vpss-csc";
        reg = <0x0 0x8000000 0x0 0x10000>;
        xlnx, colorspace-support = <1>;
        xlnx,csc-enable-window = "false";
        xlnx,max-height = <2160>;
        xlnx,max-width = <3840>;
        xlnx,num-video-components = <3>;
        xlnx, samples-per-clk = <1>;
        xlnx, topology = <3>;
        xlnx,use-uram = <0>;
        xlnx,video-width = <8>;
      };
      hdmi_out_v_proc_ss_0: v_proc_ss_1080010000 {
        clock-names = "aclk";
        clocks = <&misc clk 1>;
        compatible = "xlnx, v-proc-ss-2.3", "xlnx, vpss-csc", "xlnx, v-vpss-csc";
        reg = <0x0 0x80010000 0x0 0x10000>;
        xlnx, colorspace-support = <1>;
         xlnx,csc-enable-window = "false";
        xlnx,max-height = <2160>;
        xlnx,max-width = <3840>;
        xlnx,num-video-components = <3>;
        xlnx, samples-per-clk = <1>;
        xlnx, topology = \langle 3 \rangle;
        xlnx,use-uram = <0>;
        xlnx, video-width = <8>;
      };
       /* VDMA TE0701 HDMIO*/
      axi vdma 0: vdma 0@80050000 {
        #dma-cells = <1>;
        clock-names = "s axi lite aclk", "m axi mm2s aclk", "m axis mm2s aclk";
        clocks = <&misc_clk_0>, <&misc_clk_1>, <&misc_clk_1>;
        compatible = "xlnx, axi-vdma-6.3", "xlnx, axi-vdma-1.00.a";
         interrupt-names = "mm2s introut";
        interrupt-parent = <&gic>;
        interrupts = <0 90 4>;
         reg = <0x0 0x80050000 0x0 0x10000>;
         xlnx,addrwidth = <0x20>;
         xlnx,flush-fsync = <0x1>;
        xlnx,num-fstores = <0x1>;
        dma-channel@80050000 {
signal processing
```

```
5/25
```



```
compatible = "xlnx,axi-vdma-mm2s-channel";
      interrupts = <0 90 4>;
     xlnx,datawidth = <0x20>;
     xlnx, device-id = <0x0>;
   };
 };
  /* VDMA IMAGEON HDMI IN/OUT */
 axi vdma 1: vdma 1080060000 {
    #dma-cells = <1>;
   clocks = <&misc_clk_0>, <&misc_clk_1>, <&misc_clk_1>,
            <&misc_clk_1>, <&misc_clk_1>;
   compatible = "xlnx,axi-vdma-6.3", "xlnx,axi-vdma-1.00.a";
interrupt-names = "mm2s_introut", "s2mm_introut";
   interrupt-parent = <&gic>;
   interrupts = <0 92 4 0 93 4>;
   reg = <0x0 0x80060000 0x0 0x10000>;
    xlnx,addrwidth = <0x20>;
    xlnx,flush-fsync = <0x1>;
   xlnx,num-fstores = <0x7>;
   dma-channel@80060000 {
     compatible = "xlnx,axi-vdma-mm2s-channel";
     interrupts = <0 92 4>;
     xlnx, datawidth = <0x18>;
     xlnx,device-id = <0x1>;
     xlnx,genlock-mode ;
   };
   dma-channel@80060030 {
     compatible = "xlnx,axi-vdma-s2mm-channel";
      interrupts = <0 93 4>;
     xlnx, datawidth = <0x18>;
     xlnx, device-id = <0x1>;
     xlnx,genlock-mode ;
   };
 };
};
&clk_wiz_2 {
   compatible = "generic-uio";
   status = "okay";
};
&axi vdma 0 {
   compatible = "generic-uio";
   status = "okay";
};
&axi vdma 1 {
   compatible = "generic-uio";
   status = "okay";
};
&hdmi out_v_tc_0 {
   compatible = "generic-uio";
   status = "okay";
};
&hdmi_in_v_proc_ss_0 {
   compatible = "generic-uio";
   status = "okay";
};
```



```
6/25
```

https://sp.utia.cas.cz



```
&hdmi_out_v_proc_ss_0 {
    compatible = "generic-uio";
   status = "okay";
};
&hdmi in axis vid det 0 {
   compatible = "generic-uio";
   status = "okay";
};
&hdmi in hblank_det_0 {
   compatible = "generic-uio";
   status = "okay";
};
&gpio {
   emio-gpio-width = <1>;
   status = "okay";
};
```

5. Customize device-tree of the Petalinux project for u-boot. Open file

~/work/te0820_15_240/test_board/os/petalinux/project-spec/meta-user/recipes-bsp/uboot-device-tree/files/system-user.dtsi

and append these sections into the file

/delete-node/ &hdmi_in_v_proc_ss_0; /delete-node/ &hdmi_out_v_proc_ss_0;

6. Add Vitis-AI 2.0 library recipes to the Petalinux project. Copy

~/vitis_ai_2_0/tools/Vitis-AI-Recipes/recipes-vitis-ai

to

~/work/te0820_15_240/test_board/os/petalinux/project-spec/meta-user/

- 7. Configure Petalinux rootfs.
 - a. Append user defined rootfs packages to the petalinux menuconfig. To the file

~/work/te0820_15_240/test_board/os/petalinux/project-spec/metauser/conf/user-rootfsconfig

append these lines:

```
CONFIG xrt
CONFIG xrt-dev
CONFIG zocl
CONFIG_opencl-clhpp-dev
CONFIG_opencl-headers-dev
CONFIG packagegroup-petalinux-opencv
CONFIG_packagegroup-petalinux-opencv-dev
CONFIG dnf
CONFIG e2fsprogs-resize2fs
CONFIG_parted
CONFIG resize-part
CONFIG packagegroup-petalinux-vitisai
CONFIG packagegroup-petalinux-self-hosted
CONFIG cmake
CONFIG_packagegroup-petalinux-vitisai-dev
CONFIG mesa-megadriver
CONFIG packagegroup-petalinux-x11
CONFIG_packagegroup-petalinux-v4lutils
CONFIG_packagegroup-petalinux-matchbox
CONFIG_vitis-ai-library
CONFIG_vitis-ai-library-dev
```



7/25

https://sp.utia.cas.cz



```
CONFIG_vitis-ai-library-dbg
CONFIG_packagegroup-petalinux-gstreamer
CONFIG_libomxil
CONFIG_packagegroup-core-ssh-dropbear
CONFIG_imagefeature-ssh-server-dropbear
CONFIG_imagefeature-ssh-server-openssh
CONFIG_openssh
CONFIG_openssh-sftp-server
CONFIG_openssh-sshd
CONFIG_openssh-scp
CONFIG_imagefeature-package-management
```

b. Open the petalinux rootfs configuration window, execute from the terminal:

petalinux-config -c rootfs

c. Go to the **user packages** and deselect packages **imagefeature-ssh-serverdropbear** and **packagegroup-core-ssh-dropbear**. Enable all other packages.

```
[*] libomxil
[*] cmake (NEW)
[*] dnf
[*] e2fsprogs-resize2fs
[*] imagefeature-package-management
[ ] imagefeature-ssh-server-dropbear
[*] imagefeature-ssh-server-openssh
[*] mesa-megadriver
[*] opencl-clhpp-dev
[*] opencl-headers-dev (NEW)
[*] openssh
[*] openssh-scp
[*] openssh-sftp-server
[*] openssh-sshd
[ ] packagegroup-core-ssh-dropbear
[*] packagegroup-petalinux-gstreamer
[*] packagegroup-petalinux-matchbox
[*] packagegroup-petalinux-opencv
[*] packagegroup-petalinux-opencv-dev
[*] packagegroup-petalinux-self-hosted
[*] packagegroup-petalinux-v4lutils
[*] packagegroup-petalinux-vitisai (NEW)
[*] packagegroup-petalinux-vitisai-dev (NEW)
[*] packagegroup-petalinux-x11
[*] parted
[*] resize-part
[*] vitis-ai-library (NEW)
[*] vitis-ai-library-dbg (NEW)
[*] vitis-ai-library-dev (NEW)
[*] xrt
[*] xrt-dev
[*] zocl
```

- d. Exit twice and select Yes to save changes
- 8. Configure Petalinux kernel.
 - a. Launch kernel config:

petalinux-config -c kernel

 Disable CPU IDLE. Ensure the following items are TURNED OFF by entering 'n' in the [] menuconfig selection:

```
CPU Power Management --->

CPU Idle --->

[] CPU idle PM support

CPU Frequency scaling --->

[] CPU Frequency scaling
```

department of signal processing

8/25

https://sp.utia.cas.cz



NOTE: When the processor is not in use it is switched to CPU IDLE (WFI). When JTAG is connected, the hardware server on host machine talks to the processor regularly. If it talks to a processor in IDLE state, the system will hang because of incomplete AXI transactions. So, it is recommended to disable the CPU IDLE feature during project development phase. It can be re-enabled after the design has completed to save power in final products.

c. Enable simple framebuffer to support graphical output via HDMI on the TE0701.

```
Device Drivers --->
Graphics support --->
Frame buffer Devices --->
-*- Support for frame buffer devices --->
[*] Simple framebuffer support
```

- d. Exit kernel configuration window, select Yes to save changes.
- 9. Configure common Petalinux settings.
 - a. Launch Petalinux menuconfig window.

petalinux-config

b. Set Petalinux to use an extra EXT4 partition on the SD card for the file system. Switch the root file system from INITRD to **EXT4**:

```
Image Packaging Configuration --->
Root filesystem type (INITRD) --->
(X) EXT4 (SD/eMMC/SATA/USB)
```

Go back to

Image Packaging Configuration --->

level and change the Device node of SD device from the default value

/dev/mmcblk0p2

to new value (use the second partition):

/dev/mmcblk1p2

c. Set Petalinux to build only the EXT4 file as the root file system image. The original setting may cause the Petelinux build to fail. Go to level

Image Packaging Configuration --->

and modify the Root filesystem formats string from

cpio cpio.gz cpio.gz.u-boot ext4 tar.gz jffs2

to

ext4

d. Set the Petalinx boot arguments to use the EXT4 partition, pre-allocate 512 MB for CMA, and enable the generic UIO driver. Go to:

9/25

```
DTG Settings --->
Kernel Bootargs --->
```

Disable

generate boot args automatically

option, and set

user set kernel bootargs



Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.

string to

earlycon console=ttyPS0,115200 clk_ignore_unused root=/dev/mmcblk1p2 rw rootwait cma=512M uio_pdrv_genirq.of_id=generic-uio

- e. Exit this menuconfig window, select Yes to save changes.
- 10. Build the Petalinux image:

petalinux-build

The image files will be generated in the directory:

~/work/te0820_15_240/test_board/os/petalinux/images/linux

Compilation Petalinux takes some time, requires an internet connection and enough free disk space.

11. Create Petalinux SDK. It is a copy of the Petalinux root file system that is used by the Vitis tool to cross compile applications for newly created platfom. In terminal, execute:

petalinux-build --sdk

The generated sysroot package sdk.sh will be located in directory

~/work/te0820_15_240/test_board/os/petalinux/images/linux

Generation of the SDK package takes some time and requires enough free disk space.

3.4 Build Platform

To build the final extensible Vitis platform follow steps bellow:

1. Create the main platform folder te0820_15_240_pfm and the required subfolders:

mkdir -p ~/work/te0820_15_240/test_board_pfm/pfm/boot
mkdir -p ~/work/te0820_15_240/test_board_pfm/pfm/sd_dir

2. Go to the platform directory

cd ~/work/te0820_15_240/test_board_pfm

- 3. Copy all already compiled files to the platform directory.
 - a. FSBL (see Section 3.1, Step 5)

cp ../test_board/prebuilt/software/4ev_1e_2gb/fsbl.elf pfm/boot/

b. Petalinux image files (see Section 3.3, Step 10)

ср	/test	board/os/peta	alinux/images/	'linux/bl31.elf	pfm/boot/
ср	/test	board/os/peta	alinux/images/	'linux/pmufw.elf	pfm/boot/
ср	/test	_board/os/peta	alinux/images/	'linux/system.dtb	pfm/boot/
ср	/test	_board/os/peta	alinux/images/	'linux/u-boot-dtb.elf	pfm/boot/u-boot.elf

cp ../test board/os/petalinux/images/linux/boot.scr pfm/sd dir/ cp ../test board/os/petalinux/images/linux/system.dtb pfm/sd dir/

c. Trenz **init.sh** script that is an place-holder for user defined bash code to be executed during the boot sequence:

cp ../test_board/misc/sd/init.sh pfm/sd_dir/



https://sp.utia.cas.cz

Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i. The platform directory te0820_15_240_pfm should contain these files:

pfm	
<u> </u>	boot
	bl31.elf
İ	fsbl.elf
ĺ	- pmufw.elf
	system.dtb
	L u-boot.elf
L	sd dir
	boot.scr
	- init.sh
	L system.dtb

4. Generate the Petalinux root file system (SYSROOT) used for cross compilation within the Vitis tool. See result of Section 3.3, Step 11.

../test_board/os/petalinux/images/linux/sdk.sh -d .

The SYSROOT directories and files for PC and for Zyng Ultrascale+ will be created in:

~/work/te0820_15_240/test_board_pfm/sysroots/x86_64-petalinux-linux ~/work/te0820_15_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-linux

NOTE: Once it is created, do not move these SYSROOT directories (due to some internally created paths).

5. Set path to the Vitis tool

source /tools/Xilinx/Vitis/2021.2/settings64.sh

6. Start Vitis

vitis

In Vitis IDE Launcher, set the workspace for the extensible platform compilation:

~/work/te0820_15_240/test_board_pfm

Close Welcome page.

- 7. Create a new platform project,
 - a. Open menu File → New → Platform Project...
 - b. Type name of the extensible platform te0820_15_240_pfm. Click Next.

		New Platform Project	•
Create new platform project			Γ.,
Enter a name for your platform	project		
This wizard will guide you thr platform. A platform will ena Platforms are currently suppo	ough creation of a p ble you to specify op orted for embedded	slatform project from the output of Vivado [Xilinx Shell Archive (XSA)] or from an existing ptions for the kernels, BSPs, as well as settings required for creating new applications. software developers.	
Platform project name: te0	820_15_240_pfm		
Platform Project	System Project	 A platform provides hardware information and software environment settings. A system project contains one or more applications that run at the same time. 	
Processor Domain	Арр	• A domain provides runtime for applications, such as operating system or BSP.	
XSA		• A workspace can contain unlimited platforms and unlimited system projects.	
A new platform project can b	pe created from one	of the two inputs:	
From hardware specification Create a new platform pro customized later from the	n (XSA) oject from a hardwa platform project ed	re specification file. You can specify the OS and processor to start with. The platform can l iitor.	be
From existing platform Load the platform definiti	on from an existing	platform. You can choose any platform from the platform repository as a base for your	
(?)		< Back Next > Cancel Finisi	1



ŪTĬA

https://sp.utia.cas.cz

11/25

- c. In Hardware Specification browse for the XSA platform file:
 ~/work/te0820_15_240/test_board/vivado/test_board_4ev_1e_2gb.xsa
- d. In Software specification select: linux

In Boot Components unselect Generate boot components

			New Platform Project				• 🙁
Platform Choose a platform for	r your project. You can als	o create an ap	plication from XSA through the 'Create	a new platform	ı from hardware (XSA)' tab.		5
🗄 Create a new plat	form from hardware (XSA	A) 🖬 Select a j	platform from repository				
Hardware Specifica XSA File: /home/d	tion level/work/te0820_15_24	0/test_board/	'vivado/test_board_4ev_1e_2gb.xsa			• Browse	
Software Specificat Specify the details file	ion for the initial domain to b	e added to the	e platform. More domains can be after	the platform is	created by double clicking t	he platform.s	pr
Operating system:	linux	•					
Processor:	psu_cortexa53	•					
Architecture:	64-bit	•					
Note: The Linu project editor Boot Components Generate boot c	ix domain added to the pla before generating the pla omponents	atform project tform.	t needs more details to generate a platf	orm. Please spe	cify the missing details in th	ne platform	
1				< Back	Next > Cancel	Finis	h

Click Finish. New window te0820_15_240_pfm is opened.

- 8. Modify *Linux Domain* of the platform.
 - a. Click on linux on psu_cortex53 to open window Domain: linux_domain.
 - b. In Description: write xrt
 - c. In Bif File find and select the pre-defied option Generate Bif
 - d. In Boot Components Directory select:

~/work/te0820_15_240/test_board_pfm/pfm/boot

e. In FAT32 Partition Directory select:

```
~/work/te0820_15_240/test_board_pfm/pfm/sd_dir
```

Elle Egit Search Xillin groject Window Help Image: Search Xillin Xillis/2012/2012/2012/2012/2012/2012/2012/201			test_board_pfm -	te0820_15_240_pfm/platform.spr - Vitis IDE					- • 😣
Image: Image:	<u>File Edit Search Xilinx Project</u>	<u>W</u> indow <u>H</u> elp							
Steplerer il Image il Imag	🗂 • 🗟 🐘 🔍 • 🐐 • O • 🤞	* • • • • • • • • •					۹ 🗖	Design	Debug
	🔒 Explorer 🛛 🕒 📴 🛍 🔋 🔍 🗆	√ te0820_15_240_pfm ¤					•	🛎 Outline 🛛	= - -
<pre> • elw • elogs • elogs • elogs • elogs • elogs • elogs • elogs • elogs • elogs • elogs • elogselogs • resources • platform.spr * platform.spr * elogselogs • elogselogs • elogselogs • elogselogselogselogselogselogselogselogs</pre>	✓ I te0820_15_240_pfm	type filter text 🛛 🖻 🖷 🍁 🕷	Domain: linux domain	n				There is no	active
 Liggs 	≻ 🗠 hw	✓ ■ te0820_15_240_pfm		R				an outline	t provides
Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona pour cortexas3 Polaciona Polacion Polaciona Polaciona Polaciona Polaciona Pola	> logs	▼ ○ psu_cortexa53	OS:	unux					
Image: Platform.tcl Supported Runtumes: Operation Supported Runtumes: Display Name: linux on psu_cortexa53 Display Name: linux on psu_cortexa53 Description: xrt Bif File: /home/devel/work/te0820_15_240/test_board_pfm/te0820_15_240_pfm/resou Bif File: /home/devel/work/te0820_15_240/test_board_pfm/te0820_15_240_pfm/resou Usuastant # • • Bif File: Bif File: /home/devel/work/te0820_15_240/test_board_pfm/te0820_15_240_fm • Bootmode Supported Runtumes: > • Bif File: /* Assistant # • • • Bif File: /home/devel/work/te0820_15_240/test_board_pfm/pfm/boot Bigowse • Bootmode SD • • • • • • • • • • • • • • • • • • • • • •	v platform.spr	+ ≡ linux on psu_cortexa53	Processor:	psu_cortexass					
Display Name: inux no pau_cortexas3 Display Name: inxx t wxt rxt Bif File: /home/devel/work/te0820_15_240_test_board_pfm/te0820_15_240_pfm/resou Bif File: /home/devel/work/te0820_15_240_test_board_pfm/te0820_15_240_pfm/resou Bot Components Directory: /home/devel/work/te0820_15_240_test_board_pfm/pfm/boot Bot Components Directory: /home/devel/work/te0820_15_240_test_board_pfm/pfm/sd_dir Browse Botonode SD ~ FAT32 Partition Directory: /home/devel/work/te0820_15_240_test_board_pfm/pfm/sd_dir Browse Sysroot Directory: /home/devel/work/te0820_15_240_test_board_pfm/pfm/sd_dir Browse QEMU Data: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem Browse Q is Main Hardware Specification Wu QEMU Arguments: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem Browse Q is Wools/Rill nor/ticl Ocosle if @ Problems @ Visit Log @ Guidance Imax/demu args_txt1 Imax/demu/data/ongs/2xndmp/sw/a53_linux/qem Browse Q is Wools/Ling.v/tilis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem Browse Q is Imax/demu/data/sw/data/sw/data/sw/data/sw/data/sw/data/sw/data/sw/data/sw/data/sw/	∠ platform.tcl	Elbraries	Supported Runtimes:	OpenCL •			_		
Assistant Console Problems Vicios/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qemu args.txt			Display Name:	linux on psu_cortexa53		_!			
Assistant # C Assistant # Assistant # C Assistant # Assistant # C Assistant # Assistant # C Assistant # Ass			Description:	xrt			-		
VAssistant # = 0 VAssistant # = 0 South Components Directory: (home/devel/work/te0820_15_240/test_board_pfm/pfm/boot Browse Boot Components Directory: (home/devel/work/te0820_15_240/test_board_pfm/pfm/boot Browse South Components Directory: (home/devel/work/te0820_15_240/test_board_pfm/pfm/sdidir Browse FAT32 Partition Directory: (home/devel/work/te0820_15_240/test_board_pfm/pfm/sd_dir Brogge System Directory: (home/devel/work/te0820_15_240/test_board_pfm/pfm/sd_dir Brogge QEMU Data: (tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browge						4			
Main Hardware Specification			Bif File:	/home/devel/work/te0820_15_240/test_board_pfm/te0820_15_240_pfm/resou	Browse	•	R.		
Assistant # ***********************************			Boot Components Directory: Linux Rootfs: Bootmode FAT32 Partition Directory: Sysroot Directory: QEMU Data:	/home/devel/work/te0820_15_240/test_board_pfm/pfm/boot	B <u>r</u> owse	Q	R.		
VAssistant # •••• Bootmode SD •• FA32 Partition Directory: /home/devel/work/te0820_15_240/test_board_pfm/pfm/sd_dir Browse Q • QEMU Data: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem Browse Q • QEMU Arguments: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem Browse Q • Main Hardware Specification • © Console # Problems Phut Option in to console • Platform Tcl Console • Identification •					Browse	Q	Ge Contraction		
FAT32 Partition Directory: /home/devel/work/te0820_15_240/pfm/pfm/sd_dir Brgwse Q Sysroot Directory: QEMU Data: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q QEMU Data: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q QEMU Arguments: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q Main Hardware Specification @ Console # Problems Vitis Log @ Guidance Platform Tcl Console @ Console # (frogls/Xilinz/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q	✓Assistant ¤ □			SD 🔻					
Sysroot Directory: QEMU Data: QEMU Data: QEMU Arguments: PMU QEMU Ar	E B 🗢 🔦 O 🎋 🕴			/home/devel/work/te0820_15_240/test_board_pfm/pfm/sd_dir	Br <u>o</u> wse	Q	R		
QEMU Data: /tools/Xilinx/Vitis/2011.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browge Q QEMU Arguments: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browge Q Main Hardware Specification /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q © Console # Problems Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q # Console # Problems Vitis Log © Guidance Imax/gemu/gemu args.1x1 Imax/gemu/gemu args.1x1	te0820_15_240_pfm [Platform]				Bro <u>w</u> se	Q	ila I		
QEMU Arguments: /tools/Xilinx/Vitis/2011.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q PMU QEMU Arguments: /tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/s53_linux/qem Browse Q Main Hardware Specification Console # 12 Problems = Vitis Log 10 Guidance D D D Platform Tcl Console Image: args /tools/Xilinx/Vitis/2021.2/data/emulation/olatforms/zynqmp/sw/s53_linux/gemu args.txt 3 D D D				/tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem	Browse	Q	2		
PMU QEMU Arguments: /tools/Xilinx/Vitis/2012/2/data/emulation/platforms/zyngmp/sw/a53_linux/gem Browse, Q Main Hardware Specification Console # D Problems D Vitis Log ① Cuidance Platform Tcl Console domain config - cemu-args //tools/Xilinx/Vitis/2021.2/data/emulation/olatforms/zyngmo/sw/a53_linux/gemu/gemu args.txt)		QEMU Arguments: PMU QEMU Arguments:	/tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem	Brows <u>e</u>	Q	A			
Main Hardware Specification Console M E Problems E Vitis Log ① Cuidance Platform Tcl Console domain config -cemu-args //tools/Xilinz/Vitis/2021.2/data/enulation/olatforms/zynomo/sw/a53 linux/cemu/gemu args.txt)			PMU QEMU Arguments:	/tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qem	Browse	Q	2		
Console M D Problems D Vitis Log ① Cuidance D		Main Hardware Specification							
Platform Tcl Console domain_configoemu-aros {/tools/Xilinx/Vitis/2021.2/data/emulation/olatforms/zvnomo/sw/a53_linux/oemu/gemu_aros.txt}		🔍 Console 🛛 🗈 Problems 🗏 Vitis I	.og ① Guidance			B,	61 B		9 D
domain config -gemu-args {/tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zyngmp/sw/a53 linux/gemu/gemu args.txt}		Platform Tcl Console							
<pre>platform write domain config -pmuqemu-args {/tools/Xilinx/Vitis/2021.2/data/emulation/platforms/zynqmp/sw/a53_linux/qemu/pmu_args.txt} platform write</pre>		domain config -qemu-args {/tr platform write domain config -pmuqemu-args platform write	ools/Xilinx/Vitis/2021.2/d {/tools/Xilinx/Vitis/2021.	ata/emulation/platforms/zynqmp/sw/a53_linux/qemu/qemu_args.txt} 2/data/emulation/platforms/zynqmp/sw/a53_linux/qemu/pmu_args.txt}					





- 9. Build the platform. In Vitis IDE **Explorer** section, click on **te0820_15_240_pfm** to highlight it. Right-click on the highlighted **te0820_15_240_pfm** and select **Build Project** in the opened submenu. Platform is compiled in few seconds.
- 10. Close the Vitis tool, menu: File \rightarrow Exit.

The Vits extensible platform *te0820_15_240_pfm* has been created in the directory:

~/work/te0820_15_240/test_board_pfm/te0820_15_240_pfm/export/te0820_15_240_pfm

3.5 Platform Inspection

When the path to Vitis tools is set, the **platforminfo** tool can be used to report the extensible Vitis platform (XPFM) information. From the **test_board_pfm** folder execute:

platforminfo te0820_15_240_pfm/export/te0820_15_240_pfm/te0820_15_240_pfm.xpfm

Report:

Basic Platform Informa	tion	
Platform: te0820_ File: /home/c export/ Description: te0820_15_240_pfm	===== 15_240_pfm level/work/t 'te0820_15_2	e0820_15_240/test_board_pfm/te0820_15_240_pfm/ 40_pfm/te0820_15_240_pfm.xpfm
Hardware Platform (She	ell) Informa	==== tion
Vendor: Board: Name: Version: Generated Version: Hardware: Software Emulation: Hardware Emulation: Hardware Emulation Pla FPGA Family: FPGA Device: Board Vendor: Board Name: Board Part:	utform:	<pre>==== trenz zusys zusys 2.0 2021.2.1 1 1 1 0 zynquplus xczu4ev trenz.biz trenz.biz:te0820_4ev_1e:2.0 xczu4ev-sfvc784-1-e</pre>
Clock Information		
Default Clock Index: Clock Index: Frequency: Clock Index: Frequency: Clock Index: Frequency: Clock Index: Frequency: Clock Index: Frequency:	4 1 100.000000 2 200.000000 3 400.000000 4 240.000000	
Memory Information Bus SP Tag: HP2 Bus SP Tag: HP3 Bus SP Tag: HPC1		
signal processing		https://sp.utia.cas.cz

13/25



```
_____
Software Platform Information
_____
Number of Runtimes:
Default System Configuration: te0820 15 240 pfm
System Configurations:
  System Config Name:
                                             te0820_15_240_pfm
  System Config Description:
                                             te0820_15_240_pfm
  System Config Default Processor Group: linux_domain
  System Config Default Boot Image: st
System Config Default Boot Image: 1
                                             standard
  System Config Is QEMU Supported:
  System Config Processor Groups:
   Processor Group Name: linux on psu_cortexa53
  Processor Group CPU Type: cortex-a53
Processor Group OS Name: linux
System Config Boot Images:
    Boot Image Name:
                                standard
    BOOT Image BIF:te0820_15_240_pfm/boot/linux.bifBoot Image Data:te0820_15_240_pfm/linux_domain/imageBoot Image Boot Mode:sd
    Boot Image RootFileSystem:
                               /mnt
    Boot Image Mount Path:
    Boot Image Read Me:
                               te0820 15 240 pfm/boot/generic.readme
    Boot Image QEMU Args:
te0820 15 240 pfm/qemu/pmu args.txt:te0820 15 240 pfm/qemu/qemu args.txt
    Boot Image QEMU Boot:
    Boot Image QEMU Dev Tree:
Supported Runtimes:
Runtime: OpenCL
```

4 Facedetect Vitis-Al Demo

At this point, we use the extensible Vitis platform compiled in previous steps. The platform will be extended with the Xilinx Deep Learning Processor Unit (DPU). To add the DPU to the platform follow steps bellow:

1. Create new directory test_board_dpu_trd:

```
mkdir -p ~/work/te0820_15_240/test_board_dpu_trd
cd ~/work/te0820 15 240/test board dpu trd
```

2. Start Vitis, in Ubuntu terminal execute

vitis

In Vitis IDE Launcher, select your working directory

~/work/te0820_15_240/test_board_dpu_trd

Click on Launch button to start Vitis.

- 3. Add Vitis-AI Repository to Vitis:
 - a. Open menu Window → Preferences.
 - b. Go to Library Repositories tab.
 - c. Add Vitis-AI by clicking **Add** button and fill the form as shown below, use absolute path to your home folder in field **Location**.



https://sp.utia.cas.cz

	Prefere	nces		• 😣
type filter text	Library Repositories			← ▼ ⇒ ▼ 8
 Xilinx Example Repositories Guidance Droject Preferences Software Repositories Toolchain Preferences Additional General C/C++ Run/Debug Team 	Repository Vitis Accelerated Libraries Repository Vitis Al Add Remove	Settings ID Name Description Location Git URL Branch	vitis-ai Vitis AI Vitis AI /home/devel/vitis_ai_2_0 2.0 Restore Defaults	Apply
? è Z			Cancel	/ and Close

d. Click Apply and Close

NOTE: Field "Location" says that the Vitis-AI repository from GITHUB has been already cloned into ~/vitis_ai_2_0 folder during the stage of the Petalinux configuration in Section 3.2. It is the same Vitis-AI 2.0 package downloaded from the branch 2.0. Use the absolute path to your home directory. It depends on the user name. The user name in the figure is "devel". Replace it by your user name.

e. Check that the library is correctly added, it appears in Libraries. Open menu **Xilinx→Libraries...**. You can see there just added Vitis-Al library marked as *Installed*.

nd:	Œ	Details:
Vitis Accelerated Libraries Repository	Installed	Name: Vitis Accelerated Libraries Repository Directory: /home/kohoutl/.Xilinx/Vitis/2021.2/vitis libraries
 ✓ Vitis DLAS Library ✓ Vitis Codec Library 	Installed	Description: The Vitis software development platform includes an
✓Vitis Data Analytics Library	Installed	libraries that offer out-of-the-box acceleration with minim
✓Vitis Data Compression Library	Installed	to zero-code changes to your existing applications.
✓Vitis Database Library	Installed	URL: https://github.com/Xilinx/Vitis_Libraries/tree/2021.2
✓Vitis DSP Library	Installed	
✓Vitis Genomics Library	Installed	
✓Vitis Graph Library	Installed	
✓Vitis HPC Applications	Installed	
✓Vitis Quantitative Finance Library	Installed	
✓Vitis Security Library	Installed	
✓Vitis Solver Library	Installed	
✓ Vitis SPARSE Library	Installed	
✓Vitis Utility Library	Installed	
✓Vitis Vision Library	Installed	
🖻 Vitis Al	Installed	
🕶 dsa		
🕶 XVDPU-TRD		
🝷 🗠 vck190 platform		
✓ ∞ overlays		
Vitis_Libraries		
✓Vitis Vision Library	Installed	Add to project x
		Add to project
Defrech - Last updated on Dec 13, 2022, 1:3	8:56 PM	ОК
Pofresh - Last updated on Dec 13, 2022, 1:3	8:56 PM	ОК

- Create a Vitis-AI design for our te0820_15_240 custom platform. 4.
 - Select File → New → Application Project.... Click Next. Skip welcome page if it is a. shown.
 - Click on + Add icon and select the custom extensible platform te0820_15_240_pfm b. located in directory:

~/work/te0820 15 240/test board pfm/te0820 15 240 pfm/export/te0820 15 240_pfm.

- Click Next. C.
- In Application Project Details window type into Application project name: d. dpu_trd
- Click Next. e.
- f. In **Domain window** type (or select by browse):

Field	Value
Sysroot path	~/work/te0820_15_240/test_board_pfm/sysroots/cortexa72-cortexa53- xilinx-linux
Root FS	~/work/te0820_15_240/test_board/os/petalinux/images/linux/rootfs.ext4
Kernel Image	~/work/te0820_15_240/test_board/os/petalinux/images/linux/Image

Click Next. g.

	New Applicatio	n Project	0 😫
Domain			•••
Select a domain for your project or create a new domain			
Select the domain that the application would link to or create a new domain			
Note: New domain created by this wizard will have all the requirements of the application ter	nplate selected in the r	next step	
Select a domain	Domain details		
linux on psu_cortexa53	Name:	linux_domain	
	Display Name:	linux on psu_cortexa53	
	Operating System:	linux •	
	Processor:	psu_cortexa53	
	Application settings		
	Sysroot path: /ho	me/devel/work/te0820_15_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-linux	Browse
	Root FS: /ho	me/devel/work/te0820_15_240/test_board/os/petalinux/images/linux/rootfs.ext4	Browse
	Kernel Image: /ho	me/devel/work/te0820_15_240/test_board/os/petalinux/images/linux/Image	Browse
Ð		< Back Next > Cancel	Finish

- h. In the dsa folder, select: DPU Kernel (RTL Kernel).
- i. Click Finish. New project template is created now.

Templates			
Select a template to create your project.			
Available Templates:			
Find: dpu 🗳 (1 match) 🖻 🗷	DPU Kernel (RTL Kernel)		
 Acceleration templates with PL and AIE accelerators 	Run DPU Kernel as GUI flow		
▼ dsa	Location:		
DPU Kernel (RTL Kernel)	/opt/Xilinx/vitis_ai_2_0_git/dsa/DPU-TRD		
	Key concepts:		
□ Show only certified examples	• RTL Kernel		
Vitis IDE Examples Vitis IDE Libraries			
3	< Back Next > Cancel		





- 5. Configure project and DPU:
 - a. In dpu_trd window switch **Active build configuration** from Emulation-SW to **Hardware**.
 - b. Configure DPU to utilize internal Ultra RAMs instead of Block RAMs. Open file

dpu_trd_kernels/src/prj/Vitis/dpu_conf.vh

and change line 37 from

`define URAM DISABLE

to

`define URAM_ENABLE

NOTE: this step applies to FPGAs with Ultra RAMs, which is the case of the Xilinx *zcu04-ev* device.

This modification is necessary for successful implementation of the DPU on the used module.

- c. Go to **dpu_trd_system_hw_link** and open **dpu_trd_system_hw_link.prj** file. Remove **sfm_xrt_top** kernel from binary container by right clicking on it and choosing remove. **Reduce number of DPU kernels to one**.
- d. Configure connection of DPU kernels. On the same tab right click on **dpu** and choose **Edit V++ Options**.

Click "..." button on the line of V++ Configuration Settings and modify configuration as follows:

```
[clock]
freqHz=20000000:DPUCZDX8G_1.aclk
freqHz=40000000:DPUCZDX8G_1.ap_clk_2
[connectivity]
sp=DPUCZDX8G_1.M_AXI_GP0:HPC1
sp=DPUCZDX8G_1.M_AXI_HP0:HP2
sp=DPUCZDX8G_1.M_AXI_HP2:HP3
```

test board dou trd - dou trd system hw link/dou trd system hw link.ori - Vi

File Edit Search Xilinx Proiect	Window Help)	sc_board_apa_cra = apa_cra_s	system_nw_ank/apa_tra_sy				
□ - □ · □ · □ · • · • · • • • • • • • • • •								
Explorer 🛛 🗉 😵	èn 8 = 0	🛎 dpu	trd system 🔀 dpu trd 🔀	dpu trd system hw link 🛱			🗄 Outline 🛙	
✓ ■ dpu_trd_system [te0820_15_2 > ♥ dpu_trd_kernels	40_pfm]	😠 Hai	rdware Link Project Se	ettings	Active build configuration: Hardwa	are 🔹 🔊	There is no activ that provides ar	ve editor n outline.
✓ [™] dpu_trd_system_hw_link [pl	1	Gene	ral	Options				
% dpu_trd_system_hw_link.pr	i i	Ргоје	ct name: <u>dpu_trd_system_hw</u>	<u>link</u> Target:	Hardware			
• 🖶 dpu_trd [linux on psu_corte>				Binary Container	Settings		- 8	
> @ Includes	type filter te	xt E	= dpu				0+0+8	1
Emulation-HW	- [™] dpu trd s	vstem	News	[dev				
→ 🐸 Hardware	> @dpu_trd	, ,	Name:	dpu				
⊧ 🧉 src	> 目dpu_trd	kernel	Trace memory:	FIFO - 8K -				
🔀 dpu_trd.prj	- Edpu_trd	_systen	V++ configuration settings:	[clock] freqHz=200000000	DPUCZDX8G_1.aclk freqH; App	bly		
⊧ œ_ide	• < Emula	tion-SV	V++ command line options:	V++	configuration settings	*		
4 dpu_trd_system.sprj	 * * Emula * * Hardw 	cion-H\	Compute Unit Settings	Enter v++ linker settings to	add to the generated config file	_		
🖌 Assistant 🛛 📄 🖷 🗢 🔦 🧿	≡ dpu	die	Name Co	[clock]		Data 1	ransfer Exec	
 ♥ dpu_trd_system [System] ♥ dpu_trd_system_hw_link [Hw < Emulation-SW [Software Ei ♥ dpu < © Dpu 			V++ Linker Command Line - Einfichter Command Line - Einfichter Comment Einfichter Comment Einfichter Comment Einfichter Comment Einfichter Comment	freqHz=200000000:DPUCZ freqHz=400000000:DPUCZ [connectivity] sp=DPUCZDX8G_1.M_AXI_ sp=DPUCZDX8G_1.M_AXI_	DX8G_1.aclk /DX8G_1.ap_clk_2 GP0:HPC1 HP0:HP2	Data		
 Fmulation-HW [Hardware F 			0.000 0.0000 0.00000 0.00000 0.000000	<pre>sp=DPUCZDX8G_1.M_AXI_</pre>	HP2:HP3	0100 0100 0100 0100 0100 0100	000000000000000000000000000000000000000	
- ≡ dpu						evert		
 DPUCZDX8G [User Mar Hardware [Hardware] dpu 		Build C	onsole [dpu_trd, Emulation-SV	0			pply and Close	8
OPUCZDX8G [User Man DPUCZDX8G [User Man dpu_trd_kernels [Hw Kernel]	naged] mulation] ged]				Cancel	К		
department of	ng						https://sp.u	utia.cas.cz

17/25

ŪTĬA

- Build DPU TRD application 6.
 - In Explorer section of Vitis IDE, click on dpu trd system[te0820 15 240 pfm] to а select it.
 - b Right Click on dpu trd system[te0820_15_240_pfm] and select in the opened sub-menu Build project.
- Run the board with the DPU TRD design. 7.
 - Write sd card.img to SD card using SD card reader. The image file is an output a. product of the Vitis compilation and packaging phase. It is located in directory ~/work/te0820_15_240/test_board_dpu_trd/dpu_trd_system/Hardware/ package/

NOTE: For writing the image file to the SD card can be used BalenaEtcher tool downloadable from https://www.balena.io/etcher. The tool is available for Windows, Linux or MAC platforms.

- Boot the board and open a terminal connection, USB UART or SSH. b.
 - UART connection via USB UART/JTAG (connector J8 on the board). On your host machine identify the device to be connected (COM port on Windows or ttyUSB device on Linux). On Windows machine it can be any port. On Linux machine are USB serial ports counted from zero. As the board provides two virtual devices and the second one is the UART, the device will be /dev/ttyUSB1 (in case that only one board is connected). To connect the board use PUTTY, for instance. The settings are:
 - Baud rate 115200 \cap
 - Data bits 8 0
 - Stop bits 1 0
 - Parity none 0
 - Flow control none 0

NOTE: If you want to forward graphical applications started from this terminal to the board monitor connected via display port, you should also set DISPLAY variable correctly (export DISPLAY=:0.0).

SSH connection via Ethernet (preffered). To find out the board IP address use UART connection, execute command:

ifconfig

Or from Linux host machine, you can use *arp-scan* command. Example:

```
sudo arp-scan -l -I enp4s0
Interface: enp4s0, type: EN10MB, MAC: c4:6e:1f:01:b9:0d, IPv4:
10.42.0.1
Starting arp-scan 1.9.7 with 256 hosts
(https://github.com/royhills/arp-scan)
                 80:1f:12:d0:7d:0a Microchip Technology Inc.
10.42.0.102
```

```
1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.997 seconds (128.19
hosts/sec). 1 responded
```

NOTE: The SSH connection should have X11 forwarding activated, if you want to forward graphical applications to your host machine. You should also set DISPLAY variable correctly (export DISPLAY=:10.0).



https://sp.utia.cas.cz



c. Resize the second partition of the SD card to its maximal size, from the board terminal use command *resize-part*:

```
resize-part /dev/mmcblk1p2
/dev/mmcblk1p2
Warning: Partition /dev/mmcblk1p2 is being used. Are you sure you want to
continue?
Yes/No? yes
End? [4295MB]? 100%
Information: You may need to update /etc/fstab.
resize2fs 1.45.6 (20-Mar-2020)
Filesystem at /dev/mmcblk1p2 is mounted on /media/sd-mmcblk1p2; on-line
resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/mmcblk1p2 is now 3539968 (4k) blocks long.
```

- d. Initialize HDMI output on the TE0701 carrier board to display X11 desktop on the monitor. The HDMI output is fixed at 1920x1080p60 resolution.
 - Copy prebuilt SW application for HDMI output from the attached package to board. Connect the board using SFTP and copy :

hdmio/hdmio.elf to /mnt/sd-mmcblk1p1/

Source codes of the application are also part of the attached ZIP file.

Make file hdmio.elf executable, from the board terminal execute:

chmod +x /mnt/sd-mmcblk1p1/hdmio.elf

• Configure the *X11* server to start the *hdmio.elf* application automatically on when it starts. Connect the board using SFTP and copy from the enclosed ZIP file to the board:

hdmio/01hdmio.sh to /etc/X11/Xsession.d

Make file 01hdmio.sh executable, from the board terminal execute:

chmod +x /etc/X11/Xsession.d/01hdmio.sh

 Configure X11 server to use a Simple Frame Buffer. On the board modify file /etc/X11/xorg.conf:

```
Section "InputDevice"
 Identifier "System Mouse"
  Driver "mouse"
 Option "Device" "/dev/input/mouse0"
EndSection
Section "InputDevice"
 Identifier "System Keyboard"
 Driver "kbd"
 Option "Device" "/dev/input/event0"
EndSection
Section "Screen"
  Identifier "DefaultScreen"
 Device "/dev/fb0"
 DefaultDepth 24
EndSection
Section "Device"
 Identifier "myfb"
 Driver "fbdev"
 Option "fbdev" "/dev/fb0"
EndSection
```

signal processing

•

```
19/25
```

https://sp.utia.cas.cz



The modified file is included within the attached ZIP file in folder hdmio. On the next boot, the Petalinux desktop will be displayed on the monitor. To reboot the board you can use command

reboot

- Run the *facedetect* demo using DPU. The input data are taken from the HDMI input on 8. the AVNET FMC card a processed data are shown on the monitor connected via HDMI output of the AVNET FMC card.
 - Copy the precompiled executable file of the *facedetect demo* and the corresponding a. model for the DPU from the enclosed ZIP package. Connect the board using SFTP and copy to the board folder:

facedetect/ to /home/root

Source codes of the *facedetect* application are also part of the attached ZIP file.

- Make the *facedetect* SW application executable, from the board terminal execute: b. chmod +x /home/root/facedetect/hdmi facedet2.elf
- Copy scripts optimizing the data transfers between the DPU and the memory C. controller from the PC to the home folder on the board. These scripts are available in attached ZIP file. To copy them use SFTP:

dpu_sw_optimize to /home/root

- d. Switch back to the board terminal.
- Optimize the data transfers between the DPU and the memory controller, execute: e.

```
cd /home/root/dpu_sw_optimize/zynqmp
chmod -R +x *
./zynqmp_dpu_optimize.sh
```

NOTE: The scripts are modified compare to scripts provided by Xilinx.

f. Set needed environment variable pointing the DPU firmware.

export XLNX VART FIRMWARE=/mnt/sd-mmcblk1p1/dpu.xclbin

- Connect the HDMI input on the AVNET FMC card. It can be an HDMI camera or an g. output from a PC. In the case of PC output, just play some video where there are people.
- Connect an HDMI monitor to the HDMI output of the AVNET FMC card. The output h. video signal is 1920x1080p60.
- i. Run facedetect demo

cd /home/root/facedetect ./hdmi facedet2.elf

The result is shown on the monitor.

5 VCU Decoder

The platform includes H.264/H.265 Video Codec Unit (VCU), which is configured to use the decoder. The encoder is disabled, because the currently used FPGA does not have enough free resources for it. To test the decoder, follow these steps.

1. Upload any MP4 H.264 encoded video to the board. Connect the board using SFTP and copy the video file to the board folder /home/root/.





The video can be downloaded here, for instance:

http://distribution.bbb3d.renderfarming.net/video/mp4/bbb_sunflower_1080p_60fps_nor_mal.mp4.

The name of the example video is bbb_sunflower_1080p_60fps_normal.mp4.

If the board has the Internet connection, the video file can be downloaded directly to the board, from the board terminal execute:

cd /home/root wget http://distribution.bbb3d.renderfarming.net/video/mp4/bbb sunflower 1080p 60fps normal.mp4

2. Inspect the video file on the board, use command gst-discover-1.0:

```
cd /home/root
gst-discoverer-1.0 bbb sunflower 1080p 60fps normal.mp4
Analyzing file:///home/root/bbb_sunflower_1080p_60fps_normal.mp4
Done discovering file:///home/root/bbb sunflower 1080p 60fps normal.mp4
Missing plugins
Topology:
 container: Quicktime
    audio: AC-3 (ATSC A/52)
    audio: MPEG-1 Layer 3 (MP3)
    video: H.264 (High Profile)
Properties:
 Duration: 0:10:34.533333333
 Seekable: yes
 Live: no
 Tags:
      audio codec: MPEG-1 audio
     maximum bitrate: 165120
     bitrate: 160000
     datetime: 2013-12-16T17:59:32Z
     title: Big Buck Bunny, Sunflower version
     composer: Sacha Goedegebure
      artist: Blender Foundation 2008, Janus Bager Kristensen 2013
     comment: Creative Commons Attribution 3.0 -
http://bbb3d.renderfarming.net
      genre: Animation
      QT atom: buffer of 39 bytes
      container format: ISO MP4/M4A
      has crc: false
      channel mode: joint-stereo
      video codec: H.264 / AVC
```

3. Play the video

a. Petalinux desktop

```
export DISPLAY=:0.0
```

```
gst-launch-1.0 filesrc location=bbb_sunflower_1080p_60fps_normal.mp4 ! \
    qtdemux name=demux demux.video_0 ! h264parse ! \
    omxh264dec ! queue ! videoconvert! Autovideosink
```

b. X11 forwarding via SSH

export DISPLAY=:10.0

gst-launch-1.0 filesrc location=bbb_sunflower_1080p_60fps_normal.mp4 ! \
 qtdemux name=demux demux.video_0 ! h264parse ! \
 omxh264dec ! queue ! videoconvert! Autovideosink

21/25





Both methods are too slow because of the output video software processing after the VCU decodes the data. To get the true performance of the VCU, displaying of the decoded video data has to be suppressed.

```
gst-launch-1.0 filesrc location=bbb_sunflower 1080p 60fps normal.mp4 ! \
               qtdemux name=demux demux.video 0 ! h264parse ! \
               omxh264dec ! queue ! videoconvert! \
               fpsdisplaysink text-overlay=0 video-sink=fakevideosink -v
```

Automations and Optimizations 6

Some steps can be automatized to avoid doing them after each boot.

6.1 QoS

To automatize starting the scripts from the /home/root/dpu sw optimize folder follow next steps:

1. Copy file **sripts/qos.sh** from the provided ZIP file to the **/etc/init.d/** folder on the board, use SFTP. Make it executable:

chmod +x /etc/init.d/gos.sh

2. Create a link of the **gos.sh** file for the runlevel 5.

ln -s /etc/init.d/qos.sh /etc/rc5.d/S99qos

6.2 Monitor Optimizations

A monitor connected via the HDMI port on the TE0701 (Petalinux desktop) will go into sleep mode when keyboard and mouse are in idle state. To disable this behavior append an extra section to the file /etc/X11/xorg.conf:

```
Section "ServerFlags"
    Option "BlankTime" "0"
EndSection
```

The modified file is included within the attached ZIP file in folder hdmio.

6.3 DPU Firmware

To set a variable pointing the DPU firmware automatically during the boot time, copy file sripts/dpu fw.sh from the provided ZIP file to the /etc/profile.d/ folder on the board, use SFTP. Make the file executable:

chmod +x /etc/profile.d/dpu fw.sh

Resource	Utiliz	ation	Available	Utilization %		
	All	DPU		All	DPU	
LUT	73128	48255	87840	83.25	54.94	
LUTRAM	9304	5595	57600	16.15	9.71	
FF	138250	97563	175680	78.69	55.53	
BRAM	99	84	128	77.34	65.63	
URAM	48	46	48	100.00	95.83	
DSP	710	690	728	97.53	94.78	
10	62	0	252	24.60	0.00	
BUFG	13	0	352	3.69	0.00	
MMCM	3	0	4	75.00	0.00	

Used Resources 7



22/25





8 **Power Consumption**

Conditions of the power consumption measurement:

- Measured with wattmeter Electrobock EMF-1
 - Voltage range: 190-276 VAC (accuracy +/-1%)
 - Current range: 0.01-16A (+/-1%)
 - Power range: 0.2-4416 W (+/-1%)
- Measurement covers these components:
 - Power adapter.
 - Connected to the Ethernet.
 - USB connections: keyboard, mouse.
 - Connected to the HDMI monitor, used resolution is 1920x1080p60 (HDMI out on TE0701 - Petalinux desktop).
 - Connected to the HDMI monitor, used resolution is 1920x1080p60 (HDMI out on AVNET FMC card – *facedetect* demo out).
 - Running demo: *facedetect*.
 - Connected UART/JTAG USB.

The power consumption was measured in two situations:

- 1. The whole system was running without the *facedetect* demo: **11.5 W**.
- 2. Running the *facedetect* demo: 14.8 W.

23/25



https://sp.utia.cas.cz

Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.

9 Package Content



10 References

- [1]. Trenz Electronic, "MPSoC Module with Xilinx Zynq UltraScale+ ZU4EV-1I, 2 GByte DDR4 SDRAM, 4 x 5 cm," [Online]. Available: <u>https://shop.trenz-electronic.de/en/TE0820-05-4DI21MA-MPSoC-Module-with-Xilinx-Zynq-UltraScale-ZU4EV-1I-2-GByte-DDR4-SDRAM-4-x-5-cm</u>.
- [2]. Trenz Electronic, "Carrier Board for Trenz Electronic 4 x 5 Modules," [Online]. Available: <u>https://shop.trenz-electronic.de/en/TE0701-06-Carrier-Board-for-Trenz-Electronic-4-x-5-Modules</u>.
- [3]. AVNET, "AES-FMC-HDMI-CAM-G," [Online]. Available: <u>https://www.avnet.com/shop/us/products/avnet-engineering-services/aes-fmc-hdmi-cam-g-3074457345635221625/</u>.
- [4]. Trenz Electronic, UTIA, "TE0820 test board Vitis AI Tutorial," [Online]. Available: <u>https://wiki.trenz-electronic.de/display/PD/TE0820+test+board+Vitis+AI+Tutorial</u>.

24/25



https://sp.utia.cas.cz

Disclaimer

This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by UTIA AV CR v.v.i., and to the maximum extent permitted by applicable law:

(1) THIS APPLICATION NOTE AND RELATED MATERIALS LISTED IN THIS PACKAGE CONTENT ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND UTIA AV CR V.V.I. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and

(2) UTIA AV CR v.v.i. shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or UTIA AV CR v.v.i. had been advised of the possibility of the same.

Critical Applications:

UTIA AV CR v.v.i. products are not designed or intended to be fail-safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of UTIA AV CR v.v.i. products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.

25/25

