

**Application Note** 



# Support for TE0821 modules with Vitis AI 3.0 DPU

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, kadlec@utia.cas.cz, zdenek.pohl@utia.cas.cz, kohoutl@utia.cas.cz

### **Revision history**

Rev.	Date	Author	Description
v01	1.2.2024	J.K	Initial release
v02	2.2.2024	J.K	Manual creation of extensible platform
v03	20.2.2024	J.K	Added fast track script, 3 module types
v04	17.2.2025	J.K.	Modified references

### Contents

1	Introduction	1
	1.1 Low cost systems used by UTIA in EECONE T4.3 and T4.4	2
	1.2 Module based systems used by UTIA in EECONE T4.3 and T4.4	3
	1.3 Objective of This Application Note and Evaluation Package	
2		5
	2.1 Reference HW for TE0821 module	
	2.2 Create Extensible Platform HW	
	2.3 Fast Track for Creation of Extensible platform HW	22
	2.4 Validate Design	
	2.5 Compile Created HW and Custom SW with Trenz Scripts	
	2.6 Copy Created Custom First Stage Boot Loader	
3	5 11	
	3.1 Vitis AI 3.0 support	
	3.2 Building Petalinux for Extensible Design Flow	
	3.3 Disable CPU IDLE in Kernel Config	
	3.4 Add EXT4 rootfs Support	
	3.5 Let Linux Use EXT4 rootfs During Boot	
	3.6 Build PetaLinux Image	
	3.7 Create Petalinux SDK	
	3.8 Copy Files for Extensible Platform	
	3.9 Create Extensible Platform zip File	
	3.10 Generation of SYSROOT	
	3.11 Generation of Extensible Platform for Vitis	
4	Platform Usage	
	4.1 Read Platform Info	
	4.2 Create and Compile Vector Addition Example	
_	4.3 Run Compiled test_vadd Example Application	
5	Vitis AI 3.0 DPUCZDX8V_VAI_v3.0 Installation	
	5.1 Create and Build Vitis Design.	
	5.2 Add DPU Project template to the Vitis Extensible Flow	
	5.3 Configure Project for the Vitis Extensible Flow with DPU	
	5.4 Configure Connection of DPU kernel	
~	5.5 Build the test_dpu_trd Project	
6	Prepare SD card with test_dpu_trd DPU	50
	6.1 Resize EXT4 Partition	
	6.2 Test the Integrated DPUCZDX8G	51
	6.3 Remote Monitoring and Configuration Support	
	6.4 Remote Control from Ubuntu X11 Desktop.	54
	6.5 Remote Control in x-session-manager on Ubuntu X11 Desktop	54
	6.6 Display Test Pattern and Test USB Camera	55
	6.7 Vitis AI 3.0 TE0821-01-2AE31KA, TE0701-06, DPU (B1024)	
	6.8 Vitis AI 3.0 TE0821-01-3AE31KA, TE0701-06, DPU (B1600)	
7	6.9 Vitis AI 3.0 TE0821-01-3BE21FA, TE0701-06, DPU (B1600)	
7	References	01



ŪTIA

### Acknowledgement

The EECONE project is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112065.

https://zs.utia.cas.cz/index.php?ids=projects/eecone

https://eecone.com/eecone/home/



https://sp.utia.cas.cz

iii

# 1 Introduction

EECONE project <u>https://eecone.com/eecone/home/</u> work package 4, task 4.3 is investigating measures to support second life of electronics due to modular design.

Work package 4 task 4.4 is investigating measures to support extension of life of electronics due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system.

UTIA AV CR, v.v.i. (Institute of Information Theory and Automation of the Czech Academy of Sciences, in short UTIA) is not-for profit research institute located in Prague, Czech Republic. UTIA is involved as partner in both tasks, T4.3 and T4.4.

Both EECONE task require specification of comparable reference systems which are based on modular HW with potential for "second life" by reuse of modules or use cost optimized PCB HW without modularity.

Systems (with HW modularity or low cost single PCB) should be capable to perform similar challenging tasks. Systems have to be capable to accelerate in HW AI inference algorithms with video camera input for edge application like person detection, face detection, car-make or car-type detection and graphical output to local display or to the remote PC connected by wired Ethernet in a local network.

Systems should also support remote monitoring and control from remote PC connected by wired Ethernet in a local network.

The investigated measures and methodologies to support "second life" of electronic modules (T4.3) and measures to support extension of life of electronics (T4.4) due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system. We target developers designing the final commercial, Al inference based edge applications, mainly in the area of home automation.

Based on these requirements UTIA have selected two types of systems:

- Low cost systems. See [2], [3]
- Modul based systems. See [4] and [5]. [4] is this application note.

Both compared types of systems use STMicroelectronic STM32H573I-DK board for:

- local system control on small graphical touch screen display
- remote system control from www browser based on www-server or secure communication based on mqtt client. Board is supported by STMicroelectronic CubeMX SW framework and also by NetXDuo SW framework on top of ThreadX OS and FileX SW package.

The MCU used on STM32H573I-DK board is a 40nm chip with 32 bit ARM M33 MCU operating with 250 MHz clock, 2 MBytes of program flash memory and 640 KBytes of RAM.

Compared systems use 16nm AMD ZynqUltrascale+ device with 64 bit ARM A53 Microprocessor and programmable logic in the same device and Petalinux OS.

- Low-cost systems have an AMD ZynqUltrascale+ device and DDR4 with all peripheral interfaces soldered on a single, low cost PCB
- Module-based systems have an AMD ZynqUltrascale+ device and DDR4 soldered on an 4x5 cm module connected by connectors to a carrier board with all peripheral interfaces



1/61



# 1.1 Low cost systems used by UTIA in EECONE T4.3 and T4.4

[1]	STM32H573I-DK	https://www.st.com/en/evaluation- tools/stm32h573i-dk.html	Local or remote system control (www-server or secure mqtt client) for [2], [3]
[2]	TE0802-02-1BEV2-A	https://shop.trenz- electronic.de/en/TE0802-02- 1BEV2-A-MPSoC-Development- Board-with-AMD-Zynq-UltraScale- ZU1EG-and-1-GB-LPDDR4?c=474	AMD Vitis AI 3.0 AMD DPU in PL USB camera HD VGA display or remote X11 desktop
[3]	TE0802-02-2AEV2-A	MPSoC Development Board mit AMD Zyng™ UltraScale+™ ZU2 und 1 GB LPDDR4   Trenz Electronic GmbH Online Shop (EN) (trenz-electronic.de)	AMD Vitis AI 3.0 AMD DPU in PL USB camera HD VGA display or remote X11 desktop





2/61

https://sp.utia.cas..cz

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

# 1.2 Module based systems used by UTIA in EECONE T4.3 and T4.4

[1]	STM32H573I-DK TE0701-06 Carrier Board for Trenz Electronic 4 x 5 Modules TE0821 or TE0821	https://www.st.com/en/evaluation- tools/stm32h573i-dk.html https://shop.trenz- electronic.de/en/TE0701-06- Carrier-Board-for-Trenz-Electronic- 4-x-5-Modules?c=261	Local or remote system control (www-server or secure mqtt client) for [4], [5] Carrier Board for range of 4x5 cm modules [4], [5].
[4]	TE0821 Module: <b>17 module types</b> (to be supported)	https://shop.trenz- electronic.de/en/Products/Trenz- Electronic/TE08XX-Zynq- UltraScale/TE0821-Zynq- UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera remote X11 desktop
[5]	TE0820 Module: <b>100 module types</b> (to be supported)	https://shop.trenz- electronic.de/en/Products/Trenz- Electronic/TE08XX-Zynq- UltraScale/TE0820-Zynq- UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera remote X11 desktop





3/61

https://sp.utia.cas..cz

**UTTA** Akad Ústav

Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.  $\ensuremath{\textcircled{}}$  © 2024 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

This application note [4] and the accompanying evaluation package describe support for systems based on TE0821 modules. It is available for free public download from UTIA server dedicated to UTIA contributions to EECONE project: https://zs.utia.cas.cz/index.php?ids=projects/eecone

It will be also available for free public download in format of an wiki tutorial on Trenz-Electronic wiki server:

<u>https://wiki.trenz-</u> electronic.de/display/PD/Vitis+AI+and+Vitis+Acceleration+Tutorials+with+Trenz+Electronic+ Modules

# 1.3 Objective of This Application Note and Evaluation Package

This application note and the accompanying evaluation package describe system [4].

This application note describes how to design custom HW platform with AMD DPU for Vitis 2022.2 AI 3.0 inference for family of Trenz Electronic modules TE0821 with AMD Zynq Ultrascale+ device.

This application note [5] is using AMD Vitis 2022.2 and PetaLinux 2022 tools installed on Ubuntu 20.04. The described configuration integrated AMD DPU IP, version v4.1.0, with architecture DPUCZDX8G.

Described board configuration can operate as small standalone computer with 1 Gb Ethernet connectivity, and remote X11 desktop. Support package for this application note will be available for public download from [5].

The installed AMD DPU configurations require recompilation of Vitis AI 3.0 examples and inference models in the Vitis AI framework. This compilation process will be described in separate application note [6].

This application supports family of TE0821 modules listed in next tables with ID 1 to 17.

Vitis AI 3.0 sample application results are presented for these modules:

- ID=1 module: TE0821-01-2AE31KA, device xczu2cg-sfvc784-1-e, 4GB DDR4
- ID=3 module: TE0821-01-3AE31KA, device xczu3cg-sfvc784-1-e, 4GB DDR4
- ID=5 module: TE0821-01-3BE21FA, device xczu3eg-sfvc784-1-e, 2GB DDR4

Module TE0821-01-2AE31KA has two A53 ARM cores and 4 GB DDR4. It is possible to implement configurations of the AMD DPU (from B512 up to B1024) in PL. Implementation of B1024 is demonstrated.

Module TE0821-01-3AE31KA has two A53 ARM cores and 4 GB DDR4. It is possible to implement configurations of the AMD DPU (from B512 up to B1600) in PL. Implementation of B1600 is demonstrated.

Module TE0821-01-3BE21FA has four A53 ARM cores and 2 GB DDR4. It is possible to implement configurations of the AMD DPU (from B512 up to B1600) in PL. Implementation of B1600 is demonstrated.

department of signal processing

4/61

https://sp.utia.cas..cz



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

Specification for each module ID defined in TE0821\_board\_files.csv file is input to the Vivado 2022.2 HW bring-up scripts. Is provided by company Trenz Electronic. It is part of the package provided for complete family of modules TE0821 for AMD Vivado 2022.2.

List of supported TE0821 modules is reprinted from TE0821\_board\_files.csv file included in the evaluation package associated of this application note.

This application note and associated evaluation package enables support for "second-life" of 17 types of TE0821 modules.

Modules mighed have been used originally in another context. That context might become obsolete or outdated, now. We provide support for reuse of such modules again in a large and challenging range of Vitis AI 3.0 HW accelerated applications.

				SHORTNAM			DCD D	DDR		EMM
ID	PRODID	PARTNAME	BOARDNAME	E	ZYNQFLASHTYP	FPGAFLASHTYP	EV	_SIZ E	FLASH SIZE	C_SIZ E
1	TE0821-01-2AE31KA	xczu2cg-sfvc784-1-e	trenz.biz:te0821 2cg 1e:part0:3.0	2cg 1e 4gb	qspi-x8-dual parallel	mt25qu512-qspi-x8-dual parallel	REV01	4GB		64GB
2	TE0821-01-2AE31PA	xczu2cg-sfvc784-1-e	trenz.biz:te0821_2cg_1e:part0:3.0	2cg_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	64GB
3	TE0821-01-3AE31KA	xczu3cg-sfvc784-1-e	trenz.biz:te0821_3cg_1e:part0:3.0	3cg_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	64GB
4	TE0821-01-3AE31PA	xczu3cg-sfvc784-1-e	trenz.biz:te0821_3cg_1e:part0:3.0	3cg_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	64GB
5	TE0821-01-3BE21FA	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
6	TE0821-01-3BE21FC	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
7	TE0821-01-3BE21FL	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
8	TE0821-01-3BE21MA	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
9	TE0821-01-3BE21ML	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
10	TE0821-01-3BE91ND	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:3.0	3eg_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	32GB
11	TE0821-01-3BI21FA	xczu3eg-sfvc784-1-i	trenz.biz:te0821_3eg_1i:part0:2.0	3eg_1i_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
12	TE0821-01-3BI21FL	xczu3eg-sfvc784-1-i	trenz.biz:te0821_3eg_1i:part0:2.0	3eg_1i_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
13	TE0821-01-3BI21MA	xczu3eg-sfvc784-1-i	trenz.biz:te0821_3eg_1i:part0:2.0	3eg_1i_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
14	TE0821-01-4DE31FL	xczu4ev-sfvc784-1-e	trenz.biz:te0821_4ev_1e:part0:3.0	4ev_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	8GB
15	TE0821-01-4DE31ML	xczu4ev-sfvc784-1-e	trenz.biz:te0821_4ev_1e:part0:3.0	4ev_1e_4gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	4GB	128MB	8GB
16	TE0821-01-S003	xczu3eg-sfvc784-1-e	trenz.biz:te0821_3eg_1e:part0:2.0	3eg_1e_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB
17	TE0821-01-S004	xczu3cg-sfvc784-1-i	trenz.biz:te0821_3cg_1i:part0:2.0	3cg_1i_2gb	qspi-x8-dual_parallel	mt25qu512-qspi-x8-dual_parallel	REV01	2GB	128MB	8GB

Supported modules with  $ID = 1 \dots 17$ 

# 2 Prepare Reference Design for Extensible Custom Platform

The design proces is described on module with ID=3: TE0821-01-3AE31KA has two A53 ARM cores and 4 GB DDR4. If your module has different ID, replace 3 with that ID.

In Ubuntu terminal, source paths to Vitis and Vivado tools by

\$ source /tools/Xilinx/Vitis/2022.2/settings64.sh

Download TE0821 test\_board Linux Design file(see Reference Design download link on chapter <u>Requirements</u>) with pre-build files to

~/Downloads/TE0821-test\_board-vivado\_2022.2-build\_8\_20230919135517.zip

This TE0821 test\_board ZIP file contains bring-up scripts for creation of Petalinux for range of modules in zipped directory named "test\_board".

Unzip the file to directory:

~/work/TE0821 3 240

All supported modules are identified in file:

signal processing

5/61



 We will select module ID=3 with name TE0821-01-3AE31KA, device xczu3cg-sfvc784-1-e, 4GB DDR4

on TE0701-06 carrier board. We will use default clock 240 MHz. That is why we name the package TE0821\_3\_240 and proposed to unzip the TE0821 test\_board Linux Design files into the directory:

~/work/TE0821 3 240

### 2.1 Reference HW for TE0821 module

In Ubuntu terminal, change directory to the test\_board directory:

\$ cd ~/work/TE0821 3 240/test board

Setup the test\_board directory files for a Linux host machine. In Ubuntu terminal, execute:

\$ chmod ugo+rwx ./console/base sh/\*.sh

```
$ chmod ugo+rwx ./_create_linux_setup.sh
```

```
$ ./_create_linux_setup.sh
```

Select option (0) to open Selection Guide and press Enter

r₁ de	vel@ubuntu: ~/work/te0821_3_240/test_boar	rd C	٦ =		×
h devel@ubuntu:~/work/ .sh devel@ubuntu:~/work/ Run Design with:	<pre>te0821_3_240/test_board\$ chmod ugo+ te0821_3_240/test_board\$ chmod ugo+ te0821_3_240/test_board\$ ./_create_ Set design paths _create_linux_setup.sh /home/devel/work/te0821_3_240/test_</pre>	linux	/_crea _setup 	te_lin	_ I
	create_linux_setup.cmd TE Reference Design				
(x) Exit Batch ( (0) Module selec (1) Create minim (2) Create maxim (3) (internal or (g) Install Boar (a) Start desig	ntation (Web Documentation) nothing is done!) tion guide, project creation um setup of CMD-Files and exit Batc um setup of CMD-Files and exit Batc ly) Dev d Files from Xilinx Board Store (be n with unsupported Vivado Version ( module selection guide):	:h eta)			



6/61

Select variant 3 from the selection guide, press enter and agree selection

Create Vivado Project with option 1

Image: mark of the constraint
For better table view please resize windows to full screen!
Select Module will be done in 2 steps:
Step 1: (select column filter): -Change module list size (for small monitors only), press: 'full' or 'small' -Display current module list, press: 'L' or 'l' -Restore whole module list, press: 'R' or 'r' -Reduce List by ID, press: 'ID' or 'id' or insert ID columns value directly(filt er step is bypassed and id number is used)
-Reduce List by Article Number, press: 'AN' or 'an' -Reduce List by SoC/FPGA, press: 'FPGA' or 'fpga' -Reduce List by PCB REV, press: 'PCB' or 'pcb' -Reduce List by DDR, press: 'DDR' or 'ddr' -Reduce List by Flash, press: 'FLASH' or 'flash'
-Reduce List by EMMC, press: 'EMMC' or 'emmc' -Reduce List by Others, press: 'OTHERS' or 'others' -Reduce List by Notes, press: 'NOTES' or 'notes' -Exit without selection, press: 'Q' or 'q'
Please Enter Option: 3

7/61



https://sp.utia.cas..cz

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

devel@ubuntu: ~/work/t	:e0821_3_240/t	est_board	Q ≡	_ 0 (	8
:					
SoC/FPGA Ty  Notes		Flash 			PC 5
E31KA  xczu3cg-sfv			3cg_1e_4gb  64GB	  NA	RE
NA		ا 			
ke to do?		0			
	:  SoC/FPGA Ty  Notes E31KA  xczu3cg-sfv  NA with this device? y ke to do? delivery binary fol	:  SoC/FPGA Typ  DDR Size  Notes E31KA  xczu3cg-sfvc784-1-e  4GB  NA with this device? y/N ke to do? delivery binary folder, press	:  SoC/FPGA Typ  DDR Size  Flash  Notes   E31KA  xczu3cg-sfvc784-1-e  4GB  128MB  NA   with this device? y/N ke to do? delivery binary folder, press 0	:  SoC/FPGA Typ  SHORT DIR  DDR Size  Flash Size EMMC Siz  Notes   E31KA  xczu3cg-sfvc784-1-e  3cg_1e_4gb  4GB  128MB  64GB  NA   with this device? y/N ke to do? delivery binary folder, press 0	:  SoC/FPGA Typ  SHORT DIR    DDR Size  Flash Size EMMC Size  Others  Notes   E31KA  xczu3cg-sfvc784-1-e  3cg_1e_4gb    4GB  128MB  64GB  NA  NA   with this device? y/N ke to do? delivery binary folder, press 0

Vivado project will be generated for module with ID=3, name TE0821-01-3AE31KA, device xczu3cg-sfvc784-1-e, 4GB DDR4

HW support for Vitis Extensible Design Flow

# 2.2 Create Extensible Platform HW

This section describes manual creation of extensible platform HW. You can follow it or you can alternatively use the fast track script described in section 2.3.

In Vivado project, click in **Flow Navigator** on **Settings**. In opened Settings window, select **General** in **Project Settings**, select **Project is an extensible Vitis platform**. Click on **OK**.

8/61



5 20_04_Viti	is_2022_2_Petalinux_te0821_03	cg_4gb - VMware Workstation 16 Player							- 🗆 ×
<u>P</u> layer ▼	- 4 🗆 🛛								« 📄
Activities	s 🔥 Vivado 🔻		Feb 12 1	1:58				A (	。 () 👻
		test_board - [/home/devel/wor	rk/te0821_3_240/test	:_board/	vivado/test_board.	xpr] - Vivado 2022	2.2		• ×
	<u>F</u> ile <u>E</u> dit F <u>l</u> ow		Help Q- Quick Acces	ss				F	Ready
			1/2 B	_			= 0	Default Layout	~
•	Flow Navigator 😤		Setting	S			8		? ×
	✓ PROJECT MANAGE	Q-							205
Â	Settings	Project Settings	General Specify values for va	rious seti	ings used throughou	t the design flow.		~	•
	Add Sources	General	These settings apply			3		•	*
?	Language Temp 👎 IP Catalog	Simulation Elaboration Dataflow	Name: Project device:	test_bo	ard R). *SPRT PCB: REV01	(xczu3ca-sfvc784-1	-e)		
	✓ IP INTEGRATOR	Synthesis	Target language:	VHDL			~		
2	Create Block De	Implementation Bitstream	Default library:	xil defau	utlib		0		
_	Open Block Desi	> IP		zusys w			8		
>_	Generate Block	Tool Settings	. –		apper		<u>ن</u>	reg32_0	
		Project	Language Option	s					
	✓ SIMULATION	IP Defaults > Vivado Store	Verilog options	s:	verilog_version=Veri	og 2001		% 16/16 v1.0 (Beta)	
	Run Simulation	Source File	G <u>e</u> nerics/Para	meters:				/io_0	
	✓ RTL ANALYSIS	Display	Loop count:			1,00	0 🗘		×0[0:0]
	> Open Elaborate	Help > Text Editor	_					=8	x0[0:0] x1[0:0] TEST_OUT[0:0]
		3rd Party Simulators	✓ Project is an extension					l'Input/Output)	
	✓ SYNTHESIS	> Colors Selection Rules	Migrate the outp	ut produ	cts to .gen directory s	tructure			
	Run Synthesis	Shortcuts						-	
	> Open Synthesize	> Strategies						?.	
		> Remote Hosts > Window Behavior							
	Run Implementa							-	^_
	> Open Implement								
	✓ PROGRAM AND DEB								
	👫 Generate Bitstre								
	> Open Hardware								
		?		ОК	Cancel	Apply	Re <u>s</u> tore	srcs/sources_	1/bd/
		<							>
		Type a Tcl command here							
	External Port: TEST_OUT								

IP Integrator of project set up as an extensible Vitis platform has an additional Platform Setup window.

### Add multiple clocks and processor system reset IPs

In IP Integrator Diagram Window, right click, select **Add IP** and add **Clocking Wizard** IP **clk\_wiz\_0**. Double-click on the IP to Re-customize IP window. Select Output Clocks panel. Select four clocks with frequency 100, 200, 400 and 240 MHz. 100 MHz clock will serve as low speed clock.

200 MHz and 400 MHz clock will serve as clock for possible AI engine.

240 MHz clock will serve as the default extensible platform clock. By default, Vitis will compile HW IPs with this default clock.

Set reset type from the default Active High to **Active Low**.



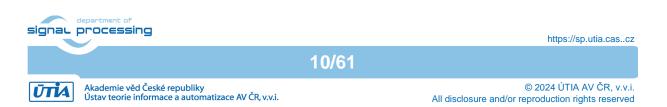
https://sp.utia.cas..cz

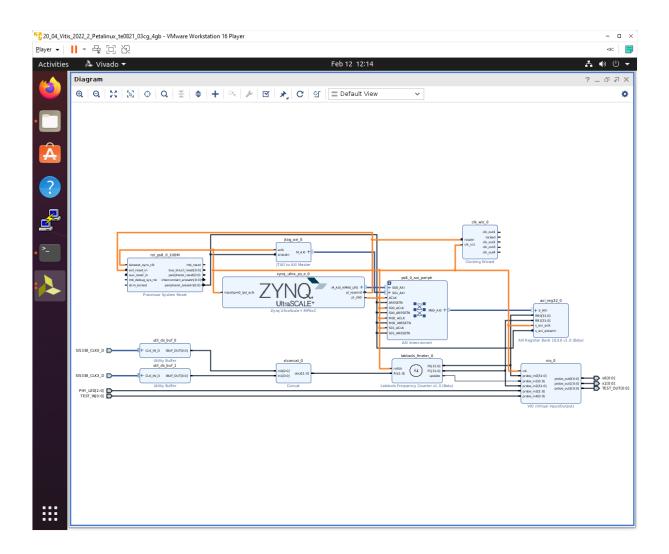
ŪTIA

	- 4 口 2			10.10.10					«
vities	🍌 Vivado 🔻			12 12:12				Å	4) (J
			Re-custo	mize IP				•	× 0
E	Clocking Wizard (6.0)								Ready
Fl	🚯 Documentation 🛛 🗁 IP Location								1
Š									? 🗆
	IP Symbol Resource	Component Name	clk_wiz_0						
	Show disabled ports	Clocking Options	Output Cl	ocko MMCN	1 Setting	s Summary			
		Clocking Options		3 100.000		100.00000	0.000	0.000 ^	
		✓ clk_out2	-	200.000		200.00000		0.000	
		✓ clk_out3	_	400.000	-	400.00000		0.000	
		✓ clk_out4	clk_out4 (	240.000	8	240.00000	0.000 🛞	0.000	
		clk_out5	clk_out5	100.000		N/A	0.000	N/A	
		clk_out6	clk_out6	100.000		N/A	0.000	N/A	_reg32_0
		clk_out7	clk_out7	100.000		N/A	0.000	N/A	
~	clk_out1 = clk_out2 = clk_out3 = clk_out3 =			ce Number					<u>via_0</u>
~		clk_out1	1		]				al Input/Out
Ť		clk_out2	1						
		clk_out3	1						_ 0
		clk_out4	1						
~		clk_out6	1						
		clk_out7	1						1/bd/
								U	
~		Enable Optional	Inputs / Out	puts for MMC	M/PLL	Reset Type		Phase !	
		✓ reset	) power_down	input_clk	stopped	0.4-44	ve Hiah () Active Low	0	858 ;
		<	, power_down	on	_ccopped	Activ	e High ( Active Low	)'* >	
									>

Clik on OK to close the Re-customize IP window.

Connect input **resetn** of **clk\_wiz\_0** with output **pl\_resetn0** of **zynq\_ultra\_ps\_e\_0**. Connect input **clk\_in1** of **clk\_wiz\_0** with output **pl\_clk0** of **zynq\_ultra\_ps\_e\_0**.





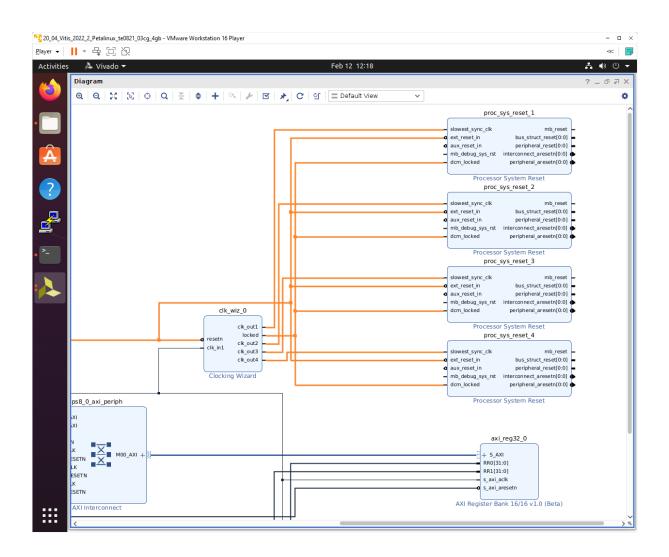
Add and connect four Processor System Reset blocks for each generated clock.

11/61



https://sp.utia.cas..cz

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



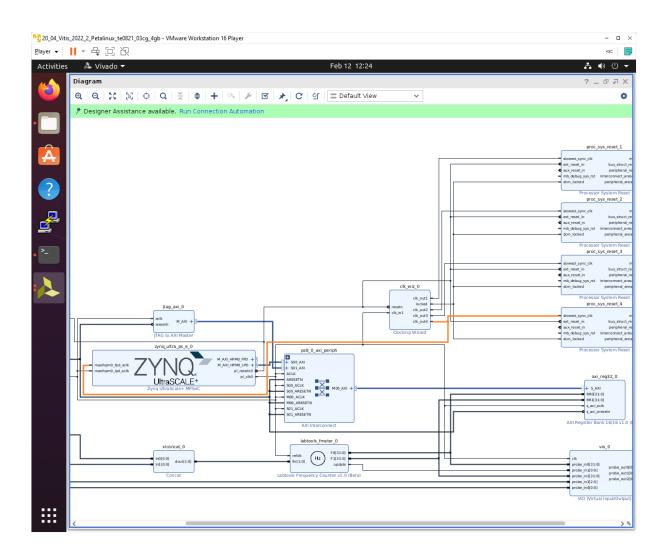
Open Platform Setup window of IP Integrator to define Clocks. In Settings, select Clock.

In "Enabled" column select all four defined clocks clk\_out1, clk\_out2, clk\_out3, clk\_out4 of clk\_wiz\_0 block.

In "ID" column keep the default Clock ID: 1, 2, 3, 4

In "Is Default" column, select **clk\_out4** (with ID=4) as the default clock. One and only one clock must be selected as default clock.





Double-click on **zynq\_ultra\_ps\_e\_0** block and enable **M\_AXI\_HPM0\_FPD** port. Select data width 32bit. It will be used for integration of inerrupt controller on new dedicated AXI stream subsystem with 240 MHz clock. It will also enable new input

pin maxihpm0\_fpd\_aclk of zynq\_ultra\_ps\_e\_0. Connect it to 240 MHz clock net.

Connect input pin **maxihpm0\_fpd\_aclk** of **zynq\_ultra\_ps\_e\_0** to the 240 MHz **clk\_out4** of **clk\_wiz\_0** IP block.

### Add, customize and connect the AXI Interrupt Controller

Add AXI Interrupt Controller IP **axi\_intc\_0**. Double-click on **axi\_intc\_0** to re-customize it.

In "Processor Interrupt Type and Connection" section select the "Interrupt Output Connection" from "Bus" to "Single".

Click on OK to accept these changes.



13/61

https://sp.utia.cas..cz

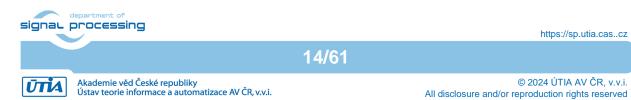
A

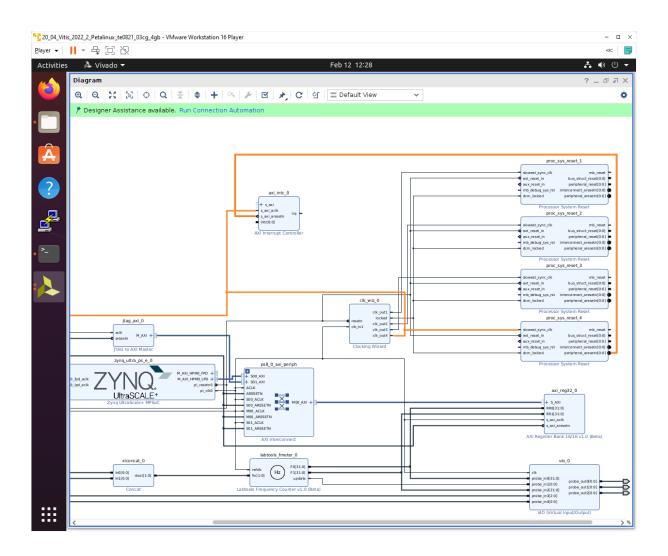
5 20_04_Vit	is_2022_2_Petalinux_te0821_	03cg_4gb - VMware Workstation 16 Player			-	- 🗆 ×
<u>P</u> layer ▼	- 4 2 2				<	« 📑
Activitie	s 🔥 Vivado 🔫		Feb 12 12:25		<b>.</b> ♠	७ ◄
-		test_board - [/home/dev	vel/work/te0821_3_240/test_board/vivado/test_board.xpr] - Vivado 2022.2			
	<u>F</u> ile <u>E</u> dit Flow		Re-customize IP		Re	ady
				ault I	Layout	~
•	Flow Navigator	AXI Interrupt Controller (4.1)	4			? ×
	✓ PROJECT MANAG	Occumentation IP Location			?	06
Â	Settings					_
	Add Sources	Show disabled ports	Component Name axi_intc_0			
?	Language Tem					
	후 IP Catalog		Basic Advanced Clocks	ult	Proc Sy	Sta
	✓ IP INTEGRATOR		Interrupt Usage			
	Create Block D		Number of Peripheral Interrupts (Auto)			- 11
	Open Block De		Fast Interrupt Mode			
• >_	Generate Block		Enable Fast Interrupt Logic			-11
	Export Platform		Interrupt Vector Address reset value (Auto) 0x00000000000000000000000000000000000		/proc_s	fixe
					/proc_s	fixe
	Run Simulation		Peripheral Interrupts Type		/proc_s	fixe fixe
	✓ RTL ANALYSIS	+ s_axi	Auro Interrupts type - Edge or Level 0xFFFFFFF 0	Ð		×
	<ul> <li>Open Elaboration</li> </ul>	– s_axi_aclk irq – • s_axi_aresetn	Auro Level type - High or Low 0xFFFFFFF 0			
	, open Lieber et	- intr[0:0]				
	✓ SYNTHESIS		Auro Edge type - Rising or Falling 0xFFFFFFFF 0			_
	Run Synthesis				? _	
	> Open Synthesi:					
			Processor Interrupt Type and Connection	100	000000"} c	lk_^
	Run Implement		Interrupt type Level Interrupt V	1000	500000 F C	.uk_
	> Open Implement		Level type Active High 🗸			
	✓ PROGRAM AND DEI		Interrupt Output Connection Single V			
	↓ <sup>III</sup> Generate Bitst					
	> Open Hardware		<			
						~
			OK Cancel			>
•••	L					

Connect interrupt controller clock input **s\_axi\_aclk** of **axi\_intc\_0** to output **dlk\_out4** of **clk\_wiz\_0**. It is the default, 240 MHz clock of the extensible platform.

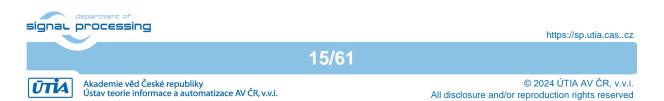
### Connect interrupt controller input **s\_axi\_aresetn** of **axi\_intc\_0** to

output **peripheral\_aresetn[0:0]** of **proc\_sys\_reset\_4**. It is the reset block for default, 240 MHz clock of the extensible platform.





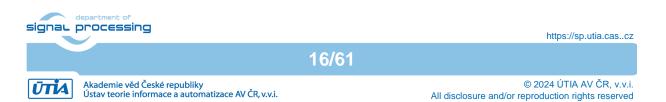
Use the **Run Connection Automation** wizard to connect the axi lite interface of interrupt controller **axi\_intc\_0** to master interface **M\_AXI\_HPM0\_FPD** of **zynq\_ultra\_ps\_e\_0**.

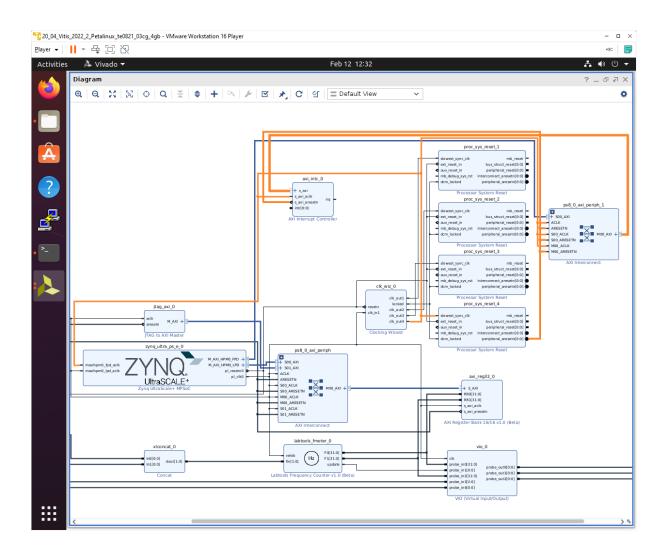


▼      ▼ 母 [				<ul> <li></li> </ul>
vities 🍌 Viva		Feb 12 12:30		
	test_board - [/	'home/devel/work/te0821_3_240/test_bo	oard/vivado/test_board.xpr] - Vivado 2022.2	_ 0
<u>Eile</u> dit	: F <u>l</u> ow <u>T</u> ools Rep <u>o</u> rts <u>W</u> indow	Layout View Help Q- Quick Access		Ready
	! ■ ! ■ ! X ! @ ! <b>⊠</b> ! ▶			🗮 Default Layout
Flow Navig				
✓ PROJECT	MANAGER	T F( Address Editor y Address M	an	? 🗆
Set Add		Run Connection Auto	omation	
	Automatically make connections in your d configuration options on the right.	lesign by checking the boxes of the interfaces	to connect. Select an interface on the left to displa	yits 🍌
	Q   素   ≑	Description		y S
Crea	All Automation (1 out of 2 selecte v v * axi_intc_0	d) Connect Slave interface (/axi_intc_0/	's_axi) to a selected Master address space.	
Ope	v ⊕ <mark>s_axi</mark>	Options		
Ger	✓ □ ≢ zynq_ultra_ps_e_0 □ ⊕ M AXI HPM0 FPD	Master interface	/zyng ultra ps e 0/M AXI HPM0 FPD ∨	
Ехр		Bridge IP	New AXI Interconnect	s fi
				s fi
V SIMULA		Clock source for driving Bridge IP	/clk_wiz_0/clk_out4 (240 MHz) ↔	s fi
			/clk_wiz_0/clk_out4 (240 MHz) ✓	s fi
Y RTL AN		Clock source for Master interface	/clk_wiz_0/clk_out4 (240 MHz) ∨	
> Ope				
✓ SYNTHE				
Run				
> Ope				_ □
✓ IMPLEM				
> Ope				
/ Ope	(?)		0/	Const
✓ PROGR.	(F)		ОК	Cancel
👫 Gener		<pre>IRQ_CONNECTION {1} [get_bd_cells axi_int</pre>		
> Open	Hardware M connect_bd_net [get_bd connect_bd_net [get_bd	d_pins axi_intc_0/s_axi_aclk] [get_bd_pin d_pins axi_intc_0/s_axi_aresetn] [get_bd_	s clk_wiz_0/clk_out4] pins proc_sys_reset_4/peripheral_aresetn]	
				>
:	Type a Tcl command here			,

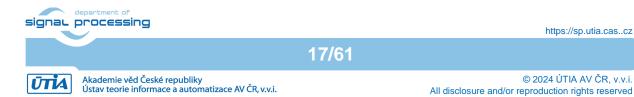
In Run Connection Automaton window, click OK.

New AXI interconnect **ps\_8\_axi\_periph** is created. It connects master interface **M\_AXI\_HPM0\_FPD** of **zynq\_ultra\_ps\_e\_0** with interrupt controller **axi\_intc\_0**.





Double-click on **zynq\_ultra\_ps\_e\_0** to re-customize it by enabling of an interrupt input **pl\_ps\_irq0[0:0]**. Click OK.



5 20_04_Vit	is_2022_2_Petalinux_te0821_	_03cg_4gb - VMware Worksta	ion 16 Player	- • ×
<u>P</u> layer ▼	- 中口 2			≪ 📑
Activities	s 🔥 Vivado 🔻		Feb 12 12:33	<b>♣ ● Ů ▼</b>
4		test_bo	ard - [/home/devel/work/te0821_3_240/test_board/vivado/test_board.xpr] - Vivado 2022.2	_ 0 ×
	<u>F</u> ile <u>E</u> dit F <u>l</u> ow		Re-customize IP	Ready
				ault Layout 🗸 🗸
•	Flow Navigator	Zynq UltraScale+	MPSoC (3.4)	? ×
	✓ PROJECT MANAG	<b>0 *</b> .		205
Â	Settings	Documentation I I Documentation	Presets 📄 IP Location	200
	Add Sources	Page Navigator	PS-PL Configuration	
?	Language Tem		← Q X +	
	후 IP Catalog	Switch To Advan		ult Proc Sy Sta
2	✓ IP INTEGRATOR	PS UltraScale+ Bloc	Search:         Q-           Name         Select	^
2	Create Block D	I/O Configuration	Name Select V General	
-	Open Block De	clash Carfinunatian	✓ Interrupts	
>_	Generate Block	Clock Configuration	V PL to PS	
	Export Platform	DDR Configuration	IRQ0[0-7] 1 ~	
		PS-PL Configuration		/proc_s fixe /proc_s fixe
	✓ SIMULATION		APU Legacy Interrupts(IRQ,  RPU Legacy Interrupts(IRQ,	/proc_s fixe
	Run Simulation		> PS to PL	/proc_s fixe
	✓ RTL ANALYSIS		> Fabric Reset Enable	>
	> Open Elaborat		> Address Fragmentation	
			Others     PS-PL Interfaces	
	✓ SYNTHESIS		> PS-L Interfaces	-
	Run Synthesis		,	? _ 0 6
	> Open Synthesi:			
	✓ IMPLEMENTATION			^
	Run Implement			
	> Open Implemer			
	✓ PROGRAM AND DEI			
	<ul> <li>Generate Bitst</li> <li>Open Hardware</li> </ul>			wiz_0/clk_out4 (
	y open hardware			64K ]>.
			OK Cancel	>

Modify the automatically generated reset network of AXI interconnect ps\_8\_axi\_periph .

Disconnect input **S00\_ARESETN** of **ps\_8\_axi\_periph** from the network driven by output peripherial\_aresetn[0:0] of proc\_sys\_reset\_4 block.

Connect input **S00\_ARESETN** of **ps\_8\_axi\_periph** block with output interconnect\_aresetn[0:0] of proc\_sys\_reset\_4 block.

Disconnect input M00\_ARESETN of ps\_8\_axi\_periph block from the network driven by output peripherial\_aresetn[0:0] of proc\_sys\_reset\_4 block.

Connect input MO0\_ARESETN of ps\_8\_axi\_periph to output interconnect\_aresetn[0:0] of proc\_sys\_reset\_4 block.

This modification will make the reset structure of the AXI interconnect **ps\_8\_axi\_periph** block identical to the future extensions of this interconnect generated by the Vitis extensible design flow.

Connect the interrupt input pl\_ps\_irq0[0:0] of zynq\_ultra\_ps\_e\_0 block with output irq of axi\_intc\_0 block.

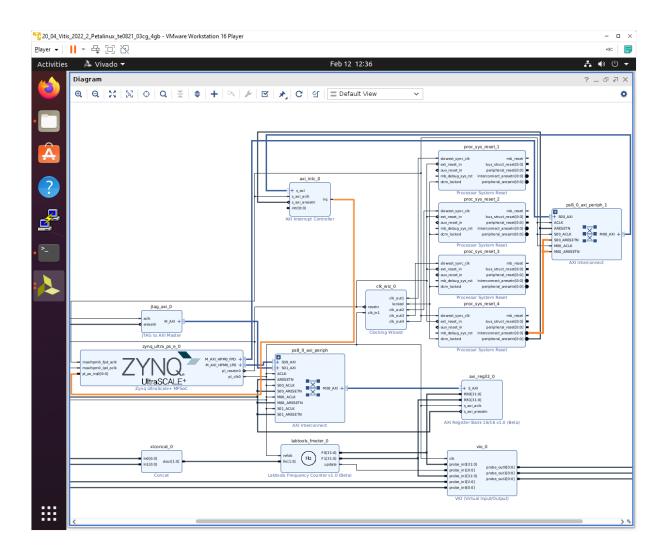


https://sp.utia.cas..cz

18/61



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



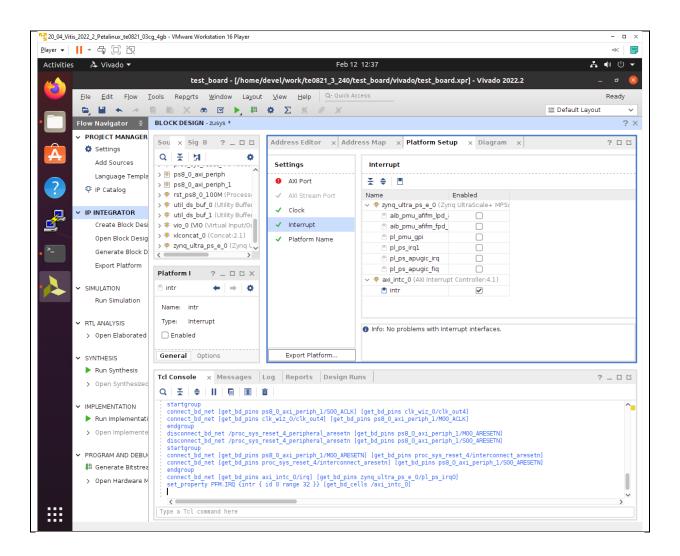
19/61

In Platform Setup, select "Interrupt" and enable intr in the "Enabled" column.



https://sp.utia.cas..cz

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



Rename automatically generated name **ps8\_0\_axi\_periph** of the interconnect to new name: **axi\_interconnect\_1**. This new name will be used in Platform Setup selection of AXI ports for the extensible platform.

In Platform Setup, select AXI Ports for zynq\_ultra\_ps\_e\_0:

SelecM\_AXI\_HPM1\_FPD in column "Enabled".

Select **S\_AXI\_HPC0\_FPD** and **S\_AXI\_HPC1\_FPD** in column "Enabled".

For **S\_AXI\_HPC0\_FPD**, change S\_AXI\_HPC to **S\_AXI\_HP** in column "Memport".

For **S\_AXI\_HPC1\_FPD**, change S\_AXI\_HPC to **S\_AXI\_HP** in column "Memport".

Select S\_AXI\_HP0\_FPD, S\_AXI\_HP1\_FPD, S\_AXI\_HP2\_FPD, S\_AXI\_HP3\_FPD in column "Enabled".

Type into the "sptag" column the names for these 6 interfaces so that they can be selected by v++ configuration during linking phase. **HPC0**, **HPC1**, **HP0**, **HP1**, **HP2**, **HP3** 





https://sp.utia.cas..cz

Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.  $\hfill \mbox{\sc c}$  2024  $\mbox{\sc UTIA}$  AV  $\mbox{\sc CR},$  v.v.i. All disclosure and/or reproduction rights reserved

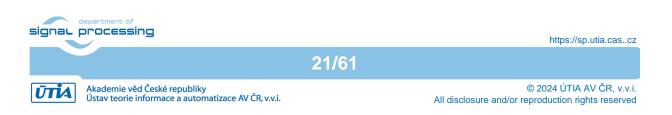
	is_2022_2_Petalinux_te0821_03c									-	- <
tivities	s 🍌 Vivado 🔫			Feb 1	2 12:42					<b>.</b> ♠	0 -
		test_l	board - [/home/devel/wor	k/te0821_3_240/	test_board/viv	ado/test_board	.xpr] - Vivado 202	2.2			o 🜔
2	<u>F</u> ile <u>E</u> dit F <u>l</u> ow ]	ools Rep <u>o</u> rts	<u>W</u> indow Layout ⊻iew	Help Q- Quick /	Access					Rea	ady
	🕞 🔛 👞 🗼 🗆		🗹 🕨 👬 🔅 Σ	<u>%</u>					📰 Default	Layout	~
	Flow Navigator 😤	BLOCK DESIGN -	zusys *								?
	× PROJECT MANAGER										
Â	🍄 Settings	? _ 0 6	Audress Editor X   Au	ness hap X	-lationin Setup	X Diagram	~			£	
	Add Sources	Q ≚ 対 » ≢ labtools fn ∧	Settings	AXI Por	t						
	Language Templa	<pre># nabtools_nix</pre>	✓ AXI Port	¥ \$							
?	👎 IP Catalog	proc_sys_r									
		🌻 proc_sys_r	🖌 🛛 AXI Stream Port	Name	ultra ns e 0 (7	Enabled ynq UltraScale+ 1	Memport MPSoC(3,4)		SP Tag	Memory	
2,	✓ IP INTEGRATOR	<pre>proc_sys_r</pre>	<ul> <li>Clock</li> </ul>		AXI HPM1 FPD		M AXI GP	~	~		
2	Create Block Desi	ps8_0_axi_	<ul> <li>Interrupt</li> </ul>		AXI HPC0 FPD		S AXI HP	~	HPC0 V		
	Open Block Desig	ps8_0_axi_ # rst_ps8_0_	<ul> <li>Platform Name</li> </ul>		AXI_HPC1_FPD		S_AXI_HP	~	HPC1 V		
	Generate Block D	<pre>util_ds_buf</pre>			AXI_HP0_FPD		S_AXI_HP	~	HP0 🗸		
		util_ds_buf		🔳 S	AXI_HP1_FPD		S_AXI_HP	~	HP1 v		
	Export Platform	🌻 vio 0 (V10 (🗸		🖻 S	AXI_HP2_FPD		S_AXI_HP	~	HP2 ∨		
		<>		🖪 S	AXI_HP3_FPD		S_AXI_HP	~	HP3 🗸		
	<ul> <li>SIMULATION</li> </ul>	? _ O Ľ X		🗏 S	_AXI_LPD						
	Run Simulation			∨ 🖲 ps8	_0_axi_interconne	ect_1 (AXI Interco	nnect:2.1)				
					01_AXI						
	<ul> <li>RTL ANALYSIS</li> </ul>	Name: S AXI			02_AXI						
	> Open Elaborated	-			03_AXI						
		Type: AXI P		S	04 AXI						
	<ul> <li>SYNTHESIS</li> </ul>	🕑 Enabled		🚯 Info: No	problems with A	XI Port interfaces					
	Run Synthesis										
	> Open Synthesized										
		Gener (→ ≡)	Export Platform								
	✓ IMPLEMENTATION										
	🕨 Run Implementati	Tcl Console 🔿	Messages Log Rep	oorts Design R	luns					? _	
	> Open Implemente	0, ₹ ≑									
		set property	PFM.AXI PORT {M AXI HPM1	FPD {memport "M #	XI GP" sptag ""	memory "" is r	ange "false"} S A	KI HPCO F	PD {mempor	t "S AXI HP	" s^
	✓ PROGRAM AND DEBU	set_property	PFM.AXI_PORT {M_AXI_HPM1_	FPD {memport "M_4	XI_GP" sptag ""	memory "" is_r	ange "false"} S_A	KI_HPC0_F	PD {mempor	t "S_AXI_HP	" s
	👫 Generate Bitstrea	set_property	PFM.AXI_PORT {M_AXI_HPM1_ PFM.AXI_PORT {M_AXI_HPM1_	FPD {memport "M_A	XI GP" sptag ""	memory "" is r	ange "false"} S A	KI HPCO F	PD {mempor	t "S AXI HP	" s
	> Open Hardware M	: set property	PFM.AXI_PORT {M_AXI_HPM1_ PFM.AXI PORT {M_AXI_HPM1_	=PD {memport "M A	XI GP" sptag ""	memory "" is r	ange "false"} S A)	KI HPCO F	PD {mempor	t "S AXI HP	" se
	y open naroware i	set_property	PFM. AXI_PORT {M_AXI_HPM1_ PFM. AXI_PORT {M_AXI_HPM1_	FPD {memport "M_A	XI_GP" sptag ""	memory is_r	ange "false"} S_A)	(I_HPC0_F	PD {mempor	t "S_AXI_HP	
		<									>
		Type a Tcl comm	and here								

In "Platform Setup", select AXI Ports for the recently renamed axi\_interconnect\_1:

Select M01\_AXI, M02\_AXI, M03\_AXI, M04\_AXI, M05\_AXI, M06\_AXI and M07\_AXI in column "Enabled".

Make sure, that you are selecting these AXI ports for the 240 MHz AXI interconnect **axi\_interconnect\_1** 

Keep all AXI ports of the 100 MHz interconnect axi\_interconnect\_0 unselected. The AXI interconnect axi\_interconnect\_0 connects other logic and IPs which are part of the initial design.



- ↓ ↓ → □										<	×
	vado <del>-</del>			Feb 1	12 12:44					<b>.</b> ↓)	<del>ا</del> ل ال
		test l	board - [/home/devel/wor	k/te0821 3 240/	/test board	/vivado/test_board	.xpr] - Vivado 2022	.2			-
Eile Ed	lit Flow To			Help Q- Quick						Rea	adv
				M W X					Default		1
	•							=	Derault	Layout	~
Flow Navi	5	BLOCK DESIGN -:	zusys								?
✓ PROJEC	T MANAGER	? _ 🗆 🖒	Address Editor × Add	dress Map 🛛 🗙	Platform Se	tup x Diagram	×			?	0 6
PROJEC     PROJEC     Setti	ings	Q.풒.鴙.»					1				
Add	Sources	Iabtools fn ∧	Settings	AXI Por	t						
Lang		<pre>proc sys r</pre>	🖌 AXI Port	<b>•</b>							
Lang 후 IP Ca		<pre>proc_sys_r</pre>		¥ \$							
V IP INTE	~	proc_sys_r	<ul> <li>AXI Stream Port</li> </ul>	Name	10.00	Enabled	Memport	SP	Tag	Memory	
	GRATOR	proc_sys_r	<ul> <li>Clock</li> </ul>		12_AXI						^
Cros	te Block Desi	🖲 ps8_0_ахі_	✓ Interrupt		13_AXI						
		🍯 ps8_0_ахі_	<ul> <li>Incerrupt</li> </ul>		14_AXI						
Oper		rst_ps8_0_	<ul> <li>Platform Name</li> </ul>		15_AXI		11.110.00				
Gene	share brock b	👎 util_ds_buf			101_AXI	<ul> <li>Image: A start of the start of</li></ul>	M_AXI_GP	~	~		
Expo		util_ds_buf			102_AXI	Solution	M_AXI_GP	~	~		
					103_AXI		M_AXI_GP	~	~		
✓ SIMULAT					104_AXI	<ul> <li>✓</li> <li>✓</li> </ul>	M_AXI_GP	~	~		- 1
		? _ 🗆 🖒 ×			105_AXI	<ul> <li>Image: Second sec</li></ul>	M_AXI_GP		~		
Kull		⊜ ←   →   ☆			106_AXI	<ul> <li>Image: A start of the start of</li></ul>	M_AXI_GP	~	~		
					107_AXI 108 AXI		M_AXI_GP	~	×		4
✓ RTL ANA		Name: M07			108_AXI 109 AXI						
> Oper	n Elaborated	Type: AXI P			109_AXI						-
				[v	ITO ANI						
✓ SYNTHE	SIS	Enabled		Info: No	problems wi	th AXI Port interfaces					
🕨 🕨 Run	Synthesis										
> Oper	n Synthesized										
		Gener 4 → Ξ	Export Platform								
► Run	Run Implementati     Tcl Console x Messages Log Reports Design Runs									? _	0 6
	n Implemente	0, ≭ ♦									
			PFM. AXI PORT {MO1 AXI {men	mount "M AVT GP"	enter "" me	mory "" is range "f	alco"} MO2 AXT {man	nort "M AYT	GP" on	ton "" mem	
✓ PROGRA	M AND DEBU	set_property	PFM.AXI PORT {MO1 AXI {men	mport "M_AXI_GP"	sptag "" me	mory "" is range "f	alse"} MO2_AXI {mem	port "M_AXI	GP" sp	tag "" memo	ory
		set_property ⊖ save bd desid	PFM.AXI_PORT {M01_AXI {men	mport "M_AXI_GP"	sptag "" me	mory "" is_range "f	alse"} MO2_AXI {men	port "M_AXI	_GP" sp	tag "" memo	ory
- Gene		Wrote : <td>ome/devel/work/te0821_3_24</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	ome/devel/work/te0821_3_24								
	n Hardware M	⊖ Wrote :	ome/devel/work/te0821_3_24	0/test board/viva	ado/test boa	rd.srcs/sources_1/b	d/zusys/ui/bd_ec757	75b2.ui>			
> Oper				_							
> Oper				-							>

The modifications of the default design for the extensible platform are completed, now.

In Vivado, save block design by clicking on icon "Save Block Design".

To continue the manual design path, go to section 2.4 Validate design.

# 2.3 Fast Track for Creation of Extensible platform HW

HW modifications can be made by sourcing this script in Vivado with open diagram in IP Integrator.

Copy file from the accompanying support package

te0821\_AI\_3\_0\_eval\_package\vivado\script\_te0821.txt

to

~/work/te0821\_3\_240/test\_board/vivado/script\_te0821.txt

Execute in Vivado Tcl console this command:

source script\_te0821.txt

signal processing

22/61



© 2024 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

# 2.4 Validate Design

Results of HW creation via Manual Track or Fast Track are identical.

Open diagram by clicking on zusys.bd if not already open.

In Diagram window, validate design by clicking on "Validate Design" icon.

20\_04\_Vitis\_2022\_2\_Petalinux\_te0821\_03cg\_4gb - VMware Workstation 16 Player - 🗆 × Player - 📔 - 🖶 🖸 🕅 ~ ٢ Activities 🛛 🍌 Vivado 🔻 Feb 12 12:45 **(**)) test\_board - [/home/devel/work/te0821\_3\_240/test\_board/vivado/test\_board.xpr] - Vivado 2022.2 × Eile Edit Flow Tools Reports Window Layout View Help Q-Quick Access Ready 💾 🔸 🛹 6 X 💷 Default Lavout Flow Navigator 😤 BLOCK DESIGN - zusys \* ? × ✓ PROJECT Critical Messages m Setup x Diagram x 205 🖨 Setti Δ ビ 
 C 
 G
 Default View
 ~ ø Add § There was one critical warning message while validating this design. Langi Messages (B) 41-759) The input pins (listed below) are either not connected or do not have a source port, and they don't have a tie-off specified. These pins are tied-off to all 0's to avoid error in Implementation flow.
 Please check your design and connect them as needed: /axi\_intc\_0/intr ₽ IP Cat P IP INTEG Creat -Open Gene Expor Open Messages View SIMULATI Run S + → ☆ RTL ANALYSIS Name: M07 ZYNQ  $\frac{\overline{X}}{\overline{X}}$ > Open Elaborated AXI P Type: Enabled SYNTHESIS Run Synthesis • > Open Synthesiz NY LEE 2.0) Gener∢ ▶ ≣ IMPLEMENTATION Tcl Console x Messages Log Reports Design Runs ? \_ 🗆 🖸 🕨 Run Implementati > Open Implemente Q, 🛣 💠 Ⅱ 🗉 🔟 WANING: [B0 41-237] Bus Interface property ARUSER WIDTH does not match between /ps8\_0\_axi\_periph/s00\_couplers/auto\_pc/S\_AXI(0) and /zyA\_ CRITICAL WARNING: [B0 41-759] The input pins (listed below) are either not connected or do not have a source port, and they don't have Please check your design and connect them as needed: /axi\_intc\_D/intr PROGRAM AND DEBU 👫 Generate Bitstrea validate\_bd\_design: Time (s): cpu = 00:00:08 ; elapsed = 00:00:07 . Memory (MB): peak = 10320.215 ; gain = 0.000 ; free physical = 4825 > Open Hardware M < Type a Tcl command here

Received 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.

Click OK.

Save design.

You can generate pdf of the block diagram by clicking to any place in diagram window and selecting "Save as PDF File". Use the offered default file name:

```
~/work/TE0821_3_240/test_board/vivado/zusys.pdf
```







# 2.5 Compile Created HW and Custom SW with Trenz Scripts

In Vivado Tcl Console, type following script and execute it by Enter. It will take some time to compile HW. HW design and to export the corresponding standard XSA package with included bitstream.

```
TE::hw build design -export prebuilt
```

An archive for standard non-extensible system is created:

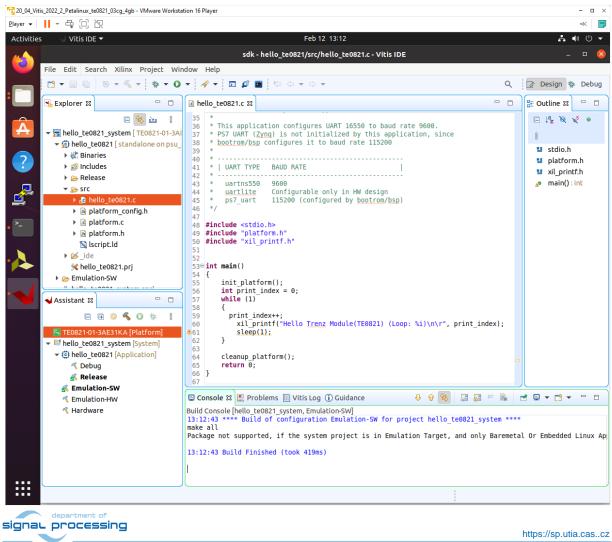
~/work/TE0821 3 240/test board/vivado/test board 3cg 1e 4gb.xsa

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 TE0821-01-3AE31KA has been created and compiled.



24/61



The TE0821-01-3AE31KA Vitis platform includes Trenz Electronic custom first stage boot loader in folder zynqmp\_fsb1. It includes SW extension specific for the Trenz module initialisation.

This custom zynqmp\_fsbl project has been compiled into executable file fsbl.elf. It is located in:

~/work/TE0821 3 240/test board/prebuilt/software/3cg 1e 4gb/fsbl.elf

This customised first stage boot loader is needed for the Vitis extensible platform.

We have used the standard Trenz scripts to generate it for next use in the extensible platform.

Exit the opened Vitis SDK project.

In Vivado top menu select File->Close Project to close project. Click OK.

In Vivado top menu select File->Exit to close Vivado. Click OK.

The exported Vitis Extensible Hardware platform named test\_board\_3cg\_1e\_4gb.xsa can be found in the vivado folder.

### 2.6 Copy Created Custom First Stage Boot Loader

Up to now, test\_board directory has been used for all development.

```
~/work/TE0821 3 240/test board
```

Create new folders:

~/work/TE0821\_3\_240/test\_board\_pfm/pfm/boot ~/work/TE0821 3 240/test board pfm/pfm/sd dir

Copy the recently created custom first stage boot loader executable file from

~/work/TE0821\_3\_240/test\_board/prebuilt/software/3cg\_1e\_4gb/fsbl.elf

to

```
~/work/TE0821 3 240/test board pfm/pfm/boot/fsbl.elf
```

# 3 Building Petalinux for Extensible Design Flow with Vitis AI 3.0 Support

### 3.1 Vitis AI 3.0 support

Download the Vitis-Al 3.0 repository. In browser, open page:

https://github.com/Xilinx/Vitis-AI/tree/3.0

signal processing

25/61

https://sp.utia.cas..cz



# Click on green Code button and download Vitis-AI-3.0.zip file. Unzip

```
Vitis-AI-3.0.zip
```

#### to directory

~/Downloads/Vitis-AI

#### Copy

~/Downloads/Vitis-AI

#### to

~/work/Vitis-AI-3.0

#### The directory

~/work/Vitis-AI-3.0

contains the Vitis-AI 3.0 framework, now.

To install the Vitis-AI 3.0 version of shared libraries into rootfs (when generating system image by PetaLinux) we have to copy recepies recipes-vitis-ai to the Petalinux project.

#### Copy

~/work/Vitis-AI-3.0/src/vai petalinux recepies/recipes-vitis-ai

#### to

```
~/work/te0802_04_240_vga/test_board/os/petalinux/project-spec/meta-
user/
```

#### Delete file:

```
~/work/te0802_04_240_vga/test_board/os/petalinux/project-spec/meta-
user/recipes-vitis-ai/vart/vart 3.0 vivado.bb
```

#### and keep only the unmodified file:

```
~/work/te0802_04_240_vga/test_board/os/petalinux/project-spec/meta-
user/recipes-vitis-ai/vart/vart_3.0.bb
```

File vart\_3.0.bb will create vart libraries for Vitis design flow with dependency on the AMD xrt software framework.

# 3.2 Building Petalinux for Extensible Design Flow

Change directory to the default Trenz Petalinux folder

```
~/work/TE0821 3 240/test board/os/petalinux
```

Source Vitis and Petalinux scripts to set environment for access to Vitis and PetaLinux tools.

```
$ source /tools/Xilinx/Vitis/2022.2/settings64.sh
```

```
$ source ~/petalinux/2022.2/settings.sh
```



Configure petalinux with the test\_board\_3cg\_4gb.xsa for the extensible design flow by executing:

\$ petalinux-config --get-hw-description=

~/work/TE0821\_3\_240/test\_board/vivado

🕫 devel@ubuntu: ~/work/te0820_84_240/test_board/os/petali 🔾 😑 🗕 🛛 🧕
/home/devel/work/te0820_84_240/test_board/os/petalinux/project-spec/configs/co
<pre>misc/config System Configuration Arrow keys navigate the menu. <enter> selects submenus&gt; (or empty submenus). Highlighted letters are hotkeys. Pressing <y> includes, <n> excludes, <m> modularizes features. Press <esc><esc> to exit, <?> for Help,  for Search. Legend: [*] built-in []</esc></esc></m></n></y></enter></pre>
-*- XYNQMP Configuration Linux Components Selection> Auto Config Settings> -*- Subsystem AUTO Hardware Settings> DTG Settings> PMUFW Configuration> FSBL Configuration> ARM Trusted Firmware Configuration> FPGA Manager> u-boot Configuration> Linux Configuration> Image Packaging Configuration> Firmware Version Configuration> Yocto Settings>
<pre></pre>

Select Exit->Yes to close this window.

In text editor, modify the user-rootfsconfig file:

```
~/work/TE0821_86_240/test_board/os/petalinux/project-spec/meta-
user/conf/user-rootfsconfig
```

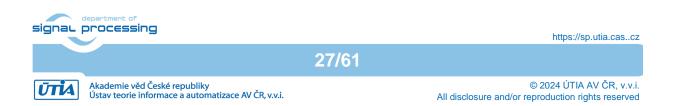
In text editor, append these lines:

#Note: Mention Each package in individual line

#These packages will get added into rootfs menu entry

CONFIG\_startup

CONFIG\_webfwu



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 packagegroup-petalinux-vitis-acceleration

CONFIG\_packagegroup-petalinux-vitis-acceleration-dev

CONFIG vitis-ai-library

CONFIG vitis-ai-library-dev

CONFIG vitis-ai-library-dbg

xrt, xrt-dev and zocl are required for Vitis acceleration flow. dnf is for package management. parted, e2fsprogs-resize2fs and resize-part can be used for ext4 partition resize.

Other included packages serve for natively building Vitis AI applications on target board and for running Vitis-AI demo applications with GUI.

28/61

The last three packages will enable use of the Vitis-AI 3.0 recepies for installation of the correspoding Vitis-AI 3.0 libraries into rootfs of PetaLinux.

Launch rootfs config:

\$ petalinux-config -c rootfs

signal processing



© 2024 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

All packages will have to have an asterisk [\*].

Only vitis-ai-library-dev and vitis-ai-library-dbg will stay indicated as unselected by: [ ].

Still in the RootFS configuration window, go to root directory by select Exit once.

### Enable OpenSSH and Disable Dropbear

Dropbear is the default SSH tool in Vitis Base Embedded Platform. If OpenSSH is used to replace Dropbear, the system could achieve faster data transmission speed over ssh. Created Vitis extensible platform applications may use remote display feature. Using of OpenSSH can improve the display experience.

Go to Image Features.

Disable ssh-server-dropbear and enable ssh-server-openssh and click Exit once.

Go to Filesystem Packages->misc->packagegroup-core-ssh-dropbear and disable packagegroup-core-ssh-dropbear.

Go to Filesystem Packages level by Exit twice.

Go to console->network->openssh and enable openssh, openssh-sftpserver, openssh-sshd, openssh-scp.

Go to root level by selection of Exit four times.

### Enable Package Management

Package management feature can allow the board to install and upgrade software packages on the fly.

In rootfs config go to Image Features and enable

package management and debug\_tweaks options. Click OK, Exit twice and select Yes to save the changes.

# 3.3 Disable CPU IDLE in Kernel Config

CPU IDLE would cause processors get into IDLE state (WFI) when the processor is not in use. When JTAG is connected, the hardware server on host machine talks to the processor regularly. If it talks to a processor in IDLE status, 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.

Launch kernel config:

\$ petalinux-config -c kernel

Ensure the following items are TURNED OFF by entering 'n' in the [] menu selection:

CPU Power Management->CPU Idle->CPU idle PM support

CPU Power Management->CPU Frequency scaling->CPU Frequency scaling

29/61

Exit and Yes to Save changes.





# 3.4 Add EXT4 rootfs Support

Let PetaLinux generate EXT4 rootfs. In terminal, execute:

```
$ petalinux-config
```

Go to Image Packaging Configuration. Enter into Root File System Type

Select Root File System Type EXT4

Change the Device node of SD device from the default value /dev/mmcblk0p2

to new value required for the TE0821 module: /dev/mmcblk1p2

Step up to

Image Packaging Configuration -->

modify Root filesystem formats from

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

to

ext4

Exit and Yes to save changes.

# 3.5 Let Linux Use EXT4 rootfs During Boot

The setting of which rootfs to use during boot is controlled by bootargs. We would change bootargs settings to allow Linux to boot from EXT4 partition.

In terminal, execute:

```
$ petalinux-config
```

### Change DTG settings->Kernel Bootargs->generate boot args automatically to NO.

Update User Set Kernel Bootargs to:

```
earlycon console=ttyPS0,115200 clk_ignore_unused root=/dev/mmcblk1p2 rw
rootwait cma=512M
```

Click OK, Exit three times and Save.

# 3.6 Build PetaLinux Image

In terminal, build the PetaLinux project by executing:

\$ petalinux-build

The PetaLinux image files will be generated in the directory:

	department of	
signau	processing	



~/work/TE0821 3 240/test board/os/petalinux/images/linux

Generation of PetaLinux takes some time and requires Ethernet connection and sufficient free disk space.

# 3.7 Create Petalinux SDK

The SDK is used by 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/TE0821 3 240/test board/os/petalinux/images/linux

Generation of SDK package takes some time and requires sufficient free disk space. Time needed for these two steps depends also on number of allocated processor cores.

# 3.8 Copy Files for Extensible Platform

Copy these four files:

Files	From	То
bl31.elf pmufw.elf system.dtb u-boot- dtb.elf	~/work/TE0821_3_240/ test_board/os/petalinux/ images/linux	~/work/TE0821_3_240/ test_board_pfm/pfm/boot

Rename the copied file u-boot-dtb.elf to u-boot.elf

The directory

~/work/TE0821 3 240/test board pfm/pfm/boot

contains these five files:

bl31.elf fsbl.elf pmufw.elf system.dtb u-boot.elf

Copy files:

Files	From	То		
boot.scr system.dtb	~/work/TE0821_3_240/ test_board/os/petalinux / images/linux	~/work/TE0821_3_240/ test_board_pfm/ pfm/sd_dir		



Copy file:

File	From	То
init.sh	~/work/TE0821_3_240/ test_board/misc/sd	~/work/TE0821_3_240/ test_board_pfm/pfm/sd_dir

init.sh is an place-holder for user defined bash code to be executed after the boot:

```
#!/bin/sh
normal="\e[39m"
lightred="\e[91m"
lightgreen="\e[92m"
green="\e[32m"
yellow="\e[33m"
cyan="\e[36m"
red="\e[31m"
magenta="\e[95m"
echo -ne $lightred
echo Load SD Init Script
echo -ne $cyan
echo User bash Code can be inserted here and put init.sh on SD
echo -ne $normal
```

# 3.9 Create Extensible Platform zip File

Create new directory tree:

```
~/work/TE0821_3_240_move/test_board/os/petalinux/images
~/work/TE0821_3_240_move/test_board/Vivado
~/work/TE0821_3_240_move/test_board_pfm/pfm/boot
~/work/TE0821_3_240_move/test_board_pfm/pfm/sd_dir
```

Copy all files from the directory:

Files	Source	Destination
all	<pre>~/work/TE0821_3_240/test_bo ard/os/petalinux/images</pre>	~/work/TE0821_3_240_move/test _board/os/petalinux/images
all	~/work/TE0821_3_240/test_bo ard_pfm/pfm/boot	~/work/TE0821_3_240_move/test _board_pfm/pfm/boot
all	~/work/TE0821_3_240/test_bo ard_pfm/pfm/sd_dir	~/work/TE0821_3_240_move/test _board_pfm/pfm/sd_dir
test_board _3cg_1e_4g b.xsa	<pre>~/work/TE0821_3_240/test_bo ard/Vivado/test_board_3cg_1 e_4gb.xsa</pre>	<pre>~/work/TE0821_3_240_move/test _board/Vivado/test_board_3cg_ 1e_4gb.xsa</pre>



ŪTĬA

32/61

https://sp.utia.cas..cz

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

Zip the directory ~/work/TE0821\_3\_240\_move

into ZIP archive: ~/work/TE0821\_3\_240\_move.zip

The archive TE0821\_3\_240\_move.zip can be used to create extensible platform on the same or on an another PC with installed Ubuntu 20.04 and Vitis tools, with or without installed Petalinux. The archive includes all needed components, including the Xilinx xrt library and the script sdk.sh serving for generation of the sysroot.

The archive has size approximately 3.6 GB and it is valid for the initially selected module number (84). This is the TE0821 HW module with xczu3cg-sfvc784-1-e device with 4 GB memory. The extensible Vitis platform will have the default clock 240 MHz.

Move the TE0821\_3\_240\_move.zip file to an PC disk drive.

Delete: ~/work/TE0821\_3\_240\_move ~/work/TE0821\_3\_240\_move.zip

Clean the Ubuntu Trash.

# 3.10 Generation of SYSROOT

This part of development can be direct continuation of the previous Petalinux configuration and compilation steps.

Alternatively, it is also possible to implement all next steps on an Ubuntu 20.04 without installed PetaLinux Only the Ubuntu 20.04 and Vitis/Vivado installation is needed. All required files created in the PetaLinux for the specific module (24) are present in the archive: TE0821\_3\_240\_move.zip In this case, unzip the archive to the directory: ~/work/TE0821\_3\_240 move

and copy all content of directories to ~/work/TE0821 3 240

Delete the **TE0821\_3\_240\_move.zip** file and the **~/work/TE0821\_3\_240\_move** directory to save filesystem space.

In Ubuntu terminal, change the working directory to: ~/work/TE0821 3 240/test board/os/petalinux/images/linux

In Ubuntu terminal, execute script enabling access to Vitis 2022.2 tools. Execution of script serving for setting up PetaLinux environment is not necessary:

In Ubuntu terminal, execute script

\$ source /tools/Xilinx/Vitis/2022.2/settings64.sh

Execute script:

./sdk.sh

In Ubuntu terminal, execute script

\$ ~/work/TE0821\_3\_240/test\_board\_pfm

signal processing

33/61

https://sp.utia.cas..cz



PetaLinux SDK installer version 2022.2

\_\_\_\_\_

Enter target directory for SDK (default: /opt/petalinux/2022.2):

Enter:

/home/devel/work/te0821\_3\_240/test\_board\_pfm

Reply: Y

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

```
~/work/TE0821_3_240/test_board_pfm/sysroots/x86_64-petalinux-linux
~/work/TE0821_3_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-
linux
```

Once created, do not move these sysroot directories (due to some internally created paths).

### 3.11 Generation of Extensible Platform for Vitis

In Ubuntu terminal, change the working directory to:

~/work/TE0821\_3\_240/test\_board\_pfm

Start the Vitis tool by executing

\$ vitis &

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

~/work/TE0821\_3\_240/test\_board\_pfm

Click on "Launch" to launch Vitis

Close Welcome page.

In Vitis, select in the main menu: File -> New -> Platform Project

Type name of the extensible platform: TE0821\_3\_240\_pfm. Click Next.





🔁 20_04_Vitis	_2022_2_Petalinux_	te0821_03cg_4gb - VMware Workstatio	n 16 Player			- 🗆 ×	
<u>P</u> layer ▼	- 4 []	2				≪ ₽	
Activities	🗸 Vitis ID	e ▼		Feb 1	2 13:37	♣ ● Ů ▼	
				New Platf	orm Project 🛛	× - • ×	
	File Edit S	Create new platform proje	ct		l l l l l l l l l l l l l l l l l l l		
	📬 🖛 🖪 🤅	Enter a name for your platfor	m project			参 Debug	
Â	<b>Explorer S</b> This wizard will guide you through creation of a platform project from the output of Vivado [Xilinx Shell Archive (XSA)] or from an existing platform. A platform will enable you to specify options for the kernels, BSPs, as well as settings required for creating new applications. Platforms are currently supported for embedded software developers.						
?		Platform project name:	TE0821_3_240_pl	m			
? •			Platform Project	System Project	<ul> <li>A platform provides hardware information and software environment settings.</li> <li>A system project contains one or more applications that run at the same time.</li> </ul>		
≥- -		Processor	Domain	Арр	<ul> <li>A domain provides runtime for applications, such as operating system or BSP.</li> <li>A workspace can contain unlimited platforms and unlimited system projects.</li> </ul>		
	Assistant : E	A new platform project c From hardware specific Create a new platform can be customized lai From existing platform	ation (XSA) n project from a h	ardware specification f	: ile. You can specify the OS and processor to start with. The platform	1	
	-	Load the platform de your platform project		isting platform. You ca	n choose any platform from the platform repository as a base for       < Back     Next >     Cancel     Finish	•	
	L	J					
	0 items selecte	d					
:::							

Choose for hardware specification for the platform file:

~/work/TE0821\_3\_240/test\_board/vivado/test\_board\_3cg\_1e\_4gb.xsa

In "Software specification" select: linux

In "Boot Components" unselect: Generate boot components (these components have been already generated by Vivado and PetaLinux design flow)

35/61

New window TE0821\_3\_240\_pfm is opened.

Click on linux on psu\_cortex53 to open window Domain: linux\_domain

In "Description" write: xrt

In "Bif File" find and select the pre-defied option: Generate Bif

In "Boot Components Directory" select: ~/work/TE0821 3 240/test board pfm/pfm/boot

In "FAT32 Partition Directory" select: ~/work/TE0821 3 240/test board pfm/pfm/sd dir



https://sp.utia.cas..cz



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

5 20_04_Vit	is_2022_2_Petalinux_te0821_0	3cg_4gb - VMw	vare Workstation 16 Player					- 🗆 ×
<u>P</u> layer ▼	- 4 2 2							≪ 📑
Activities	s 🚽 Vitis IDE 🔻			Feb 12 13:45		0		+ • U -
<u> </u>			test_board	l_pfm - TE0821_03_240_pfm/platform.spr - Vitis IDE				_ ē 😣
	File Edit Search	Xilinx Pro	ject Window Help					
	📑 🕶 🖬 👘 👘	- 🔨 -	5 🖉 🔳 🐐 • O • 🔗	▼ \$\$\$ \$\$\$ \$\$ \$\$ \$\$	Q	🕝 C	esig	n 🎋 Debug
	📲 E 🕱 🗖 🗖	✓ TE0821_	_03_240_pfm ☎			-		x
Á	🖻 🔄 🎽 🔋	🖴	Processor:	psu_cortexa53				There is no
A	▼ TE0821_03_240 ▶ ⊕ export		Supported Runtimes:	OpenCL 🔻				active editor that provides an
?	▶ 👝 hw		Display Name:	linux on psu_cortexa53				outline.
	<ul> <li>implementary logs</li> <li>implementary logs&lt;</li></ul>		Description:	xrt				
2	✓ platform.sp						1	
	🗹 platform.tcl		Bif File:	/home/devel/work/te0821_3_240/test_board_pfm/TE0821_03_	<u>B</u> rowse	•		
>_			Boot Components Directory:	/home/devel/work/te0821_3_240/test_board_pfm/pfm/boot	Browse	Q		
			Linux Rootfs:			Q		
-					Browse	Q		
			Bootmode	SD 🔻				
			FAT32 Partition Directory:	/home/devel/work/te0821_3_240/test_board_pfm/pfm/sd_dir	Br <u>o</u> wse	] <b>Q</b>		
	🖌 A 🐹 🗖 🗖		Sysroot Directory:		Bro <u>w</u> se	Q		
	🖻 🕀 🌼 🐔		QEMU Data:	/home/devel/work/te0821_3_240/test_board_pfm/TE0821_03_	Brow <u>s</u> e	Q		
	8 TE0821_03_240		QEMU Arguments:	/home/devel/work/te0821_3_240/test_board_pfm/TE0821_03_	Brows <u>e</u>	Q		
			PMU QEMU Arguments:	/home/devel/work/te0821_3_240/test_board_pfm/TE0821_03_	Browse	Q		
		Main Hard	lware Specification					
		📮 Console	జ 🖹 Problems 📗 Vitis Log (	🛈 Guidance 🔑 🗘 😨 📓		-	• 0	9
			ole [TE0821_03_240_pfm]	1 02 240 of a ****				
		buildplat	<pre>**** Build of project TE082 form.sh 44469 TE0821_03_246 ar Channel, tefebar#3</pre>					
			er Channel: tcfchan#1					
		13:45:10	Build Finished (took 5s.316	ms)				
	TE0821_03_240_pf	m						

In Vitis IDE "Explorer" section, click on TE0821\_03\_240\_pfm to highlight it.

Right-click on the highlighted TE0821\_03\_240\_pfm and select build project in the open submenu. Platform is compiled in few seconds. Close the Vitis tool by selection: File -> Exit.

Vits extensible platform TE0821\_03\_240\_pfm has been created in the directory:

```
~/work/TE0821_3_240/test_board_pfm/TE0821_3_240_pfm/export/TE0821_03_240_pf
m
```

## 4 Platform Usage

### 4.1 Read Platform Info

With Vitis environment setup, platforminfo tool can report XPFM platform information.

```
Platforminfo
~/work/TE0821_3_240/test_board_pfm/TE0821_3_240_pfm/export/TE0821_03_24
0_pfm/TE0821_3_240_pfm.xpfm
```



## 4.2 Create and Compile Vector Addition Example

 $Create new directory \verb+test_board_test_vadd to test Vitis extendable flow example ``vector addition"$ 

~/work/TE0821\_3\_240/test\_board\_test\_vadd

Current directory structure:

~/work/TE0821\_3\_240/test\_board
~/work/TE0821\_3\_240/test\_board\_pfm
~/work/TE0821\_3\_240/test\_board\_test\_vadd

Change working directory:

\$cd ~/work/TE0821\_3\_240/test\_board\_test\_vadd

In Ubuntu terminal, start Vitis by:

\$ vitis &

In Vitis IDE Launcher, select your working directory

~/work/TE0821 3 240/test board test vadd

Click on Launch to launch Vitis.

Select File -> New -> Application project. Click Next.

Skip welcome page if shown.

Click on [+ Add] icon and select the custom extensible platform TE0821\_03\_240\_pfm[custom] in the directory:

~/work/TE0821\_3\_240/test\_board\_pfm/TE0821\_03\_240\_pfm/export/TE0821\_03\_2 40 pfm

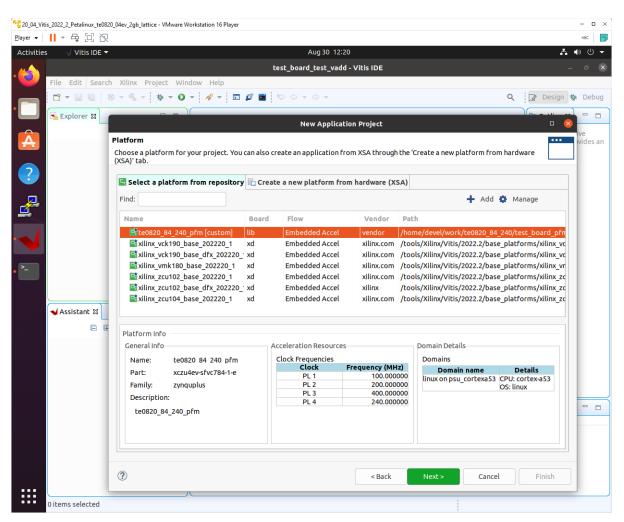
We can see available PL clocks and frequencies. PL4 with 240 MHz clock is has been set as default in the platform creation process.

37/61



https://sp.utia.cas..cz

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



#### Click Next.

In Application Project Details window type into Application project name: test\_vadd Click Next.

In Domain window type (or select by browse):

#### Sysroot path:

~/work/TE0821\_3\_240/test\_board\_pfm/sysroots/cortexa72-cortexa53-xilinxlinux

#### Root FS:

~/work/TE0821\_3\_240/test\_board/os/petalinux/images/linux/rootfs.ext4

#### Kernel Image:

~/work/TE0821\_3\_240/test\_board/os/petalinux/images/linux/Image

Click Next.

In Templates window, if not done before, update Vitis IDE Examples and Vitis IDE Libraries.

#### Select Host Examples:

In Find, type: vector add to search for the Vector Addition example.





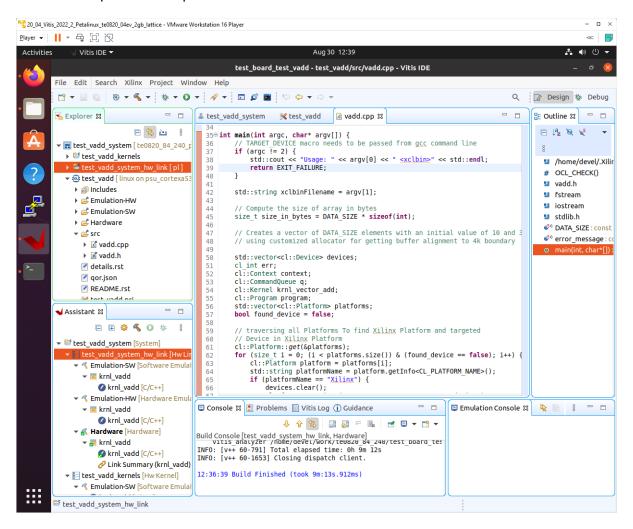
Select: Vector Addition Click Finish New project template is created.

In test\_vadd window menu "Active build configuration" switch from SW Emulation to Hardware.

In "Explorer" section of Vitis IDE, click on: test\_vadd\_system[TE0821\_03\_240\_pfm] to select it.

Right Click on: test\_vadd\_system[TE0821\_03\_240\_pfm] and select in the opened submenu: Build project

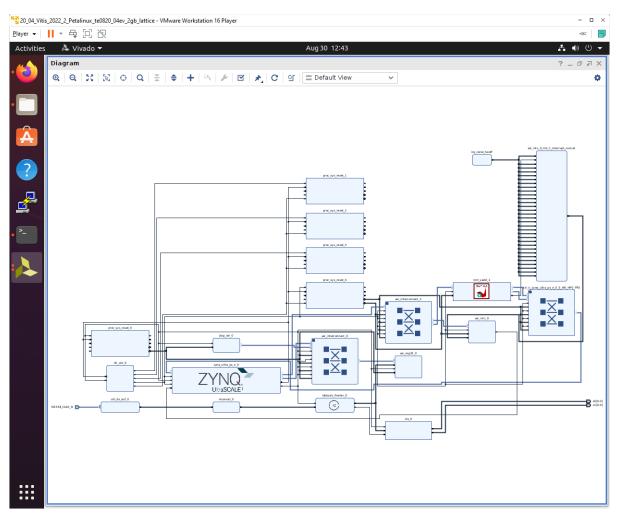
Vitis will compile. This step can take some time.





https://sp.utia.cas..cz

39/61



Created extended HW with integrated vadd IP block can be open and analysed in Vivado 2022.2.

### 4.3 Run Compiled test\_vadd Example Application

The sd\_card.img file is output of the compilation and packing by Vitis. It is located in directory:

```
~/work/TE0821_3_240/test_board_test_vadd/test_vadd_system/Hardware/pack age/sd card.img
```

Write the sd card image sd\_card.img to SD card.

In Windows Pro 10 (or Windows 11 Pro) PC, inst all program Win32DiskImager for this task. Win32 Disk Imager can write raw disk image to removable devices. <u>https://win32diskimager.org/</u>

Insert the SD card to the TE0701-06 carrier board.

Connect PC USB terminal (115200 bps) card to the TE0701-06 carrier board.

Connect Ethernet cable to the TE0701-06 carrier board.

Power on the TE0701-06 carrier board.

signal processing

40/61



In PC, find the assigned serial line COM port number for the USB terminal. In case of Win 10 use device manager.

In PC, open serial line terminal with the assigned COM port number. Speed 115200 bps.

On TE0701-06, reset button to start the system. USB terminal starts to display booting information.

In PC terminal, type:

```
sh-5.0# cd /media/sd-mmcblk1p1/
sh-5.0# ./test vadd krnl vadd.xclbin
```

The application test\_vadd should run with this output:

```
INFO: Reading krnl_vadd.xclbin
Loading: 'krnl_vadd.xclbin'
Trying to program device[0]: edge
Device[0]: program successful!
TEST PASSED
sh-5.0#
```

The Vitis application has been compiled to HW and evaluated on custom system with extensible custom TE0821\_03\_240\_pfm platform.

In PC terminal type:

# halt

System is halted. Messages relate to halt of the system can be seen on the USB terminal.

The SD card can be safely removed from the TE0701-06 carrier board, now.

The TE0701-06 carrier board can be disconnected from power.

System can be connected to the X11 terminal running on your PC Ubuntu with PuTTY application via Ethernet.

Find Ethernet IP address of your board by **ifconfig** command in PetaLinux terminal. In PC Ubuntu OS, open PuTTY application. In PuTTY, set Ethernet IP of your board. In PuTTY, select checkbox SSH->X11->Enable X11 forwarding.

Use PC Ubuntu mouse and keyboard. In PuTTY, open PetaLinux terminal and login as: user: root pswd: root.

In opened PetaLinux terminal, start X11 desktop x-session-manager by typing:

root@Trenz:~# x-session-manager &

Click on X11 icon (A Unicode capable rxvt)

signal processing

41/61



Terminal opens as an X11 graphic window. In X11 terminal rxvt, use Ubuntu PC keyboard and type:

```
sh-5.0# cd /media/sd-mmcblk1p1/
sh-5.0# ./test vadd krnl vadd.xclbin
```

The application test\_vadd should run with this output:

```
INFO: Reading krnl_vadd.xclbin
Loading: 'krnl_vadd.xclbin'
Trying to program device[0]: edge
Device[0]: program successful!
TEST PASSED
sh-5.0#
```

The test\_board has been running the PetaLinux OS and drives simple version of an X11 GUI on Ubuntu desktop.Application test\_vadd has been started from X11 xrvt terminal emulator.

Close the rxvt terminal emulator by click "x" icon (in the upper right corner) or by typing:

sh-5.0# exit

In X11, click Shutdown icon to safely close PetaLinux running on the test board.

System on the test board is halted. Messages related to halt of the system can be seen on the PC USB terminal.

The SD card can be safely removed from the test\_board, now. Close the PC USB terminal application. The TE0701-06 carrier board can be disconnected from power, now.

## 5 Vitis AI 3.0 DPUCZDX8V\_VAI\_v3.0 Installation

This test implements simple AI 3.0 demo to verify DPU integration to our custom extensible platform. This tutorial follows Xilix Vitis Tutorial for zcu104 with necessary fixes and customizations required for our case.

We have to install correct Vitis project with the DPU instance from this repository:

https://github.com/Xilinx/Vitis-Al/tree/3.0/dpu

Page description contains table with supported targets. Use the line if theis table dedicated to DPUCZDX8G DPU for MPSoC and Kria K26 devices.

It is link for download of the programmable logic based DPU, targeting general purpose CNN inference with full support for the Vitis AI ModelZoo. Supports either the Vitis or Vivado flows on 16nm Zyng® UltraScale+™ platforms.

Click on the Download link in the column: Reference Design

This will result in download of file:





~/Downloads/DPUCZDX8V\_VAI\_v3.0.tar.gz

It contains directory

~/Downloads/DPUCZDX8V VAI v3.0

Copy this directory to the directory:

~/work/DPUCZDX8V VAI v3.0

It contains HDL code for the DPU and also source files and project files to test the DPU with AI resnet50 inference example.

### 5.1 Create and Build Vitis Design

Create new directory test\_board\_dpu\_trd to test Vitis extendable flow example dpu trd

#### ~/work/TE0821\_3\_240/test\_board\_dpu\_trd

Current directory structure:

```
~/work/TE0821_3_240/test_board
~/work/TE0821_3_240/test_board_pfm
~/work/TE0821_3_240/test_board_test_vadd
~/work/TE0821_3_240/test_board_dpu_trd
```

Change working directory:

\$cd ~/work/TE0821\_3\_240/test\_board\_dpu\_trd

In Ubuntu terminal, start Vitis by:

\$vitis &

In Vitis IDE Launcher, select your working directory

~/work/TE0821\_3\_240/test\_board\_dpu\_trd

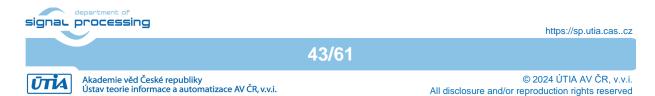
Click on Launch to start Vitis.

#### 5.2 Add DPU Project template to the Vitis Extensible Flow

Open menu Window → Preferences

Go to Library Repository tab

Add Vitis-AI by clicking Add button and fill the form as shown below, use absolute path to your home folder in field Location



20_04_Vit	tis_2022_2_Petalinux_te0808_15eg_4gb_lattice - VM	fware Workstation 16 Player					- 🗆 ×
<u>P</u> layer ▼	- 母 [2] 泡						≪ 📑
Activitie	s 🚽 Vitis IDE 🔫		Aug 18 11:10				♣ ● Ů ▼
			StarterKit_test_dpu_trd - Vitis IDE				- 🗆 💌
	File Edit Search Xilinx Project	Window Help					
		• 0 • 🛷 • 🖻 🕫				Q De	esign 🎋 Debug
•	Explorer 🕱					E Out	line 🛛 🗖 🗖
	E 😒 🔤		Preferer	nces			• 😣
		type filter text	Library Repositories			1	⇔ - ⇔ - 8
?		▼ Xilinx Example Repositor	Repository	Settings			
		Guidance	Vitis Accelerated Libraries Repository Vitis AI	ID	vitis-ai		
		Library Repositorie Project Preference	VIUS AI	Name	Vitis Al		
		Software Reposito		N	vitis AI 3.0		
• ~-		Toolchain Preferen • Additional		Description			
	Assistant 🛱 📟	General					
-		▶ C/C++		Location	/home/devel	/work/DPUCZDX8G_	VAI_v3.0
		<ul> <li>Run/Debug</li> <li>Team</li> </ul>		Git URL			
				Branch			
			Add Remove				-
						Restore <u>D</u> efaults	Apply
		? >				Cancel	pply and Close
	0 items selected						
	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>		absolute path to your home directory. It dep	ands on the use	The u	cor name in the figure	is "doual". Doplace il
			by your user name.		er name. me u	ser hame in the righte	is dever, replace in
		Corr	ectly added library appears in Libraries:				
			n menu Xilinx → Libraries				
		O You	can find there just added Vitis-AI library mark	ed as "Installed"	' as shown in in	nage:	
		Sectors Address	마지 (그가 마마지, Sun (그, 다, 고), 오가 (Mare Boldelin T Page ] +  위 () 등 등 ( VEN DC +	99-11 • Ind - VRI - DE	_		
		<u></u>	na da lant ten tent tente ten D - 1 S = S - S - E € € 1			A Display and	

### Click Apply and Close.

Field Location says that the Vitis-AI repository from github has been allready cloned into

~/work/DPUCZDX8V\_VAI\_v3.0

folder, in the stage of Petalinux configuration. 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.

44/61

Correctly added library appears in Libraries:

Open menu Xilinx → Libraries...

You can find there just added Vitis-AI library marked as Installed





ties Vitis II	DE 🕶	Au- 40 4440	
File Edit S		Aug 18 11:12	A 📣 (
File Edit S		StarterKit_test_dpu_trd - Vitis IDE	
	earch Xilinx Project Window Help		
		* 4 - 4 -	🔍 🛛 🔐 Design 🗱 De
🛃 Explorer 🖇			E Outline 🛛 🖓
		Vitis IDE Libraries	
	Vitis IDE Libraries		
		ss "Download" to download a library repository, or or "Refresi	" to check for the latest undates
	Examples from libraries will be available as templates		r to check for the latest updates.
		Detaile	
	Find:	E Details:	
	👻 🗁 Vitis Accelerated Libraries Repository	Name: Vitis AI Installed Directory: /home/devel/work/I	DPUCZDX8G VAI v3.0
	✓ ultrasound lib	Installed Description: Vitis AI 3.0	51 0C2570G_VAI_V5.0
	✓ Vitis BLAS Library	Installed	
	✓ Vitis Codec Library	Installed	
	✓ Vitis Data Analytics Library	Installed	
	✓ Vitis Data Compression Library	Installed	
	✓ Vitis Database Library	Installed	
	✓ Vitis DSP Library	Installed	
Assistant a	✓ Vitis Graph Library	Installed	
	✓ Vitis HPC Applications	Installed	
E	✓ 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 Add to project 💌	

## 5.3 Configure Project for the Vitis Extensible Flow with DPU

Select File -> New -> Application project. Click Next.

Skip welcome page, if it is shown.

Click on [+ Add] icon and select the custom extensible platform TE0821\_03\_240\_pfm[custom] in the directory:

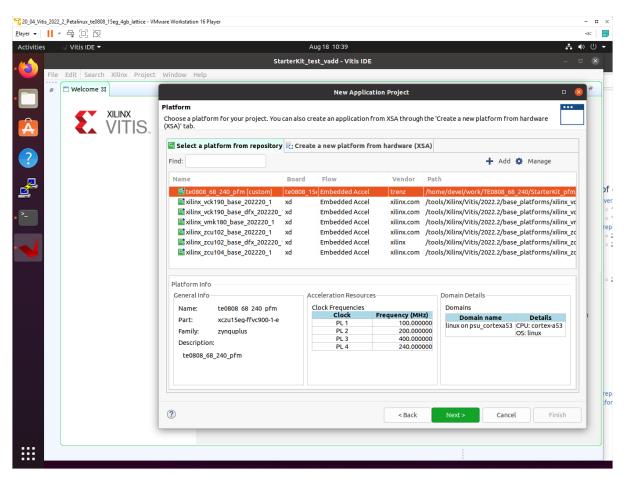
~/work/TE0821\_3\_240/test\_board\_pfm/TE0821\_03\_240\_pfm/export/TE0821\_03\_240\_p fm

We can see available PL clocks and frequencies. PL4 with 240 MHz clock was set as the default in the platform creation process.

45/61



https://sp.utia.cas..cz



#### Click Next.

In Application Project Details window type into Application project name: dpu\_trd Click Next.

In Domain window type (or select by browse):

#### "Sysroot path":

```
~/work/TE0821_3_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-linux
```

#### "Root FS":

```
~/work/TE0821 3 240/test board/os/petalinux/images/linux/rootfs.ext4
```

#### "Kernel Image":

~/work/TE0821 3 240/test board/os/petalinux/images/linux/Image

#### Click Next.

In Templates window, if not done before, update Vitis IDE Examples and Vitis IDE Libraries

46/61

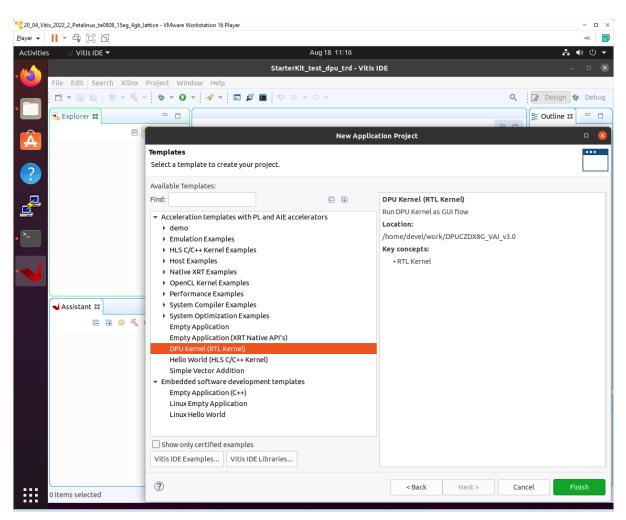
In "Find", type: dpu to search for the DPU Kernel (RTL Kernel) example.

Select: DPU Kernel (RTL Kernel)



https://sp.utia.cas..cz





### Click Finish

New project template is created.

In dpu\_trd window menu  $\mbox{Active build configuration switch from SW Emulation to Hardware}$ 

File dpu\_conf.vh located at dpu\_trd\_kernels/src/prj/Vitis directory contains DPU configuration.

**Open file dpu\_conf.vh and change in line 37**: `define B4096

to

`define B1600

and save modified file.





전 20_04_Vitis_2022_2_Petalinux_te0808_15eg_4gb_lattice - VMware \	Norkstation 16 Player	- 🗆 ×
<u>P</u> layer - 비 - 육 □ 전		≪ 📮
Activities Vitis IDE 🗸	Aug 18 11:18	
		_ 0 🕅
		•
File Edit Search Xilinx Project Wir		
	$\bullet  \not \land  \bullet  \bullet  \bullet  \bullet  \bullet  \bullet  \bullet  \bullet  \bullet $	📝 Design 🎋 Debug
Explorer 🛛 🗖 🗖	👗 test_dpu_trd_system 🛛 🛠 test_dpu_trd 📄 *dpu_conf.vh 🛪 📃 🗖	🗄 Outline 🛛 🗖 🗖
<ul> <li>Solution</li> <li>Sol</li></ul>	36//       +         37//       ' define B3136         38//       +         38//       ' define B4096         40//       -         41       -         42       ' define B4096         43       -         44//       -         45//       I f the FPGA has Uram. You can define URAM_EN parameter         46//       i f change, Don't need update model         47//       +         48//         f or zcu104 : 'define URAM_ENABLE         49//       +         51//         for zcu102 : 'define URAM_DISABLE         51//       -	There is no active editor that provides an outline.
	55 //config URAM         56 'ifdef URAM_ENABLE         57 'define def UBANK_IMGT N         58 'define def UBANK_GIAS         1         60 'elsif urAM_DISABLE         61 'define def UBANK_IMGT N         62 'define def UBANK_BIAS         63 'define def UBANK_BIAS         64'endif         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •          •         •         •         •         •	3 14 10 8 - 0
DPUCZDX8G [User Manage     Sfm_xrt_top [User Manage		

Go to dpu\_trd\_system\_hw\_link and double click on dpu\_trd\_system\_hw\_link.prj

Remove sfm\_xrt\_top kernel from binary container by right clicking on it and choosing remove.

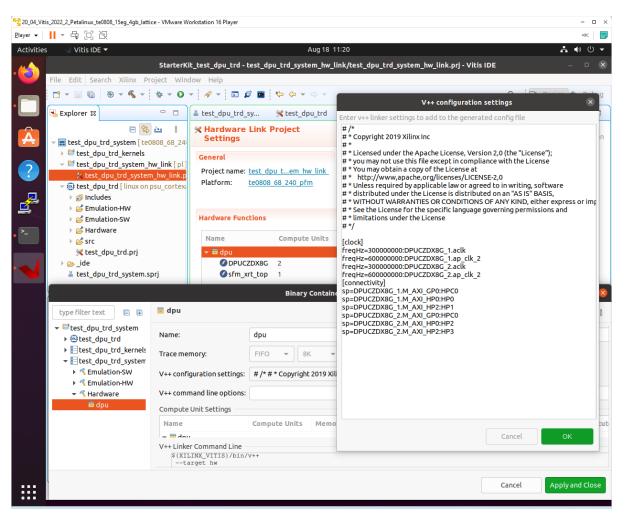
48/61

Reduce number of DPU kernels to one.

## 5.4 Configure Connection of DPU kernel

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:

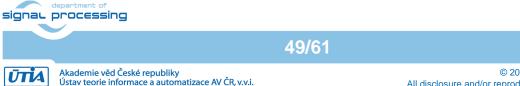
```
[clock]
freqHz=20000000:DPUCZDX8G_1.aclk
freqHz=40000000:DPUCZDX8G_1.ap_clk_2
[connectivity]
sp=DPUCZDX8G_1.M_AXI_GP0:HPC0
sp=DPUCZDX8G_1.M_AXI_HP0:HP0
sp=DPUCZDX8G_1.M_AXI_HP2:HP1
```

### 5.5 Build the test\_dpu\_trd Project

In "Explorer" section of Vitis IDE, click on:

dpu trd system[te0802 04 240 vga pfm]

to select it.



```
© 2024 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved
```

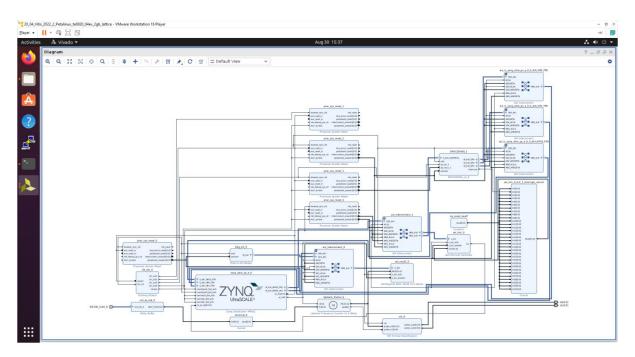
Right Click on:

```
dpu_trd_system[te0802_04_240_vga_pfm]
```

and select in the opened sub-menu: Build project

Compilation takes some time (approximately 30 minutes).

Created extended HW with integrated DPU with configuration B4096 can be open and analysed in Vivado 2022.2



# 6 Prepare SD card with test\_dpu\_trd DPU

Write sd\_card.img to SD card using SD card reader.

The **sd\_card.img** file is output of the compilation and packing by Vitis. It is located in directory:

~/work/TE0821\_3\_240/test\_board\_dpu\_trd/dpu\_trd\_system/Hardware/package

In Windows 10 (or Windows 11) PC, inst all program **Win32DiskImager** for this task. Win32 Disk Imager can write raw disk image to removable devices. <u>https://win32diskimager.org/</u>

Boot the board and open terminal on the board either by connecting serial console connection, or by opening ethernet connection to ssh server on the board, or by opening terminal directly using window manager on board. Continue using the embedded board terminal.

Detailed guide how to run embedded board and connect to it can be found in <u>Run Compiled</u> <u>Example Application for Vector Addition</u>.

50/61



https://sp.utia.cas..cz

## 6.1 Resize EXT4 Partition

#### Check ext4 partition size by:

root@Trenz:~# cd /						
<pre>root@Trenz:~# df .</pre>						
Filesystem	1K-blocks	Used	Available	Use%	Mounted	on
/dev/root	564048	398340	122364	77%	/	

#### **Resize** partition

```
root@Trenz:~# resize-part /dev/mmcblk0p2
```

Check ext4 partition size again, you should see:

root@Trenz:~# dfh						
Filesystem	Size	Used A	Available	Use%	Mounted	on
/dev/root	6.1G	390.8M	5.4G	7%	/	

The available size would be different according to your SD card size.

Set DISPLAY variable:

```
root@petalinux:~# export DISPLAY=:0.0
```

Set path to Xilinx Firmware:

```
root@petalinux:~# export
XLNX VART FIRMWARE=/run/media/mmcblk1p1/dpu.xclbin
```

### 6.2 Test the Integrated DPUCZDX8G

For both tested modules, the integrated DPU can be tested by command:

xdputil query

Command and reply in case of module with ID=3 (DPU configuration B1600):

```
sh-5.1# cd ~
sh-5.1# export XLNX_VART_FIRMWARE=/run/media/mmcblk1p1/dpu.xclbin
sh-5.1#
sh-5.1# xdputil query
{
    "DPU IP Spec":{
        "DPU Core Count":1,
        "IP version":"v4.1.0",
        "generation timestamp":"2023-02-21 21-30-00",
        bttps://sp.utia.cas..cz
51/61
```



```
"git commit id":"7d32c41",
        "git commit time":2023022121,
        "regmap":"1to1 version"
    },
    "VAI Version":{
        "libvart-runner.so":"Xilinx vart-runner Version: 3.0.0-
c5d2bd43d951c174185d728b8e5bcda3869e0b39 2024-02-02-17:09:44 ",
        "libvitis ai library-dpu task.so":"Xilinx vitis ai library
dpu_task Version: 3.0.0-c5d2bd43d951c174185d728b8e5bcda3869e0b39 2023-
01-13 06:58:30 [UTC] ",
        "libxir.so":"Xilinx xir Version: xir-
c5d2bd43d951c174185d728b8e5bcda3869e0b39 2024-02-02-17:08:29",
        "target factory":"target-factory.3.0.0
c5d2bd43d951c174185d728b8e5bcda3869e0b39"
    },
    "kernels":[
        {
            "DPU Arch": "DPUCZDX8G ISA1 B1600",
            "DPU Frequency (MHz)":300,
            "IP Type":"DPU",
            "Load Parallel":2,
            "Load augmentation": "enable",
            "Load minus mean":"disable",
            "Save Parallel":2,
            "XRT Frequency (MHz)":300,
            "cu addr":"0xa0010000",
            "cu handle":"0xaaaaea37e0d0",
            "cu idx":0,
            "cu mask":1,
            "cu name":"DPUCZDX8G:DPUCZDX8G 1",
            "device id":0,
            "fingerprint":"0x101000056010404",
            "name":"DPU Core 0"
```



52/61

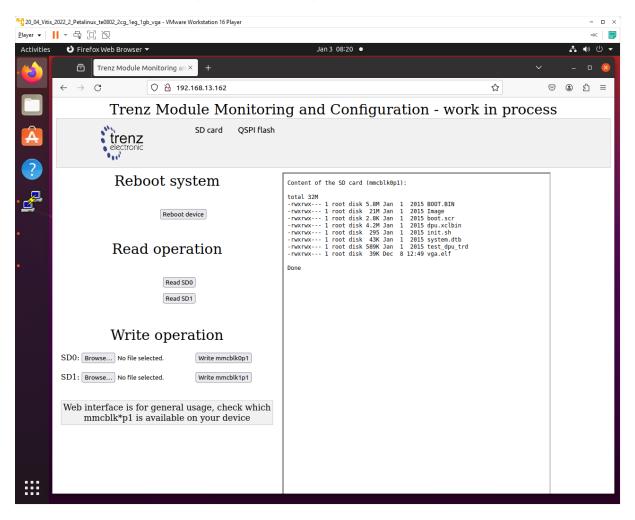


] } sh-5.1#

## 6.3 Remote Monitoring and Configuration Support

The configured OS includes work in progress version of a remote monitoring and configuration support server. It can be used for remote reading of content of the SD card partition mmcblk0p1.

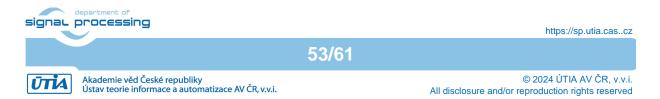
Button Reboot device can be used for system reboot. Ethernet connection is lost, but remote PC www browser remains open and waits for possible reconnection.



After reboot of the evaluation board, the network DHCP server assigns Ethernet address to the evaluation board.

If the network DHCP address assignment algorithm assigns the identical Ethernet address, the page can be refreshed and the connection is re-established again.

If the network DHCP address assignment algorithm assigns different Ethernet address, the connection has to be established on the new Ethernet address.



## 6.4 Remote Control from Ubuntu X11 Desktop.

The configured OS also supports X11 desktop on remote PC via Ethernet. In remote PC in Ubuntu OS, in PuTTY terminal utility with ssh Ethernet connection to the board with enabled X11 forwarding.

### Openning.

Log in to the evaluation board as user root with pswd root

Start two rxvt terminal emulators by typing in PuTTY terminal:

rxvt &

rxvt &

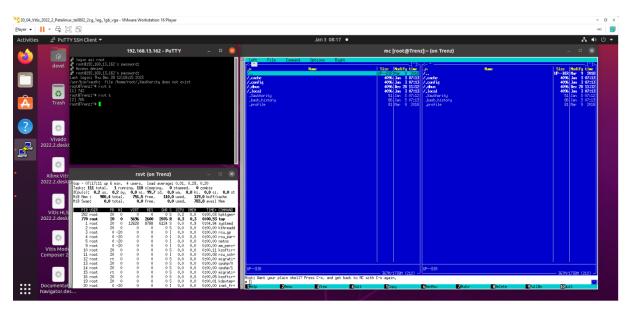
In first rxvt terminal emulator window start utility

top

In second rxvt terminal emulator start

mc

You can see two applications running on the evaluation board with output on the remote desktop. Remote PC kbd and mouse are used for control of these applications.



### Closing.

On remote PC, close top utility by Ctrl-C. Stop mc utility by F10.

Close open terminal emulators by typing exit or by mouse click on x icon in the right top corner of terminal emulator window. Close PuTTY connection by typing exit or by mouse click on x icon in the right top corner of PuTTY window.

### 6.5 Remote Control in x-session-manager on Ubuntu X11 Desktop.

The configured OS also supports x-session-manager on X11 desktop on remote PC connected via Ethernet to the evaluation board.

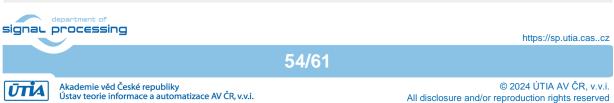
### Opening.

In remote PC in Ubuntu OS, start PuTTY terminal utility with ssh Ethernet connection to the board with enabled X11 forwarding.

Log in to the evaluation board as user root with pswd root

In PuTTY terminal, start x-session-manager by typing:

x-session-manager &



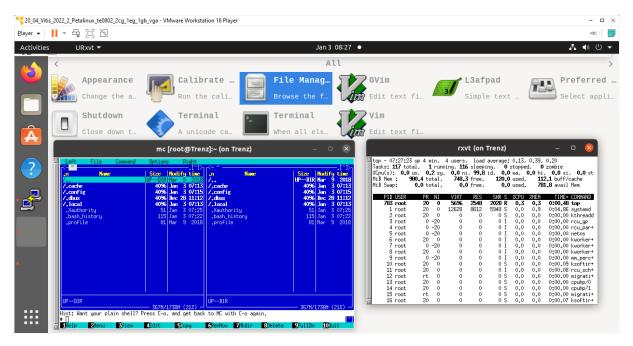
The desktop (displayed on the VGA display of the evaluation board) is also displayed in the remote PC X11 desktop. Start two rxvt terminal emulators by typing in PuTTY terminal:

rxvt &

rxvt &

In first rxvt terminal emulator window start utility top In second rxvt terminal emulator start mc

You can see two applications running on the evaluation board with output on the remote desktop. Remote PC kbd and mouse are used for control of these applications.



#### Closing.

On remote PC, close top utility by Ctrl-C. Stop mc utility by key F10.

Close open terminal emulators by typing exit or by mouse click on x icon in the right top corner of terminal emulator window. Close PuTTY connection by typing exit or by mouse click on x icon in the right top corner of PuTTY window.

### 6.6 Display Test Pattern and Test USB Camera

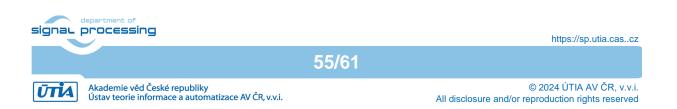
Complete video chain can be tested with output to the X11 desktop.

To display the test pattern, use this gstreamer command:

gst-launch-1.0 videotestsrc ! ximagesink

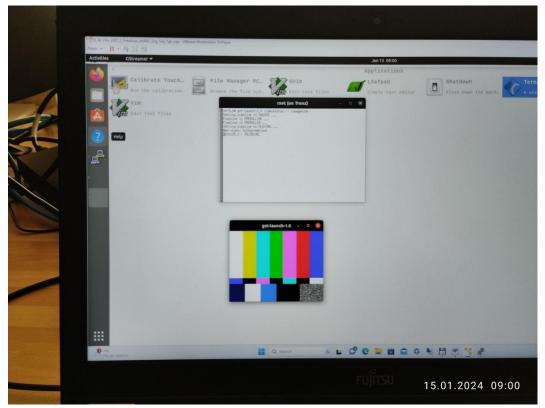
To display USB camera video, use this gstreamer command:

gst-launch-1.0 v4l2src device=/dev/video0 ! videoconvert ! ximagesink



Video output is directed to the local HD VGA display, if the command is started from local X11 console.

Video output is directed to the remote X11desktop, if the command is started from the remote X11 console.



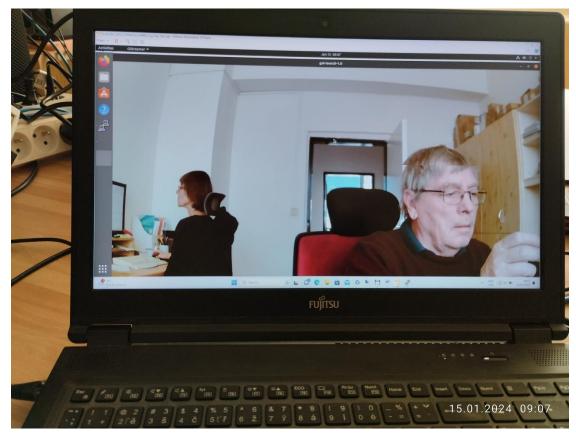
Test pattern is displayed on remote PC X11 desktop

56/61



https://sp.utia.cas..cz

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



Full HD video from USB camera is displayed as Full HD on remote PC X11 desktop.

57/61



https://sp.utia.cas..cz

Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.  $$\ensuremath{\mathbb{C}}$  2024 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

## 6.7 Vitis AI 3.0 TE0821-01-2AE31KA, TE0701-06, DPU (B1024)

Vitis AI 3.0 exampes	Perfor	Pow	Perfor	Power	GigaOp
	mance	er	mance	with	S
	input	with	input	input	input
	from	cam	from	from	from
	camera	era	file	file	file
	e2e	e2e	e2e	e2e	e2e
	(-t 1)	(-t 1)	(-t 3)	(-t 3)	(-t 3)
	[FPS]	[W]	[FPS]	[W]	[Gops]
Yolov4 face mask detection Model: pt_face-					
mask-detection_512_512_0.67G_3.0	20.0	7.7	67.0	7.5	44.9
Vehicleclassification vehicle make Model:	20.0	8.1	44.3	7.9	161.2
pt_vehicle-make-					
classification_VMMR_224_224_3.64G_3.0					
Vehicleclassification vehicle type Model:	20.0	8.1	44.3	7.9	161.2
pt_vehicle-type-		_	_	_	_
classification_CarBodyStyle_224_224_3.64G_3.0					
Classification vehicle color Model: pt_vehicle-	20.0	8.2	44.3	8.0	161.2
color-classification_VCoR_224_224_3.64G_3.0		_	_		_
Classification Model:	16.1	8.8	19.3	8.2	158.3
pt_resnet50_imagenet_224_224_8.2G_3.0					
Classification Model:	20.0	8.8	24.9	8.2	144.4
pt_resnet50_imagenet_224_224_0.3_5.8G_3.0					
Classification Model:	20.0	8.6	28.2	8.2	138.2
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0					
Classification Model:	20.0	8.5	32.1	8.2	131.6
pt_resnet50_imagenet_224_224_0.5_4.1G_3.0					
Classification Model:	20.0	8.4	37.3	8.2	123.1
pt_resnet50_imagenet_224_224_0.6_3.3G_3.0		_		_	
Classification Model:	20.0	8.3	44.1	8.1	110.2
pt_resnet50_imagenet_224_224_0.7_2.5G_3.0				_	

Measurement conditions:

• ID=1, TE0821-01-2AE31KA module 2cg-1e, 4GB DDR4, TE0701-06 carrier board

58/61

- DPU in B1024 configuration
- USB WWW camera ETERNICO ET201 Full HD, sensor JX\_F23, 1920x1080, 20 FPS
- Keyboard RPi
- Mouse RPi
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug





## 6.8 Vitis AI 3.0 TE0821-01-3AE31KA, TE0701-06, DPU (B1600)

Vitis AI 3.0 exampes	Perfor	Pow	Perfor	Power	GigaOp
	mance	er	mance	with	S
	input	with	input	input	input
	from	cam	from	from	from
	camera	era	file	file	file
	e2e	e2e	e2e	e2e	e2e
	(-t 1)	(-t 1)	(-t 3)	(-t 3)	(-t 3)
	[FPS]	[W]	[FPS]	[W]	[Gops]
Yolov4 face mask detection Model: pt_face-					
mask-detection_512_512_0.67G_3.0	20.0	7.7	76.1	7.6	50.9
Vehicleclassification vehicle make Model:	20.0	7.9	61.2	8.4	222.7
pt vehicle-make-	20.0	1.9	01.2	0.4	222.1
classification_VMMR_224_224_3.64G_3.0					
Vehicleclassification vehicle type Model:	20.0	7.9	61.2	8.4	222.7
pt_vehicle-type-	20.0	1.5	01.2	0.7	
classification_CarBodyStyle_224_224_3.64G_3.0					
Classification vehicle color Model: pt_vehicle-	20.0	7.9	61.2	8.4	222.7
color-classification_VCoR_224_224_3.64G_3.0	_0.0		•	0	
Classification Model:	20.0	8.8	27.5	8.6	225.5
pt_resnet50_imagenet_224_224_8.2G_3.0	_0.0	0.0		0.0	
Classification Model:	20.0	8.5	36.0	8.6	208.8
pt_resnet50_imagenet_224_224_0.3_5.8G_3.0		••••			
Classification Model:	20.0	8.4	39.7	8.5	194.5
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0					
Classification Model:	20.0	8.3	44.1	8.5	180.8
pt_resnet50_imagenet_224_224_0.5_4.1G_3.0					
Classification Model:	20.0	8.2	45.4	8.4	149.8
pt_resnet50_imagenet_224_224_0.6_3.3G_3.0				-	
Classification Model:	20.0	8.0	58.3	8.4	145.7
pt_resnet50_imagenet_224_224_0.7_2.5G_3.0					

Measurement conditions:

• ID=3, TE0821-01-3AE31KA module 3cg-1e, 4GB DDR4, TE0701-06 carrier board

59/61

- DPU in B1600 configuration
- USB WWW camera ETERNICO ET201 Full HD, sensor JX\_F23, 1920x1080, 20 FPS
- Keyboard RPi
- Mouse RPi
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug



## 6.9 Vitis AI 3.0 TE0821-01-3BE21FA, TE0701-06, DPU (B1600)

		Davis		D	
Vitis AI 3.0 exampes	Perfor	Pow	Perfor	Power	GigaOp
	mance	er	mance	with	S
	input	with	input	input	input
	from	cam	from	from	from
	camera	era	file	file	file
	e2e	e2e	e2e	e2e	e2e
	(-t 1)	(-t 1)	(-t 3)	(-t 3)	(-t 3)
	[FPS]	[W]	[FPS]	[W]	[Gops]
Yolov4 face mask detection Model: pt_face- mask-detection_512_512_0.67G_3.0	20.0	8.0	77.3	8.0	51.8
Vehicleclassification vehicle make Model:	20.0	8.6	61.7	9.2	224.6
pt_vehicle-make-	_0.0	0.0	• …	0	
classification_VMMR_224_224_3.64G_3.0					
Vehicleclassification vehicle type Model:	20.0	8.6	61.7	9.2	224.6
pt_vehicle-type-	_0.0	0.0	• …	0	
classification_CarBodyStyle_224_224_3.64G_3.0					
Classification vehicle color Model: pt_vehicle-	20.0	8.6	61.7	9.2	224.6
color-classification_VCoR_224_224_3.64G_3.0			• • • •	•.=	
Classification Model:	20.0	9.7	27.5	9.5	225.5
pt_resnet50_imagenet_224_224_8.2G_3.0	_0.0	0		0.0	
Classification Model:	20.0	9.3	35.9	9.5	208.2
pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	2010	0.0	0010	0.0	20012
Classification Model:	20.0	9.2	39.9	9.4	195.5
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	2010	0.2	0010	0	10010
Classification Model:	20.0	9.0	44.2	9.4	181.2
pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	_0.0	0.0		••••	
Classification Model:	20.0	8.8	45.3	9.2	145.5
pt_resnet50_imagenet_224_224_0.6_3.3G_3.0				•.=	
Classification Model:	20.0	8.7	58.6	9.2	146.5
pt_resnet50_imagenet_224_224_0.7_2.5G_3.0				•	

Measurement conditions:

• ID=5, TE0821-01-3BE21FA module 3eg-1e, 2GB DDR4, TE0701-06 carrier board

60/61

- DPU in B1600 configuration
- USB WWW camera ETERNICO ET201 Full HD, sensor JX\_F23, 1920x1080, 20 FPS
- Keyboard RPi
- Mouse RPi
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug



https://sp.utia.cas..cz



#### References 7

[1]

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for STM32H573I-DK web server. (Application note, with evaluation package, UTIA). Published for public access from: https://zs.utia.cas.cz/index.php?ids=results&id=1\_STM32H573\_DK

This application and evaluation package will be based on the STM32CubeH5 Firmware Examples for STM32H5xx Series Application based on NetXDuo: Nx\_WebServer. This STM application provides an example of Azure RTOS NetX Duo stack usage on STM32H573G-DK board, it shows how to develop Web HTTP server based application. https://htmlpreview.github.io/?https://raw.githubusercontent.com/STMicroelectronics/STM 32CubeH5/master/Projects/STM32CubeProjectsList.html

[2]

Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-1BEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note with evaluation package, UTIA). Published for public free access from: https://zs.utia.cas.cz/index.php?ids=results&id=2\_TE0802-02-1BEV2-A\_AI\_3\_0\_VGA

[3]

Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-2AEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note, with evaluation package, UTIA). Published for public access from: https://zs.utia.cas.cz/index.php?ids=results&id=3\_TE0802-02-2AEV2-A\_AI\_3\_0\_VGA

[4]

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0821 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from: https://zs.utia.cas.cz/index.php?ids=results&id=4 TE0821 AI 3 0

[5]

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0820 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from: https://zs.utia.cas.cz/index.php?ids=results&id=5\_TE0820\_AI\_3\_0

[6]

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Description of compilation of Vitis AI 3.0 models for different configurations of AMD DPUs, (Application note, with evaluation package, UTIA). Published for free public access from: https://zs.utia.cas.cz/index.php?ids=results&id=6 TE AI 3 0

61/61

