department of signal processing

https://sp.utia.cas.cz

Application Note

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

Support for TE0820 Modules in Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8G

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

Revision history

Rev.	Date	Author	Description
v01	19.5.2024	J.K.	Vitis 2023.2, Petalinux 2023.2, AI 3.5 runtime, AI 3.0 models for AMD DPU DPUCZDX8G
v02	5.12.2024	J.K.	Update for bring-up script for te0820 released by Trenz electronic 25.11.2024 in: TE0820-test_board-vivado_2023.2-build_4_20241125214948.zip
v03	22.12.2024	J.K.	Updated text related to module product changes.
v04	15.02.2025	J.K.	Updated typos
v05	16.02.2025	J.K.	Updated references

© 2024, 2025 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved.

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 ... 4
2 Prepare Reference Design for Extensible Custom Platform ... 9
2.1 Reference HW for TE0820 module ... 9
3 HW support for Vitis Extensible Design Flow ... 11
3.1 Create Extensible platform HW ... 11
3.2 Fast Track for Creation of Extensible platform HW ... 20
3.3 Validate Design ... 20
3.4 Compile Created HW and Custom SW with Trenz Scripts ... 21
3.5 Copy Created Custom First Stage Boot Loader ... 22
4 Building Petalinux for Vitis AI 3.0 and AI 3.5 SW Library ... 23
4.1 Vitis AI3.0 models and Vitis AI 3.5 library ... 23
4.2 Building Petalinux for Extensible Design Flow ... 23
4.3 Disable CPU IDLE in Kernel Config ... 27
4.4 Add EXT4 rootfs Support ... 27
4.5 Let Linux Use EXT4 rootfs During Boot ... 28
4.6 Build PetaLinux Image ... 28
4.7 Create Petalinux SDK ... 28
4.8 Copy Files for Extensible Platform ... 29
4.9 Create Extensible Platform Archive ... 30
4.10 Clean Petainux Files (optional) ... 31
4.11 Generation of SYSROOT ... 31
4.12 Generation of Extensible Platform for Vitis ... 32
5 Platform Usage ... 34
5.1 Read Platform Info ... 34
5.2 Create and Compile Vector Addition Example ... 36
5.3 Run Compiled test_vadd Example Application ... 39
5.4 Display on X11 terminal ... 40
6 Vitis AI 3.0 DPUCZDX8G_VAI_v3.0 Installation ... 41
6.1 Create and Build Vitis Design ... 42
6.2 Configure Project for the Vitis Extensible Flow with DPU ... 42
6.3 Configure Connection of DPU kernel ... 44
6.4 Build the test_dpu_trd Project ... 45
7 Prepare SD card with test_dpu_trd DPU ... 46
7.1 Resize EXT4 Partition ... 47
7.2 Test the Integrated DPUCZDX8G ... 47
7.3 Test resnet50_pt model ... 48
7.4 Remote Monitoring and Configuration Support ... 50
7.5 Remote Control from Ubuntu X11 Desktop. ... 51
7.6 Remote Control in x-session-manager on Ubuntu X11 Desktop. ... 52
7.7 Display Test Pattern and Test USB Camera ... 52
7.8 TE0820-03-04EV-1EA Module ID=23, TE0701-06, DPU (B4096) ... 54
8 References ... 55

© 2024, 2025 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

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/

© 2024, 2025 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

1 Introduction

EECONE project https://eecone.com/eecone/home/ 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, AI 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].
- Module based systems. See [4], [5] and [8], [9].
- [9] is this application note. It is update of [5].
- [5] was supporting system bring-up in AMD Vitis 2022.2 tool chain.
- This application note [9] is supporting system bring-up in AMD Vitis 2023.2 tool chain.

Both compared types of systems (low cost: [2], [3]) and (modular: [4], [5], [8], [9]) 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.

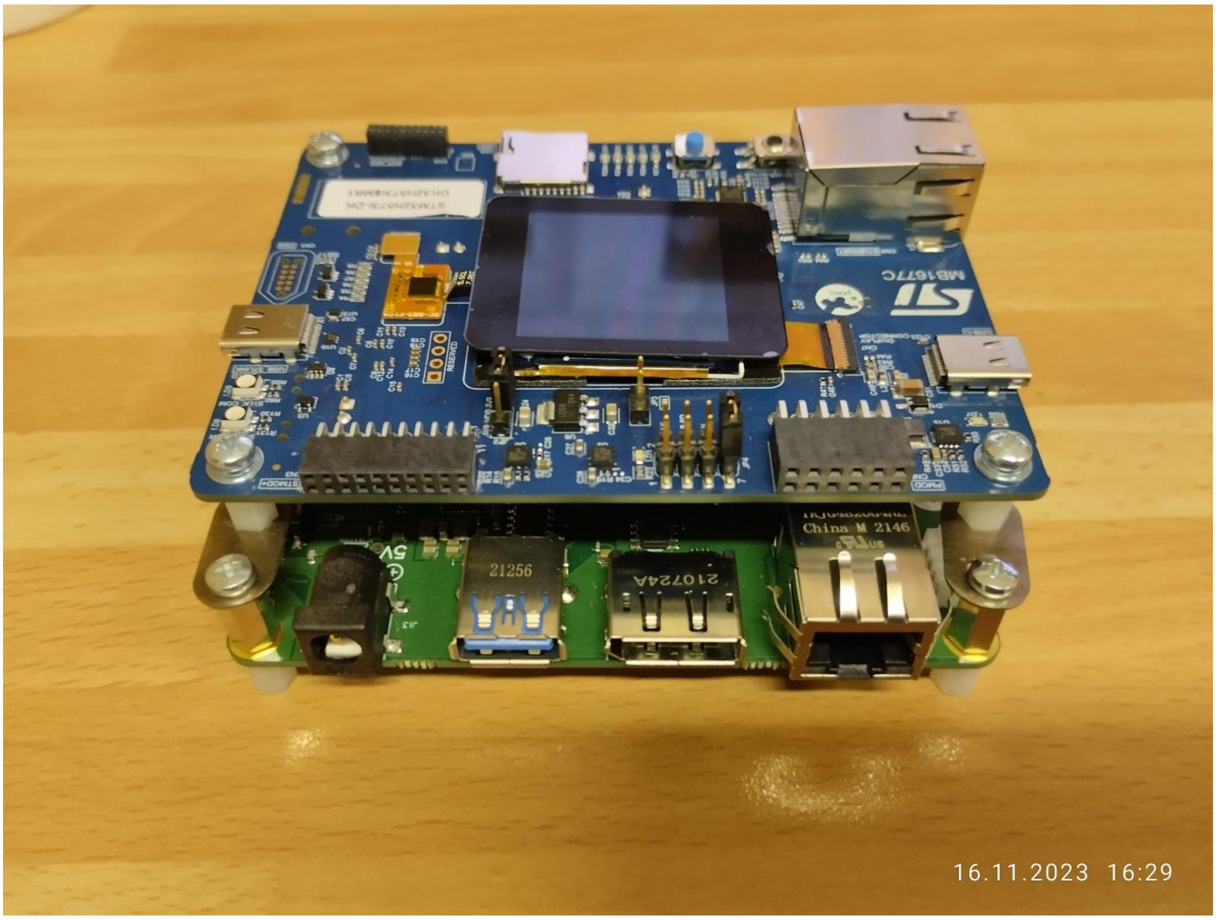
Compared systems use 16nm AMD ZynqUltrascale+ device with 64 bit ARM A53 Microprocessor and programmable logic in the same device and Petalinux OS.

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

- 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

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

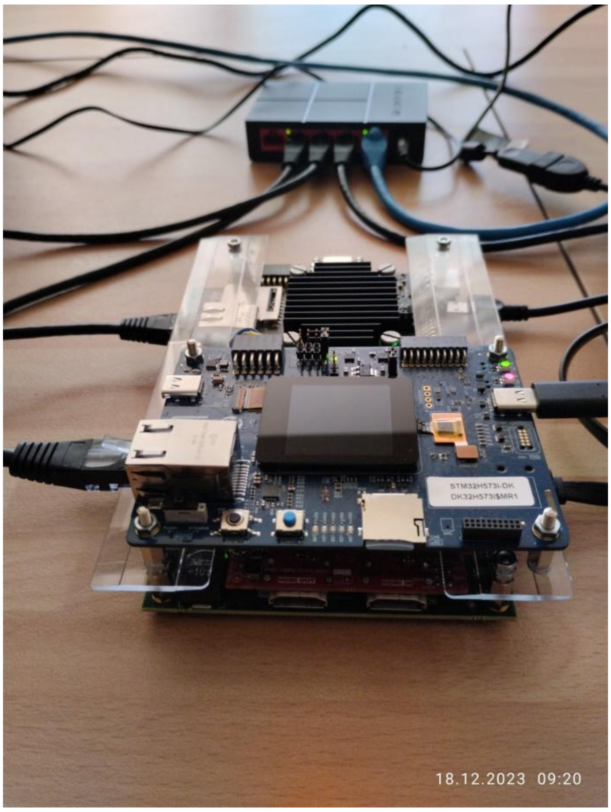
[1]	STM32H573I-DK	https://www.st.com/en/evaluation-tools/stm32h573i-dk.html	Local or remote system control (www-server or secure mqtt client) for [2], [3]
[2]	TE0802-02-1BEV2-A	https://shop.trenz-electronic.de/en/TE0802-02-1BEV2-A-MPSoC-Development-Board-with-AMD-Zynq-UltraScale-ZU1EG-and-1-GB-LPDDR4?c=474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

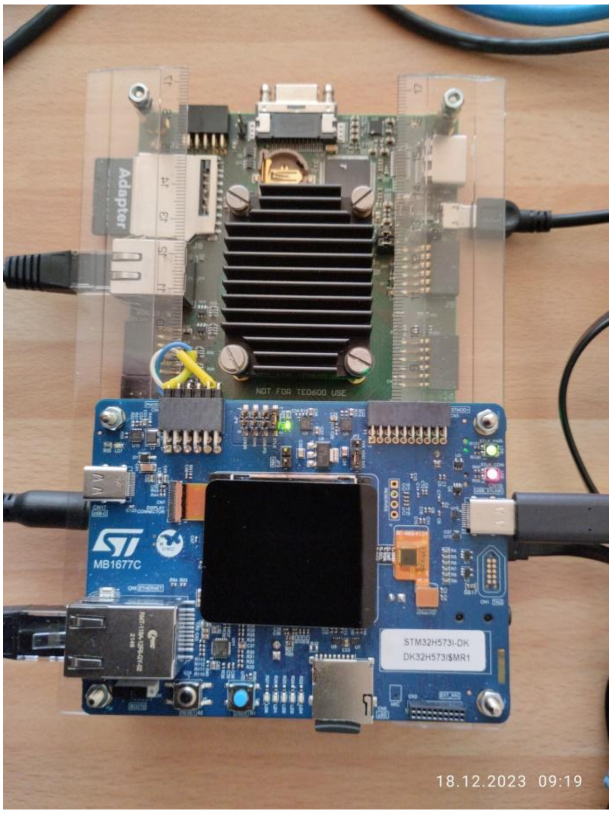


© 2024 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

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

[1] **[7]**	STM32H573I-DK TE0701-06 Carrier Board for Trenz Electronic 4 x 5 Modules TE0821 or TE0820	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 2-1, 2-2 Carrier Board for range of 4x5 cm modules [3], [4].
[4] **[8]**	TE0821 Module: **17 module types** **24 module types**	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0821-Zynq-UltraScale/	AMD Vitis AI 3.5 AMD DPU in PL USB camera remote X11 desktop
[5] **[9]**	TE0820 Module: **115 module types** **119 module types**	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0820-Zynq-UltraScale/	AMD Vitis AI 3.5 AMD DPU in PL USB camera remote X11 desktop





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

This application note [9] and the accompanying evaluation package describe support for systems based on TE0820 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 wiki tutorial on Trenz Electronic wiki server:
https://wiki.trenz-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 [9]**.**

This application note describes how to design custom HW platform with AMD DPU for Vitis 2023.2 AI 3.5 runtime inference for family of Trenz Electronic modules TE0820 with AMD Zynq Ultrascale+ device.

This application note [9] is using AMD Vitis 2023.2 and PetaLinux 2023.2 tools installed on Ubuntu 20.04. The described configuration integrated AMD DPU IP, version v4.1.0, with architecture DPUCZDX8G present in Vitis AI 3.0 distribution with corresponding AI models.

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 [9].

The installed AMD DPU configurations require recompilation of Vitis AI 3.5 SW of examples and use the inference models present in the Vitis AI 3.0 framework. This compilation process will be described in separate application note [10].

This application supports family of TE0820 modules listed in next tables with ID 15 to 127.

Process will be demonstrated for these TE0820 modules:

- ID=23 module: TE0820-03-04EV-1EA, device xczu4ev-sfvc784-1-e, 2GB DDR4
- ID=70 module: TE0820-04-4DE21FA, device xczu4ev-sfvc784-1-e, 2GB DDR4
- ID=107 module: TE0820-05-4AE21MA, device xczu4cg-sfvc784-1-e, 2GB DDR4

- ID=89 module: TE0820-05-2AE21MA, device xczu2cg-sfvc784-1-e, 2GB DDR4
- ID=103 module: TE0820-05-3BE81MA, device xczu3eg-sfvc784-1-e, 2GB DDR4

Modules with identical AMD device and identical size of DDR4 memory also envolve in time. It is mainly due to required replacement of some end-of-life components like the DDR4 devices or DC2DC convertors.

Example of time evolution of modules:
TE0820-03-04EV-1EA was produced by Trenz Electronic from **8.2018** to **8.2020**.
TE0820-04-4DE21FA was produced by Trenz Electronic from **8.2020** to **8.2022**.
TE0820-05-4AE21MA is produced by Trenz Electronic from **8.2022** to **present.**

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

HW changes of modules are described in these product change notifications:

PCN-20171117 TE0820-02 SPI Flash Change.pdf	21.11.2017
PCN-20180816 TE0820-02 to TE0820-03 Hardware Revision Change.pdf	10.09.2018
PCN-20190110a TE0820-03- SPI Flash and eMMC Change.pdf	16.02.2021
PCN-20200616 TE0820-03 to TE0820-04 Hardware Revision Change.pdf	09.09.2020
PCN-20210615 TE0820-04 eMMC Change and Product Update.pdf	27.09.2021
PCN-20220725 TE0820-04 to TE0820-05 Hardware Revision Change.pdf	04.08.2022
PCN-20231007 TE0820-05 DDR4 SDRAM and DCDC Change.pdf	23.10.2023

TE0820-03-04EV-1EA and TE0820-04-4DE21FA have four A53 ARM cores and contain BRAM and also URAM blocks in the PL part of the device. It is possible to implement all possible configurations of the AMD DPU (from B512 up to B4096) in PL. Implementaton of B4096 is demonstrated.

TE0820-05-4AE21MA has two A53 ARM cores and contains BRAM and also URAM blocks in the PL part of the device. It is possible to implement all possible configurations of the AMD DPU (from B512 up to B4096) in PL. Implementaton of B4096 is demonstrated.

TE0820-03-04EV-1EA has two A53 ARM cores and contains only BRAM blocks in relatively small PL part of the device. It is possible to implement configuration of the AMD DPU from B512 to B4096. Implementaton of B4096 is demonstrated.

TE0820-05-3BE81MA has four A53 ARM cores and contains only BRAM blocks in the PL part of the device. Therefore, it is possible to implement configurations of the AMD DPU (from B512 up to B1600) in PL. Implementaton of B1600 is demonstrated.

Specification for each module ID defined in `TE0820_board_files.csv` file is input to the Vivado 2023.2 HW bring-up scripts. It is provided by the company Trenz Electronic. It is part of package provided by Trenz Electronic for supported family of modules TE0820 for AMD Vivado 2023.2 design flows.

List of all supported TE0820 modules is reprinted from the `TE0820_board_files.csv` file included in the evaluation package associated to this application note. This application note and associated evaluation package enables support for “second-life” of 119 versions of TE0820 modules.

```
CSV_VERSION=1.4
#Comment:-do not change matrix position or remove CSV_VERSION:
```

ID	PRODID	PARTNAME	BOARDNAME
15	TE0820-03-02CG-1EA	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
16	TE0820-03-02CG-1ED	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
17	TE0820-03-02EG-1EA	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
18	TE0820-03-02EG-1EL	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
19	TE0820-03-03CG-1EA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
20	TE0820-03-03EG-1EA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
21	TE0820-03-03EG-1EL	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
22	TE0820-03-04CG-1EA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
23	TE0820-03-04EV-1EA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0

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

24	TE0820-03-2AE21FA	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
25	TE0820-03-2AI21FA	xczu2cg-sfvc784-1-i	trenz.biz:te0820_2cg_1i:part0:2.0
26	TE0820-03-2BE21FA	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
27	TE0820-03-2BE21FL	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
28	TE0820-03-2BI21FA	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
29	TE0820-03-2BI21FL	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
30	TE0820-03-3AE21FA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
31	TE0820-03-3AI210A	xczu3cg-sfvc784-1-i	trenz.biz:te0820_3cg_1i:part0:2.0
32	TE0820-03-3AI21FA	xczu3cg-sfvc784-1-i	trenz.biz:te0820_3cg_1i:part0:2.0
33	TE0820-03-3BE21FA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
34	TE0820-03-3BE21FL	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
35	TE0820-03-4AE21FA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
36	TE0820-03-4AI21FI	xczu4cg-sfvc784-1-i	trenz.biz:te0820_4cg_1i:part0:3.0
37	TE0820-03-4DE21FA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
38	TE0820-03-4DE21FC	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
39	TE0820-03-4DE21FL	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
40	TE0820-03-4DI21FA	xczu4ev-sfvc784-1-i	trenz.biz:te0820_4ev_1i:part0:2.0
41	TE0820-03-5DI21FA	xczu5ev-sfvc784-1-i	trenz.biz:te0820_5ev_1i:part0:2.0
42	TE0820-03-5DR21FA	xazu5ev-sfvc784-1Q-q	trenz.biz:te0820_5ev_1q:part0:2.0
43	TE0820-04-2AE21FA	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
44	TE0820-04-2AI21FA	xczu2cg-sfvc784-1-i	trenz.biz:te0820_2cg_1i:part0:2.0
45	TE0820-04-2AI21MC	xczu2cg-sfvc784-1-i	trenz.biz:te0820_2cg_1i:part0:2.0
46	TE0820-04-2BE21-V1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
47	TE0820-04-2BE21FA	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
48	TE0820-04-2BE21FAJ	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
49	TE0820-04-2BE21FL	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
50	TE0820-04-2BE21MA	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
51	TE0820-04-2BE21MAJ	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
52	TE0820-04-2BI21FA	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
53	TE0820-04-2BI21FL	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
54	TE0820-04-2BI21MA	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
55	TE0820-04-2BI21ML	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
56	TE0820-04-3AE21FA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
57	TE0820-04-3AE21MA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
58	TE0820-04-3AI21FA	xczu3cg-sfvc784-1-i	trenz.biz:te0820_3cg_1i:part0:2.0
59	TE0820-04-3AI21FAT	xczu3cg-sfvc784-1-i	trenz.biz:te0820_3cg_1i:part0:2.0
60	TE0820-04-3BE21FA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
61	TE0820-04-3BE21FL	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0

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

62	TE0820-04-3BE21KA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
63	TE0820-04-3BE21MA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
64	TE0820-04-3BE21ML	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
65	TE0820-04-3BE21MLZ	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
66	TE0820-04-4AE21FA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
67	TE0820-04-4AE21MA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
68	TE0820-04-4AI21FI	xczu4cg-sfvc784-1-i	trenz.biz:te0820_4cg_1i:part0:3.0
69	TE0820-04-4BI21KL	xczu4eg-sfvc784-1-i	trenz.biz:te0820_4eg_1i:part0:2.0
70	TE0820-04-4DE21FA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
71	TE0820-04-4DE21FL	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
72	TE0820-04-4DE21MA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
73	TE0820-04-4DI21FA	xczu4ev-sfvc784-1-i	trenz.biz:te0820_4ev_1i:part0:2.0
74	TE0820-04-4DI21MA	xczu4ev-sfvc784-1-i	trenz.biz:te0820_4ev_1i:part0:2.0
75	TE0820-04-5DI21FA	xczu5ev-sfvc784-1-i	trenz.biz:te0820_5ev_1i:part0:2.0
76	TE0820-04-5DI21MA	xczu5ev-sfvc784-1-i	trenz.biz:te0820_5ev_1i:part0:2.0
77	TE0820-04-5DR21FA	xazu5ev-sfvc784-1Q-q	trenz.biz:te0820_5ev_1q:part0:2.0
78	TE0820-04-S002	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
79	TE0820-04-S002C1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
80	TE0820-04-S003	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
81	TE0820-04-S004	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
82	TE0820-04-S005	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
83	TE0820-04-S006	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
84	TE0820-04-S009	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
85	TE0820-04-S010	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
86	TE0820-04-S013	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
87	TE0820-04-S016	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
88	TE0820-04-S018	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
89	TE0820-05-2AE21MA	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
90	TE0820-05-2AE21MAZ	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
91	TE0820-05-2AE81MA	xczu2cg-sfvc784-1-e	trenz.biz:te0820_2cg_1e:part0:2.0
92	TE0820-05-2AI21MA	xczu2cg-sfvc784-1-i	trenz.biz:te0820_2cg_1i:part0:2.0
93	TE0820-05-2AI81MA	xczu2cg-sfvc784-1-i	trenz.biz:te0820_2cg_1i:part0:2.0
94	TE0820-05-2BE21MA	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
95	TE0820-05-2BE21MAJ	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
96	TE0820-05-2BI21MA	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
97	TE0820-05-2BI81MA	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
98	TE0820-05-2BI81ML	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0
99	TE0820-05-3AE21MA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0

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

100	TE0820-05-3AE81MA	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
101	TE0820-05-3BE21MA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
102	TE0820-05-3BE21MAZ	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
103	TE0820-05-3BE81MA	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
104	TE0820-05-3BE81ML	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
105	TE0820-05-3BI21ML	xczu3eg-sfvc784-1-i	trenz.biz:te0820_3eg_1i:part0:2.0
106	TE0820-05-3BI81ML	xczu3eg-sfvc784-1-i	trenz.biz:te0820_3eg_1i:part0:2.0
107	TE0820-05-4AE21MA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
108	TE0820-05-4AE81MA	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
109	TE0820-05-4AI21MI	xczu4cg-sfvc784-1-i	trenz.biz:te0820_4cg_1i:part0:3.0
110	TE0820-05-4BE81MA	xczu4eg-sfvc784-1-e	trenz.biz:te0820_4eg_1e:part0:2.0
111	TE0820-05-4BI21PL	xczu4eg-sfvc784-1-i	trenz.biz:te0820_4eg_1i:part0:2.0
112	TE0820-05-4BI21PLZ	xczu4eg-sfvc784-1-i	trenz.biz:te0820_4eg_1i:part0:2.0
113	TE0820-05-4DE21MA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
114	TE0820-05-4DE81MA	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
115	TE0820-05-4DI21MA	xczu4ev-sfvc784-1-i	trenz.biz:te0820_4ev_1i:part0:2.0
116	TE0820-05-4DI81MA	xczu4ev-sfvc784-1-i	trenz.biz:te0820_4ev_1i:part0:2.0
117	TE0820-05-5DI21MA	xczu5ev-sfvc784-1-i	trenz.biz:te0820_5ev_1i:part0:2.0
118	TE0820-05-5DI81MA	xczu5ev-sfvc784-1-i	trenz.biz:te0820_5ev_1i:part0:2.0
119	TE0820-05-S002C1	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
120	TE0820-05-S003	xczu4ev-sfvc784-1-e	trenz.biz:te0820_4ev_1e:part0:2.0
121	TE0820-05-S004C1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
122	TE0820-05-S005	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
123	TE0820-05-S008C1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
124	TE0820-05-S010C1	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
125	TE0820-05-S013	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
126	TE0820-05-S014C1	xczu4cg-sfvc784-1-e	trenz.biz:te0820_4cg_1e:part0:2.0
127	TE0820-05-S016	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
128	TE0820-05-S017C1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
129	TE0820-05-S018	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
130	TE0820-05-S020	xczu3eg-sfvc784-1-e	trenz.biz:te0820_3eg_1e:part0:2.0
131	TE0820-05-S022	xczu3cg-sfvc784-1-e	trenz.biz:te0820_3cg_1e:part0:2.0
132	TE0820-05-S024C1	xczu2eg-sfvc784-1-e	trenz.biz:te0820_2eg_1e:part0:2.0
133	TE0820-05-S025	xczu2eg-sfvc784-1-i	trenz.biz:te0820_2eg_1i:part0:2.0

Supported modules with ID = 15 ... 133

Modules from this list mighed have been used originally in another context which might become obsolete, now. We provide support to reuse this module again in range of Vitis 2022 AI 3.0 [5], [6] and Vitis 2023.2.1 AI 3.5 [9], [10] applications.

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

2 Prepare Reference Design for Extensible Custom Platform

The design proces is demonstrated for module with ID=23: TE0820-03-04EV-1EA, device xczu4ev-sfvc784-1-e, 2GB DDR4. If your module has different ID, replace 23 with that ID.

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

```
$ source /tools/Xilinx/Vitis/2023.2/settings64.sh
```

Download TE0820 test_board Linux Design file for Vitis 2023.2 from:

```
https://shop.trenz-electronic.de/trenzdownloads/Trenz_Electronic/Modules_and_Module_Carriers/4x5/TE0820/Reference_Design/2023.2/test_board/TE0820-test_board-vivado_2023.2-build_4_20241125214948.zip
```

This TE0820 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/te0820_23_240
```

All supported modules are identified in file:

```
~/work/te0820_23_240/test_board/board_files/TE0820_board_files.csv
```

We will select module 23 with name TE0820-03-04EV-1EA, with device xczu04ev-sfvc784-1-e on TE0701-06 carrier board. We will use default clock 240 MHz. That is why we name the package te0820_23_240 and proposed to unzip the TE0820 test_board Linux Design files into the directory:

```
~/work/te0820_23_240
```

2.1 Reference HW for TE0820 module

In Ubuntu terminal, change directory to the test_board directory:

```
$ cd ~/work/te0820_23_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
```


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

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

```
devel@ubuntu: ~/work/te0820_89_240/test_board

devel@ubuntu:~/work/te0820_89_240/test_board$ source /tools/Xilinx/Vitis/2023.2/settings64.sh
devel@ubuntu:~/work/te0820_89_240/test_board$ chmod ugo+rwx ./console/base_sh/*.sh
devel@ubuntu:~/work/te0820_89_240/test_board$ chmod ugo+rwx ./_create_linux_setup.sh
devel@ubuntu:~/work/te0820_89_240/test_board$ ./_create_linux_setup.sh
------------------------Set design paths----------------------------
-- Run Design with: _create_linux_setup.sh
-- Use Design Path: /home/devel/work/te0820_89_240/test_board
--------------------------------------------------------------------
----------------------_create_linux_setup.cmd-----------------------
------------------------TE Reference Design-------------------------
--------------------------------------------------------------------
-- (d)  Go to Documentation (Web Documentation)
-- (x)  Exit Batch (nothing is done!)
-- (0)  Module selection guide, project creation...
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start  design with unsupported Vivado Version (beta)
----
 Select (ex.:'0' for module selection guide):
0
```

Select variant 23 from the selection guide, press enter and agree selection

```
devel@ubuntu: ~/work/te0820_89_240/test_board

----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
--------------
------------------------------------------------------------------------
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(filter 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:
89
```

Create Vivado Project with option 1

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

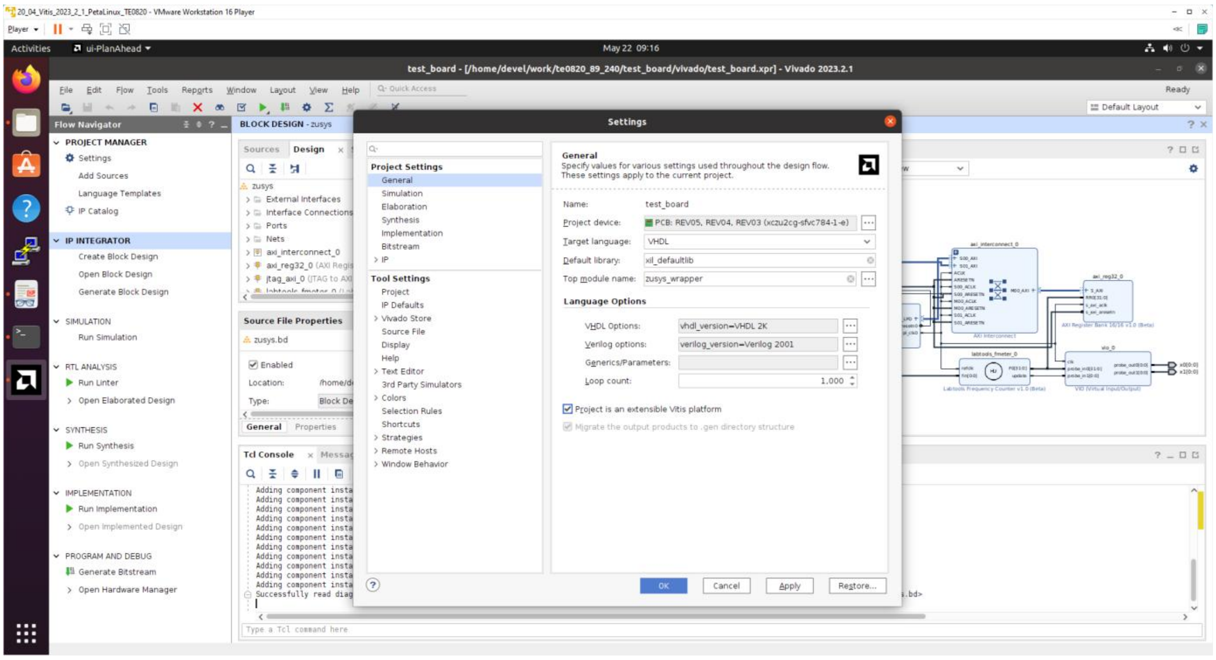
Vivado Project will be generated for the selected variant.

3 HW support for Vitis Extensible Design Flow

3.1 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 3.2.

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`.



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

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.

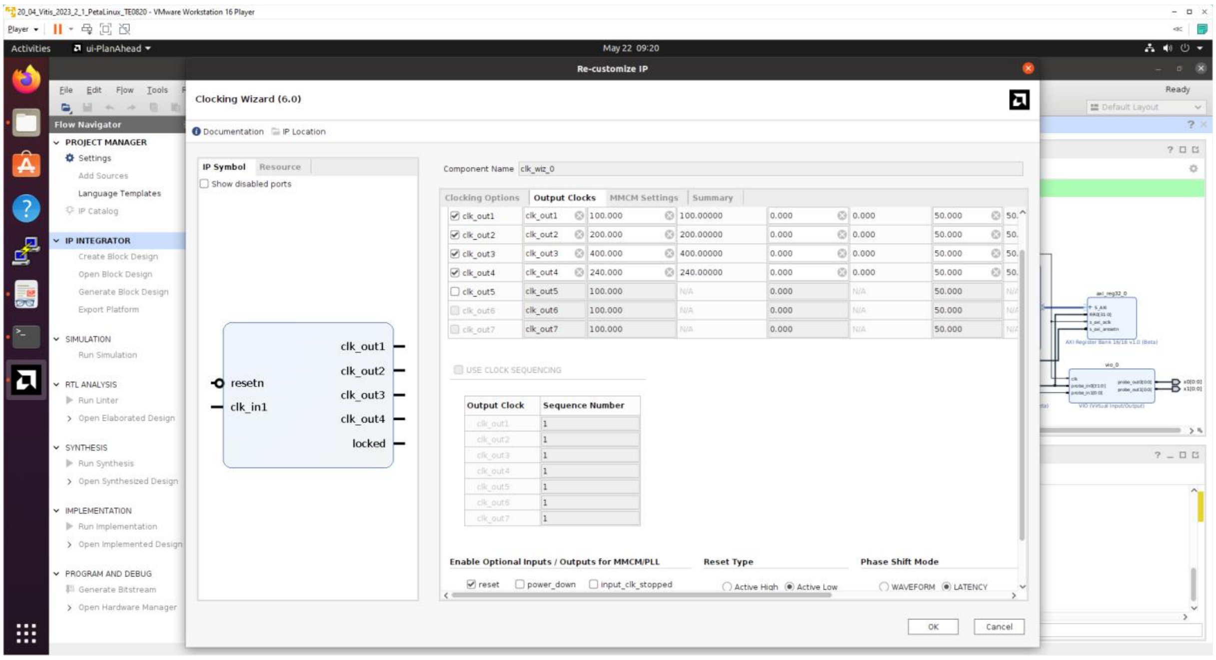
100 MHz clock will serve as low speed clock.

200 MHz and 400 MHz clock will serve as clock for the AMD DPU AI 3.0 HW IP.

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

400 MHz clock will serve as high speed clock.

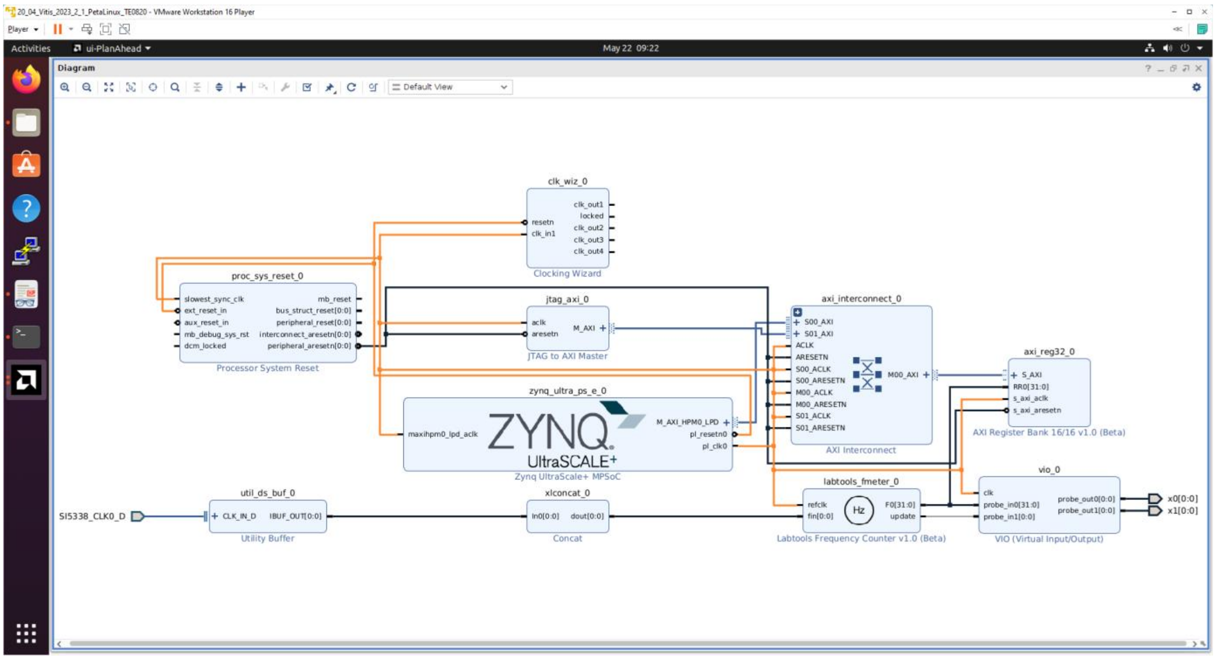
Set reset type from the default `Active High` to `Active Low`.



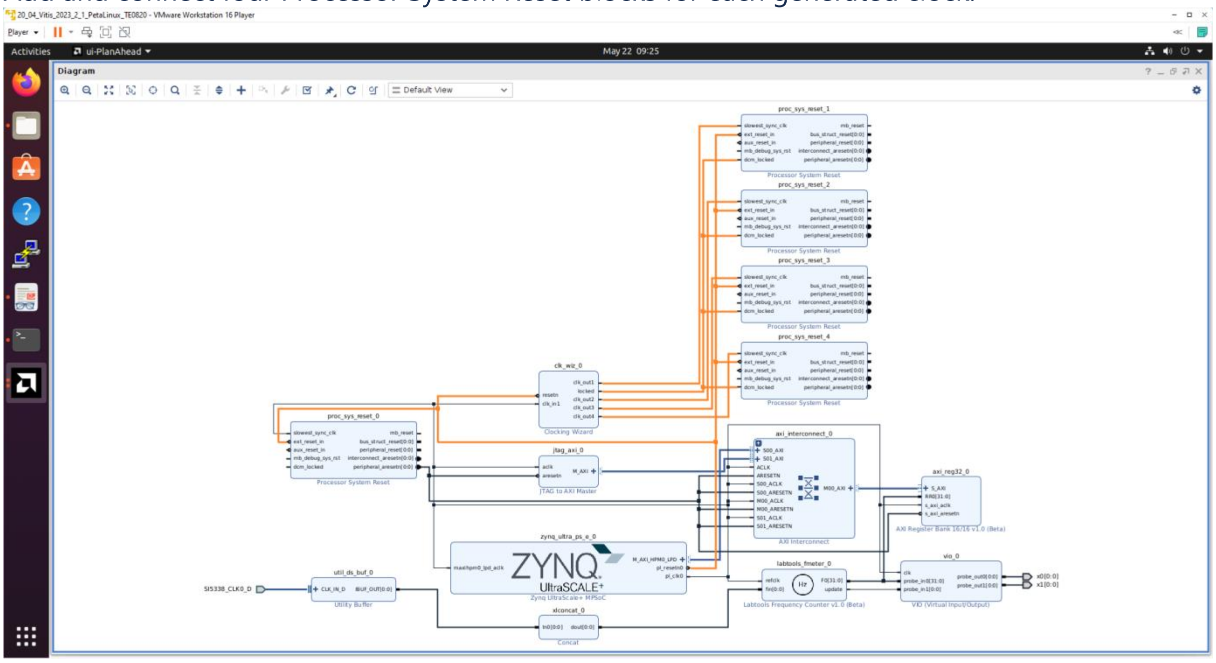
Clik on OK to close the Re-customize IP window.

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`.

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



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



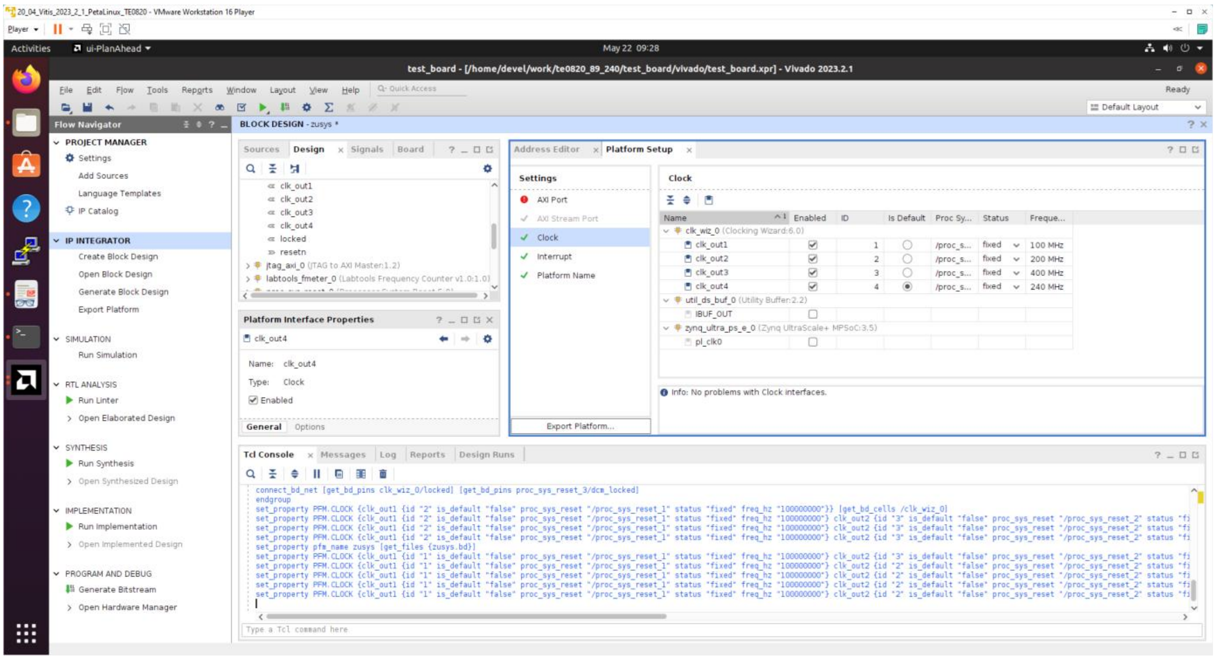
Open `Platform Setup` window of IP Integrator to define Clocks. In `Settings`, select `Clock`.

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`

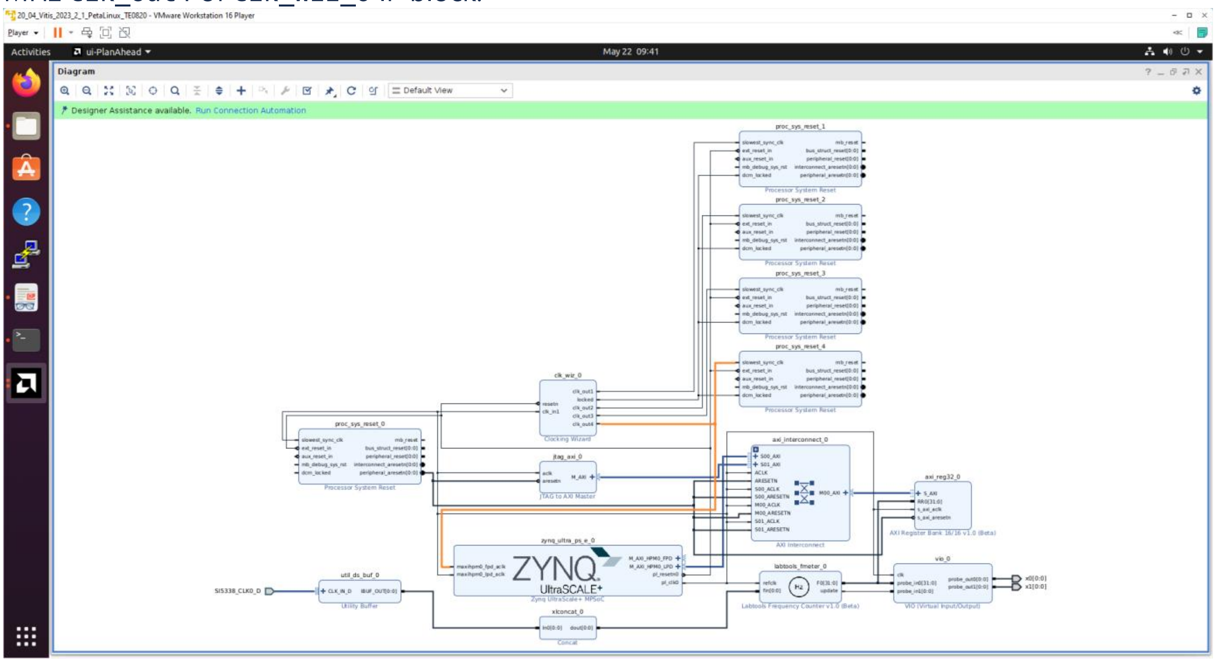
In `Is Default column`, select `clk_out4` (with `ID=4`) as the default clock. One and only one clock must be selected as default clock.

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



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 interrupt 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.



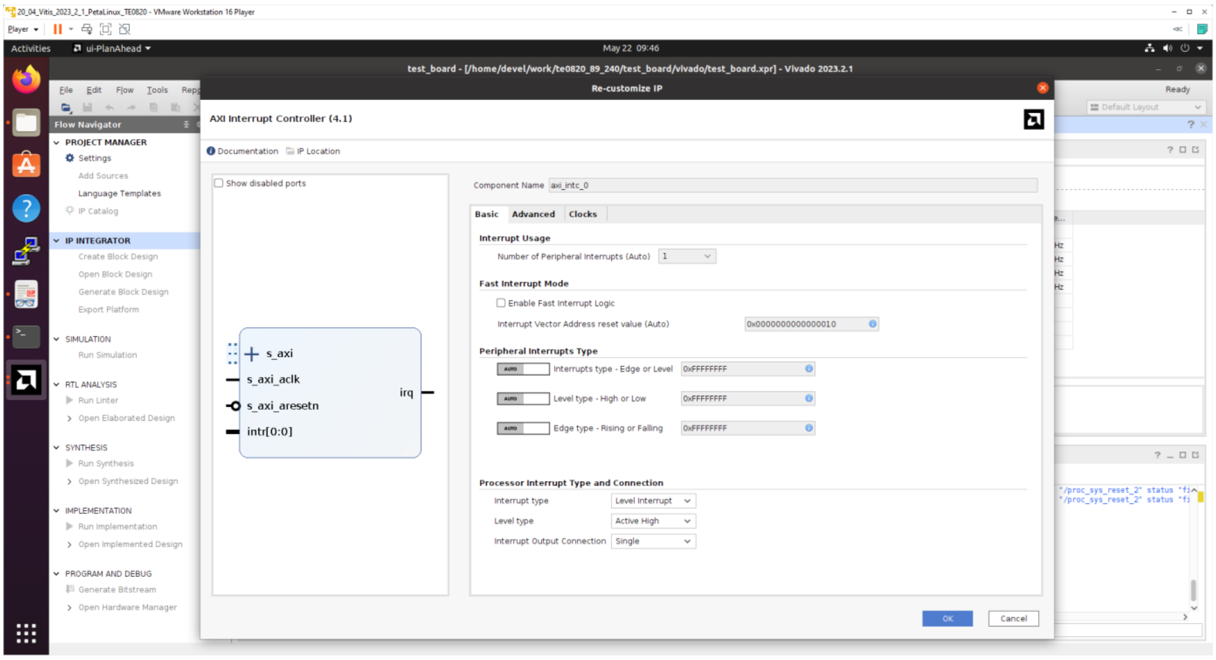
Add, customize and connect the AXI Interrupt Controller

Add AXI Interrupt Controller IP `axi_intc_0`.
Double-click on `axi_intc_0` to re-customize it.

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

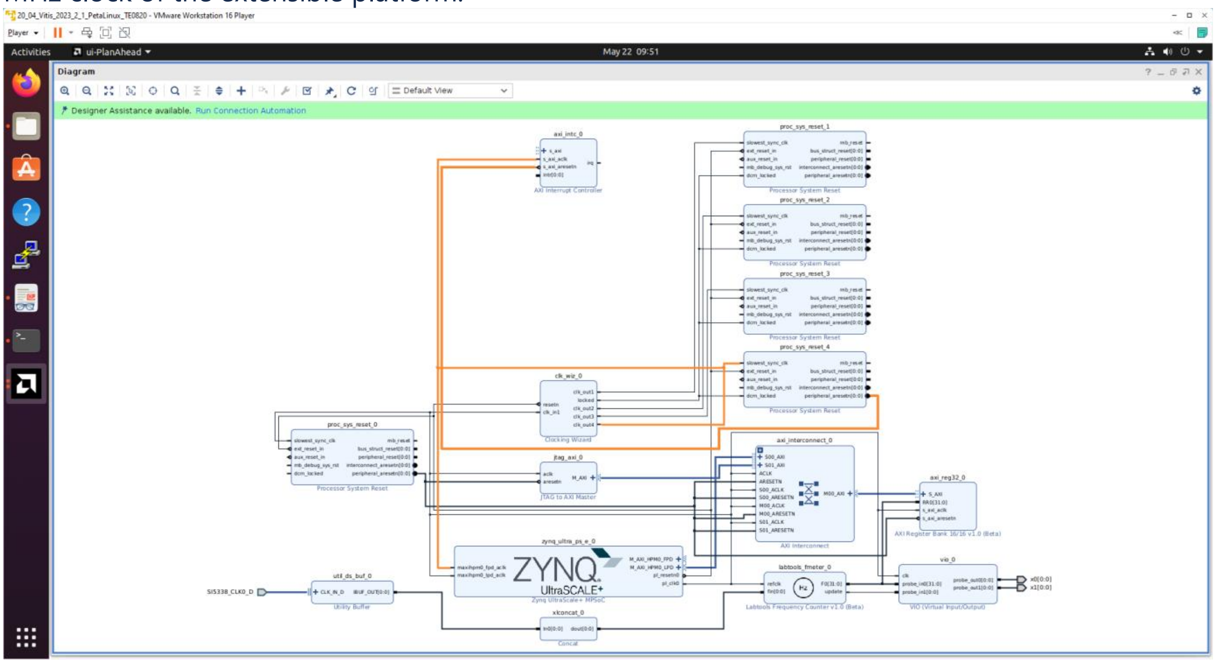
In `Processor Interrupt Type and Connection` section select the `Interrupt Output Connection` from `Bus` to `Single`.

Click on `OK` to accept these changes.



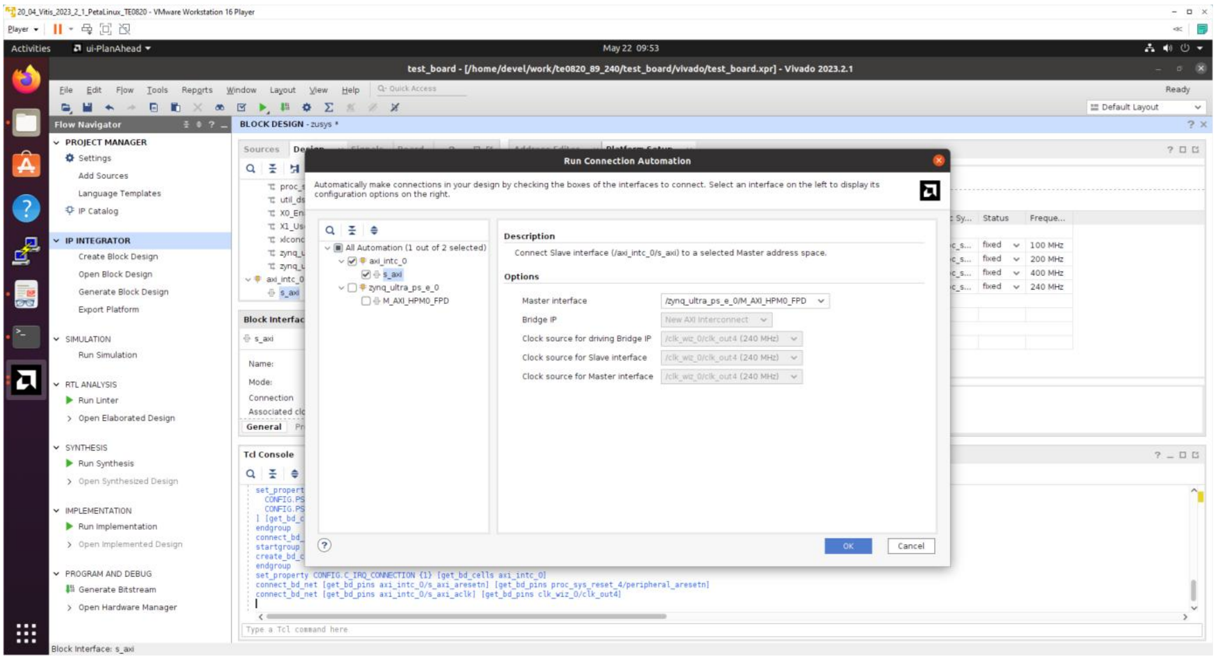
Connect interrupt controller clock input `s_axi_aclk` of `axi_intc_0` to output `dlk_out4` of `clk_wiz_0`. It is the default, 240 MHz clock of the extensible platform.

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.



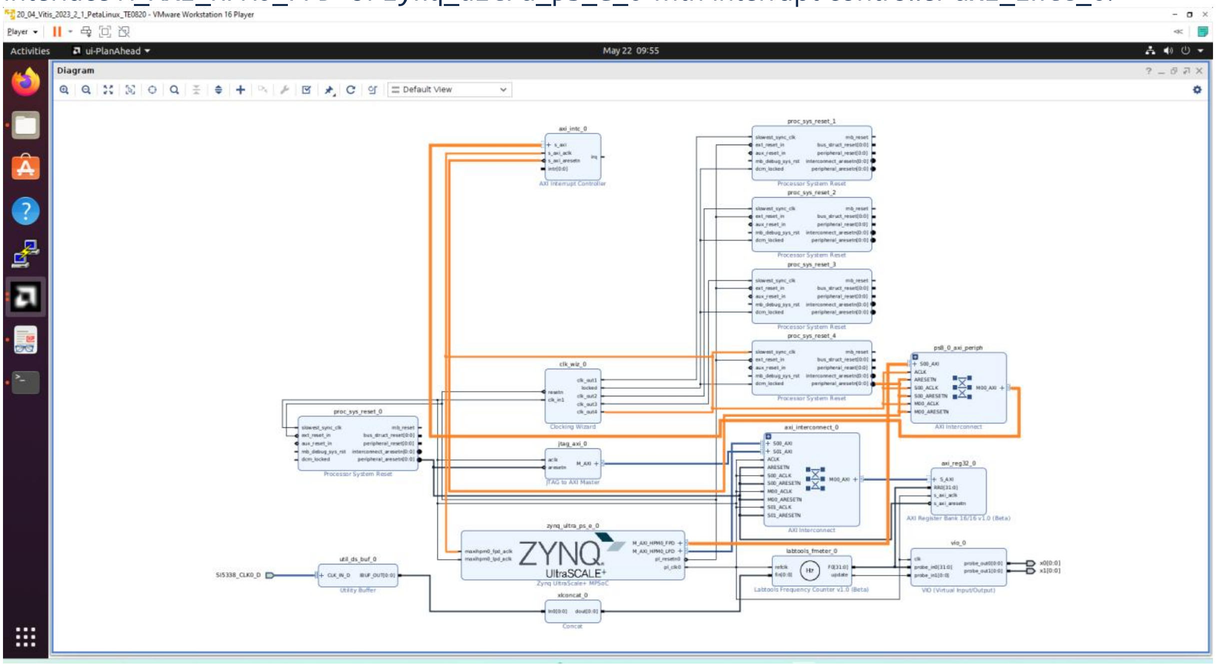
Use the `Run Connection Automation` wizard to connect the axi lite interface of interrupt controller `axi_intc_0` to master interface `M_AXI_HPM0_FPD` of `zynq_ultra_ps_e_0`.

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



In Run Connection Automaton window, click `OK`.

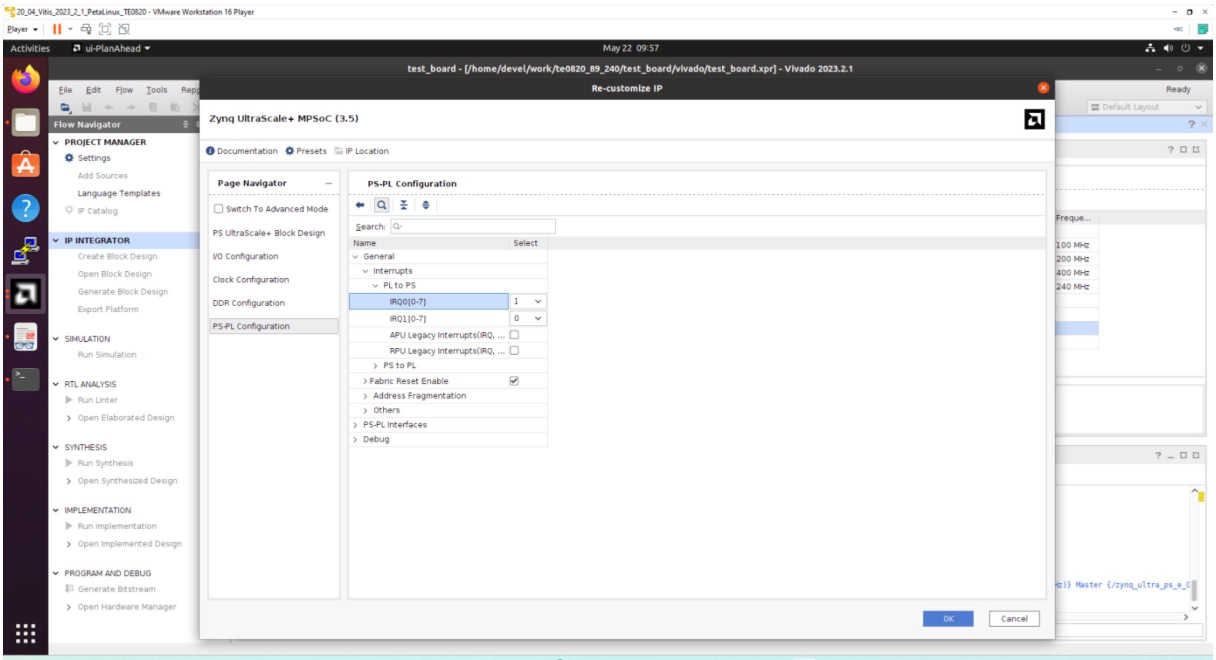
New AXI interconnect `ps_8_axi_periph` is created. It connects master interface `M_AXI_HPM0_FPD` of `zynq_ultra_ps_e_0` with interrupt controller `axi_intc_0`.



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.

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

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



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.

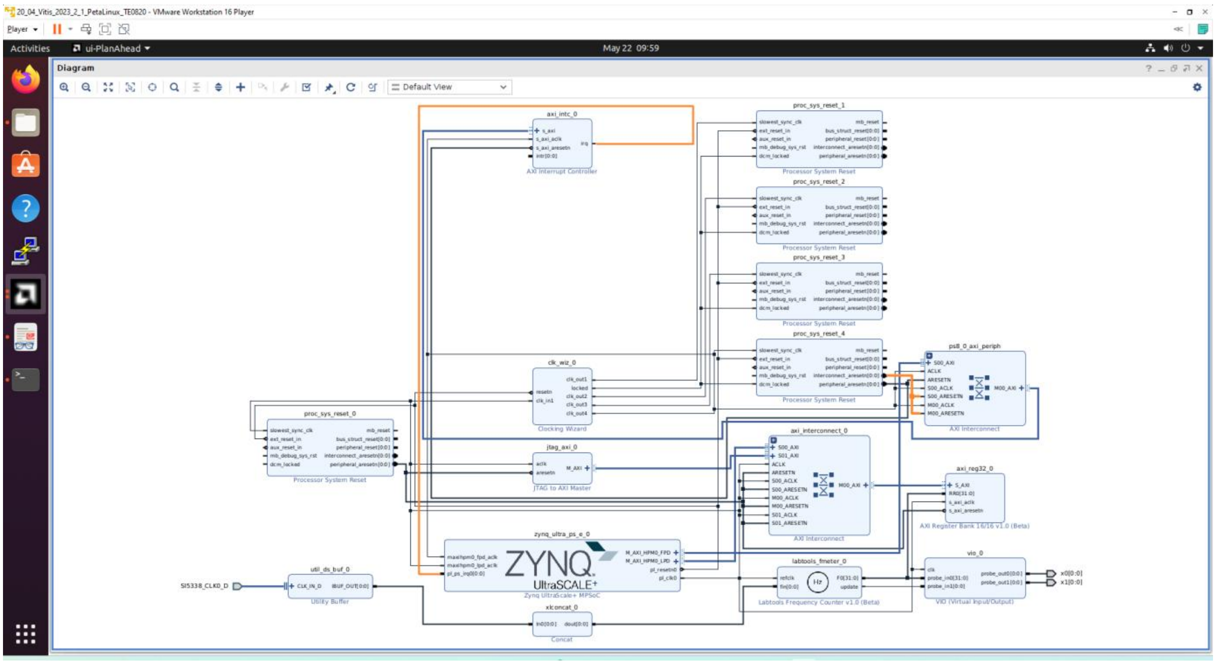
Disconnect input `M00_ARESETN` of `ps_8_axi_periph` block from the network driven by output `peripherial_aresetn[0:0]` of `proc_sys_reset_4` block.

Connect input `M00_ARESETN` of `ps_8_axi_periph` to output`interconnect_aresetn[0:0]` of `proc_sys_reset_4` block.

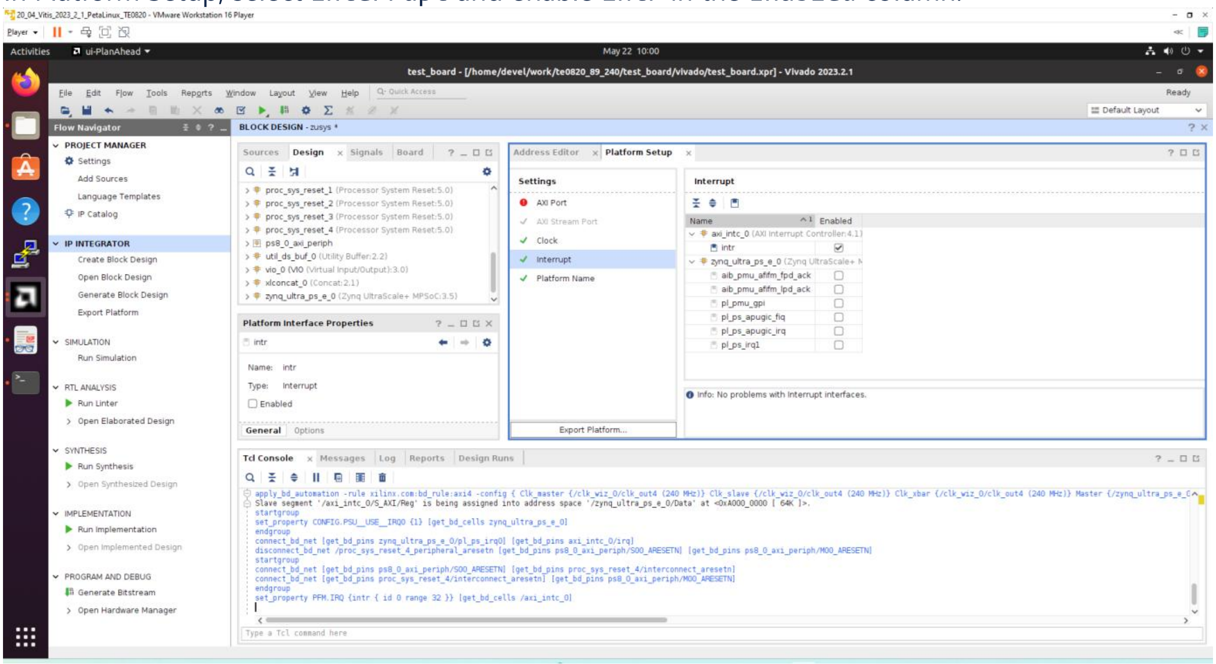
This modification will make the reset structure of the AXI interconnect `ps_8_axi_periph` block identical to the future extensions of this interconnect generated by the Vitis extensible design flow.

Connect the interrupt input `pl_ps_irq0[0:0]` of `zynq_ultra_ps_e_0` block with output `irq` of `axi_intc_0` block.

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



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



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`:

Select `M_AXI_HPM1_FPD` in column `Enabled`.

Select `S_AXI_HPC0_FPD` and `S_AXI_HPC1_FPD` in column “Enabled”.

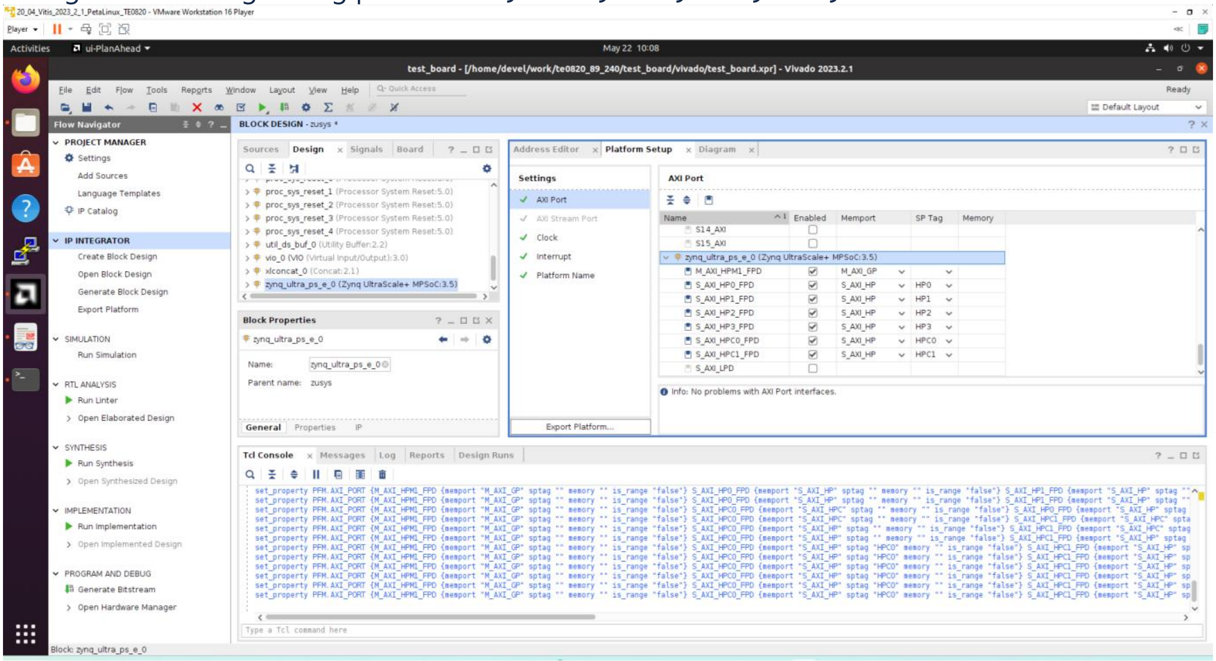
For `S_AXI_HPC0_FPD`, change `S_AXI_HPC` to `S_AXI_HP` in column `Memport`.

For `S_AXI_HPC1_FPD`, change `S_AXI_HPC` to `S_AXI_HP` in column `Memport`.

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

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`



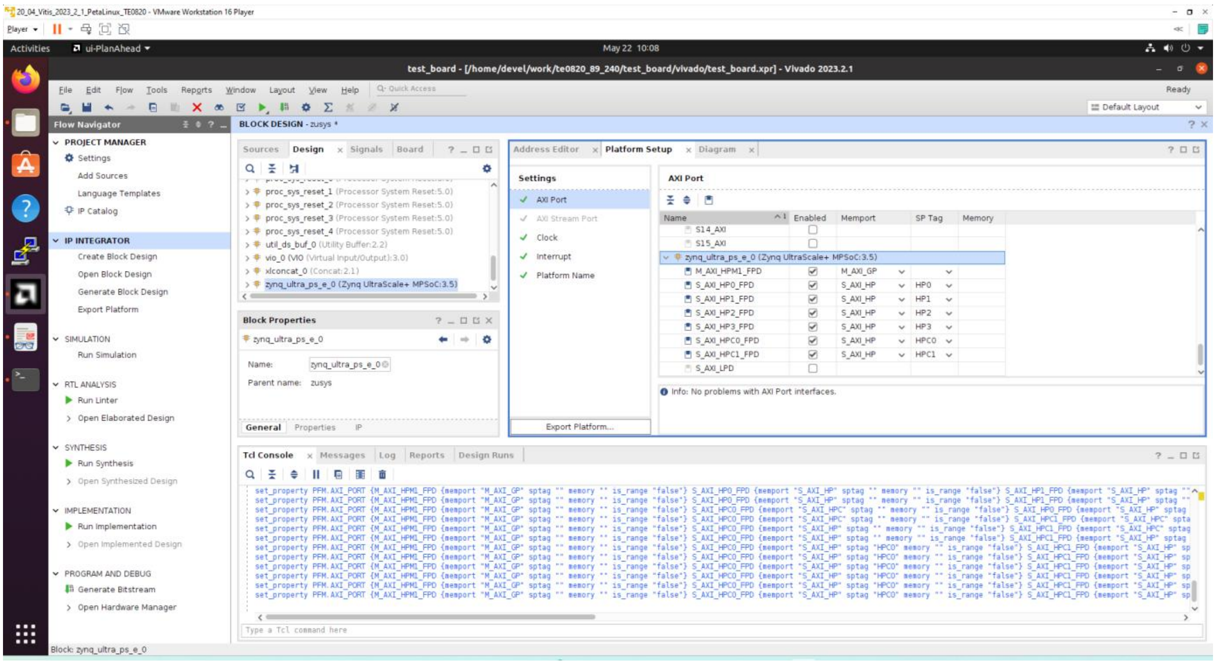
In `Platform Setup`, select AXI Ports for the recently renamed `axi_interconnect_1`:

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.

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



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 3.3 Validate design.

3.2 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

```
te0820_AI_3_0_eval_package\vivado\script_te0820.txt
```

to

```
~/work/te0820_23_240/test_board/vivado/script_te0820.txt
```

Execute in Vivado Tcl console this command:

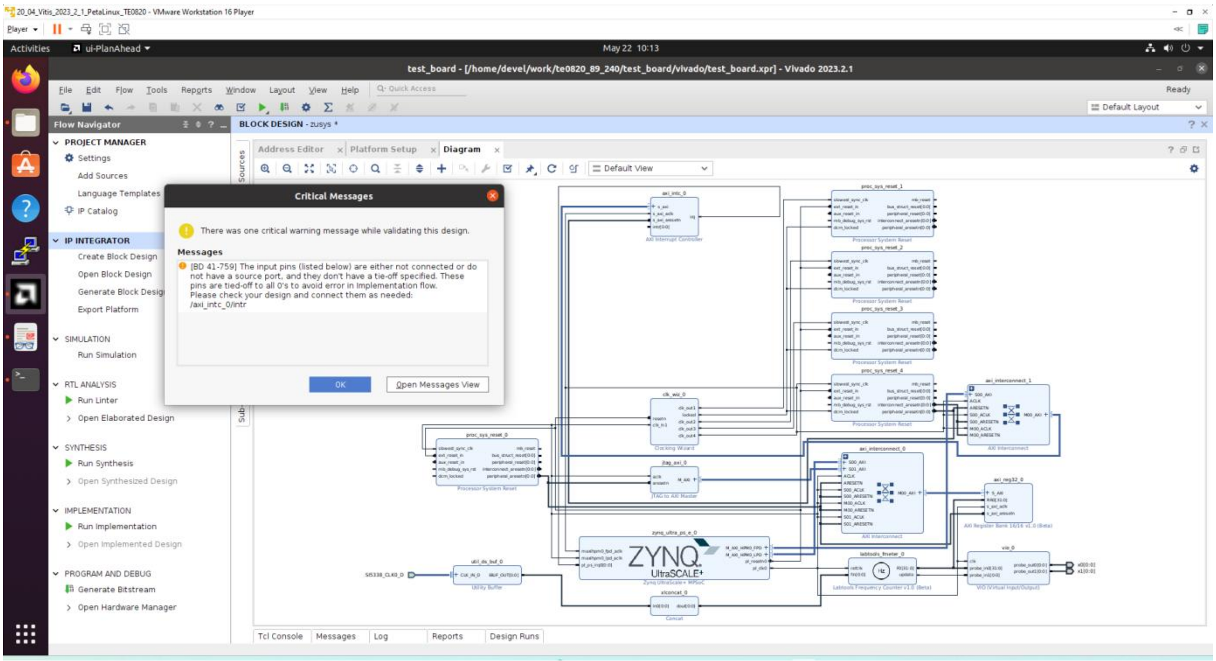
```
source script_te0820.txt
```

3.3 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.

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



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`.

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

```
~/work/te0820_23_240/test_board/vivado/zusys.pdf
```

3.4 Compile Created HW and Custom SW with Trenz Scripts

In Vivado Tcl Console, type the following script and execute it by `Enter`.

```
TE::hw_build_design -export_prebuilt
```

It taks some time to compile HW. HW design and to export the corresponding standard XSA package with included bitstream. Archive `test_board_2cg_1e_2gb.xsa` for extensible system is created:

```
~/work/te0820_23_240/test_board/vivado/test_board_2cg_1e_2gb.xsa
```

In Vivado Tcl Console, type the following script and execute it by `Enter`.

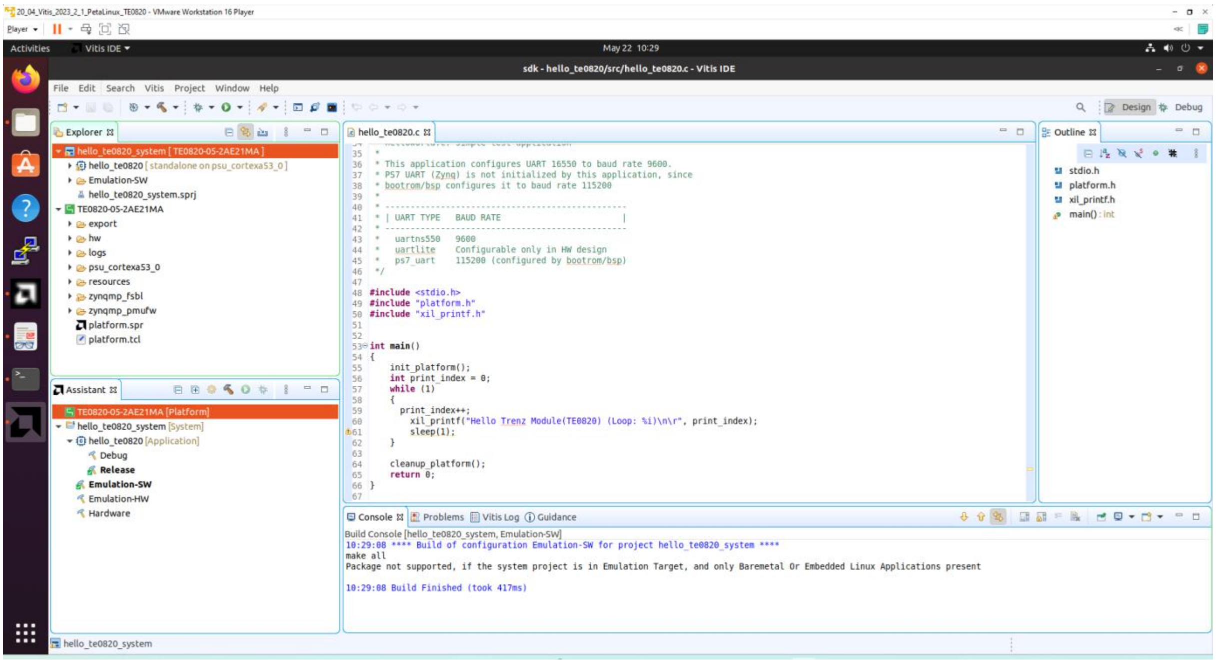
```
TE::sw_run_vitis -all
```

It will take some time to compile and finally the Vitis SDK GUI is opened.

Close the Vitis `Welcome` page.
Compile the two included SW projects.

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

Standalone custom Vitis platform TE0820-05-4DE21MA has been created and compiled.



The TE0820-03-04EV-1EA Vitis platform includes Trenz Electronic custom first stage boot loader in folder zynqmp_fsbl. It includes SW extension specific for the Trenz module initialisation.

This custom zynqmp_fsbl project created an executable file fsbl.elf :

```
~/work/te0820_23_240/test_board/prebuilt/software/2cg_1e_2gb/fsbl.elf
```

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

3.5 Copy Created Custom First Stage Boot Loader

Up to now, test_board directory has been used for all development.

```
~/work/te0820_23_240/test_board
```

Create new folders:

```
~/work/te0820_23_240/test_board_pfm/pfm/boot
~/work/te0820_23_240/test_board_pfm/pfm/sd_dir
```

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

```
~/work/te0820_23_240/test_board/prebuilt/software/2cg_1e_2gb/fsbl.elf
```


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

to

```
~/work/te0820_23_240/test_board_pfm/pfm/boot/fsbl.elf
```

4 Building Petalinux for Vitis AI 3.0 and AI 3.5 SW Library

4.1 Vitis AI3.0 models and Vitis AI 3.5 library

Petalinux 2023.2.1 contains recepies for the Vitis-AI 3.5 library.

The AMD DPU is compatible with Vitis-AI 3.0 models.

4.2 Building Petalinux for Extensible Design Flow

Change directory to the default Petalinux folder

```
~/work/te0820_23_240/test_board/os/petalinux
```

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

```
$ source /tools/Xilinx/Vitis/2023.2/settings64.sh
$ source ~/petalinux/2023.2/settings.sh
```

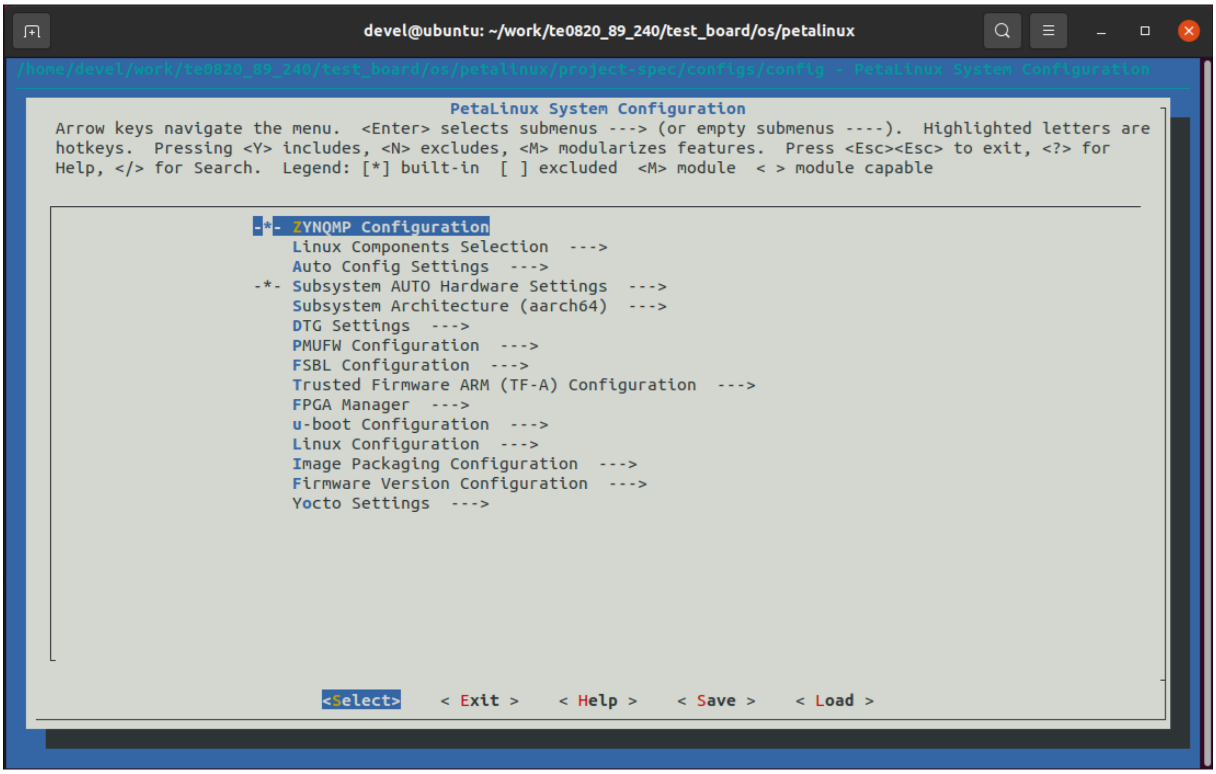
Configure petalinux with the `test_board_2cg_1e_2gb.xsa` for the extensible design flow by executing:

```
$ petalinux-config --get-hw-description=
/home/<user>/work/te0820_84_240/test_board/vivado
```

Replace <user> by your user name.
In our case, `<user>=devel` we use:

```
$ petalinux-config --get-hw-description=
/home/devel/work/te0820_84_240/test_board/vivado
```


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



Select `Exit->Yes` to close this window.

In text editor, modify the `user-rootfsconfig` file:

```
~/work/te0820_23_240/test_board/os/petalinux/project-spec/meta-user/conf/user-rootfsconfig
```

In text editor, append these lines:

```
CONFIG_xrt
CONFIG_xrt-dev
CONFIG_zocl
CONFIG_opencl-clhpp-dev
CONFIG_opencl-headers-dev
CONFIG_packagegroup-petalinux-opencv
CONFIG_packagegroup-petalinux-opencv-dev
CONFIG_dnf
CONFIG_e2fsprogs-resize2fs
CONFIG_parted
CONFIG_resize-part
CONFIG_packagegroup-petalinux-vitisai
CONFIG_packagegroup-petalinux-self-hosted
CONFIG_cmake
CONFIG_packagegroup-petalinux-vitisai-dev
CONFIG_mesa-megadriver
CONFIG_packagegroup-petalinux-x11
```


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

```
CONFIG_packagegroup-petalinux-v4lutils
CONFIG_packagegroup-petalinux-matchbox
CONFIG_packagegroup-petalinux-vitis-acceleration
CONFIG_packagegroup-petalinux-vitis-acceleration-dev
CONFIG_packagegroup-petalinux-vitis-acceleration-essential
CONFIG_packagegroup-petalinux-vitis-acceleration-essential-dbg
CONFIG_packagegroup-petalinux-vitis-acceleration-essential-dev
CONFIG_packagegroup-core-ssh-dropbear
CONFIG_imagefeature-ssh-server-dropbear
CONFIG_imagefeature-ssh-server-openssh
CONFIG_openssh
CONFIG_openssh-sftp-server
CONFIG_openssh-sshd
CONFIG_openssh-scp
CONFIG_imagefeature-package-management
CONFIG_imagefeature-debug-tweaks
```

`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.

The Vitis-AI 3.5 recepies for installation of the correspoding Vitis-AI 3.5 libraries into `rootfs` of PetaLinux are specified allready as selected recepies in PetaLinux 2023.2

Launch `rootfs` config:

```
$ petalinux-config -c rootfs
```

Enable user packages

Select `user packages`

```
user packages   --->

[*] cmake
[*] dnf
[*] e2fsprogs-resize2fs
[*] gpio-demo
[*] imagefeature-debug-tweaks
[*] imagefeature-package-management
[ ] imagefeature-ssh-server-dropbear
[*] imagefeature-ssh-server-openssh
[*] mesa-megadriver
[*] opencl-clhpp-dev
[*] opencl-headers-dev
[*] openssh
[*] openssh-scp
[*] openssh-sftp-server
```


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

```
[*] openssh-sshd
[ ] packagegroup-core-ssh-dropbear
[*] packagegroup-petalinux-matchbox
[*] packagegroup-petalinux-opencv
[*] packagegroup-petalinux-opencv-dev
[*] packagegroup-petalinux-self-hosted
[*] packagegroup-petalinux-v4lutils
[*] packagegroup-petalinux-vitis-acceleration
[*] packagegroup-petalinux-vitis-acceleration-dev
[*] packagegroup-petalinux-vitis-acceleration-essential
[ ] packagegroup-petalinux-vitis-acceleration-essential-dbg
[*] packagegroup-petalinux-vitis-acceleration-essential-dev
[*] packagegroup-petalinux-vitisai
[*] packagegroup-petalinux-vitisai-dev
[*] packagegroup-petalinux-x11
[*] parted
[*] peekpoke
[*] resize-part
[*] startup
[*] webfwu
[*] xrt
[*] xrt-dev
[*] zocl
```

Select all packages to have an asterisk `[*]`.

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

Enable Petalinux package groups

Select `Petalinux Package Groups`:

```
Petalinux Package Groups  --->
  packagegroup-petalinux-vitis-acceleration-essential   --->
      [*] packagegroup-petalinux-vitis-acceleration-essential
      [ ] packagegroup-petalinux-vitis-acceleration-essential-dbg
      [*] packagegroup-petalinux-vitis-acceleration-essential-dev
```

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`.

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

Go to `Filesystem Packages` level by Exit twice.

Go to `console->network->openssh` and enable

```
[*] openssh
[*] openssh-sftp-server
[*] 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
[*] debug_tweaks
```

options. Click OK**,** Exit twice and select Yes to save the changes.

4.3 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`

Select `Exit` and `Yes` **to** Save changes**.**

4.4 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`

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

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

to new value required for the TE0820 module:
`/dev/mmcblk1p2`

Go to

```
Image Packaging Configuration -->
```

modify `Root filesystem formats` from

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

to

```
ext4
```

Select `Exit` and `Yes` to save changes.

4.5 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.

4.6 Build PetaLinux Image

In terminal, build the PetaLinux project by executing:

```
$ petalinux-build
```

The PetaLinux image files will be generated in the directory:

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux
```

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

4.7 Create Petalinux SDK

The SDK will be used by Vitis tool to cross compile applications for newly created platfom.

In terminal, execute:

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

```
$ petalinux-build --sdk
```

The generated sysroot package **sdk.sh** will be located in directory
`~/work/te0820_23_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.

4.8 Copy Files for Extensible Platform

Copy these four files:

Files	From	To
bl31.elf pmufw.elf system.dtb u-boot-dtb.elf	~/work/te0820_23_240/test_board/os/petalinux/images/linux	~/work/te0820_23_240/test_board_pfm/pfm/boot

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

The directory

```
~/work/te0820_23_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	To
boot.scr system.dtb	~/work/te0820_23_240/test_board/os/petalinux/images/linux	~/work/te0820_23_240/test_board_pfm/pfm/sd_dir

Copy file:

File	From	To
init.sh	~/work/te0820_23_240/test_board/misc/sd	~/work/te0820_23_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
```


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

```
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
```

4.9 Create Extensible Platform Archive

Create new directory tree:

```
~/work/te0820_23_240_move/test_board/os/petalinux/images
~/work/te0820_23_240_move/test_board/Vivado
~/work/te0820_23_240_move/test_board_pfm/pfm/boot
~/work/te0820_23_240_move/test_board_pfm/pfm/sd_dir
```

Copy all files from the directory:

Files	Source	Destination
all	~/work/te0820_23_240/test_board/os/petalinux/images	~/work/te0820_23_240_move/test_board/os/petalinux/images
all	~/work/te0820_23_240/test_board_pfm/pfm/boot	~/work/te0820_23_240_move/test_board_pfm/pfm/boot
all	~/work/te0820_23_240/test_board_pfm/pfm/sd_dir	~/work/te0820_23_240_move/test_board_pfm/pfm/sd_dir
test_board_2cg_1e_2gb.xsa	~/work/te0820_23_240/test_board/Vivado/test_board_2cg_1e_2gb.xsa	~/work/te0820_23_240_move/test_board/Vivado/test_board_2cg_1e_2gb.xsa

Zip the directory

```
~/work/te0820_23_240_move
```

into ZIP archive:

```
~/work/te0820_23_240_move.zip
```

The archive `te0820_23_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, and script sdk.sh serving for generation of the sysroot .

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

The archive `te0820_23_240_move.zip` is valid for module number (23). This is the te0820 HW module with xczu4CG-sfvc784-1-e device with 2 GB memory. The Extensible Vitis platform will have the default clock 240 MHz.

4.10 Clean Petainux Files (optional)

The Petalinux compilation process has created round 50 GBytes of files which can be deleated to save Ubuntu disk space.

Change directory to the default Petalinux folder

```
~/work/te0820_23_240/test_board/os/petalinux
```

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

```
$ source /tools/Xilinx/Vitis/2023.2/settings64.sh
$ source ~/petalinux/2023.2/settings.sh
```

Clean Petalinux project files by command

```
petalinux-build -x mrproper
```

This will delete also final filres needed PetaLinux files in the `images` directory.

Restore the final needed PetaLinux files in the `images` directory.

Files	Source	Destination
all	~/work/te0820_23_240_move/test_board/os/petalinux/images	~/work/te0820_23_240/test_board/os/petalinux/images

Copy the `te0820_23_240_move.zip` archive to an disk drive backup.

Delete:

```
~/work/te0820_23_240_move
~/work/te0820_23_240_move.zip
```

Clean Ubuntu Trash.

4.11 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 (23) are present in the archive: `te0820_23_240_move.zip`

In Ubuntu terminal, change the working directory to:

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux
```

In Ubuntu terminal, execute script enabling access to Vitis 2023.2 tools.

```
$ source /tools/Xilinx/Vitis/2023.2/settings64.sh
```


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

In Ubuntu terminal, execute script

```
$ ./sdk.sh -d /home/<user>/work/te0820_23_240/test_board_pfm
```

For `<user>`= `devel`

```
$ ./sdk.sh -d /home/devel/work/te0820_23_240/test_board_pfm
```

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

```
~/work/te0820_23_240/test_board_pfm/sysroots/x86_64-petalinux-linux
~/work/te0820_23_240/test_board_pfm/sysroots/cortexa72-cortexa53-
xilinx-linux
```

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

4.12 Generation of Extensible Platform for Vitis

In Ubuntu terminal, change the working directory to:

```
~/work/te0820_23_240/test_board_pfm
```

Start the `vitis -classic` version of Vitis tool by executing

```
$ vitis -classic &
```

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

```
~/work/te0820_23_240/test_board_pfm
```

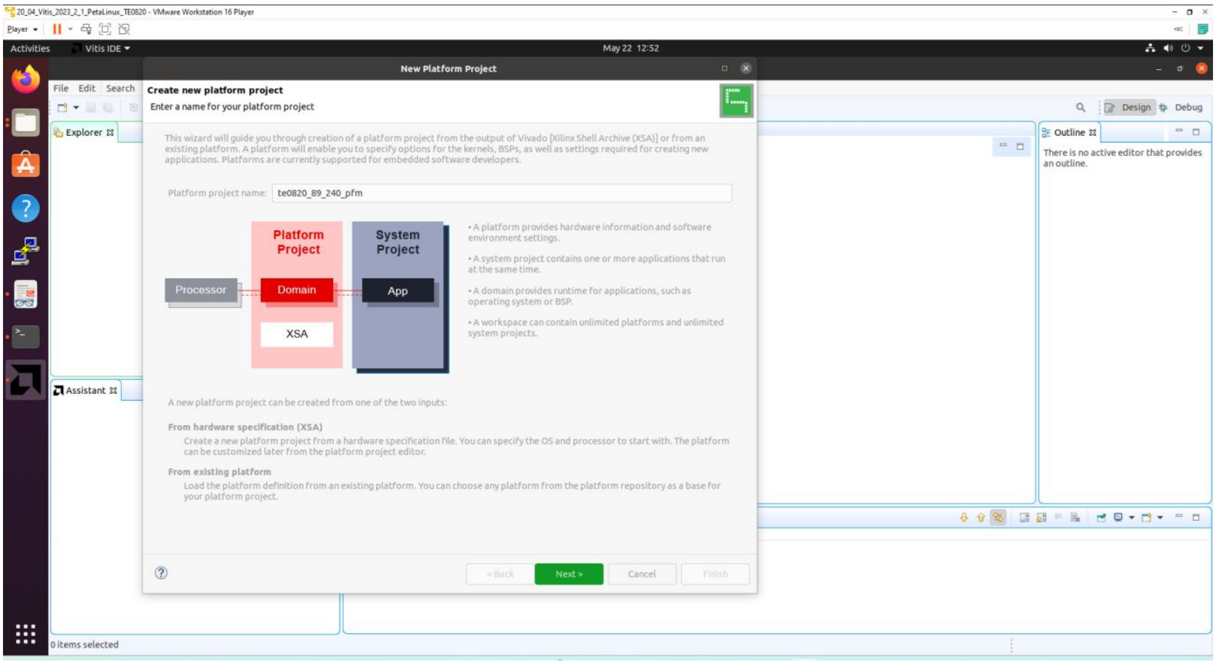
Click on "Launch" to launch `vitis -classic` version of Vitis

Close Welcome page.

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

Type name of the extensible platform: `te0820_23_240_pfm`. Click Next.

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



For hardware specification, select extensible platform archive:

```
~/work/te0820_23_240/test_board/vivado/test_board_4ev_1e_2gb.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)

New window `te0820_23_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/te0820_23_240/test_board_pfm/pfm/boot
```

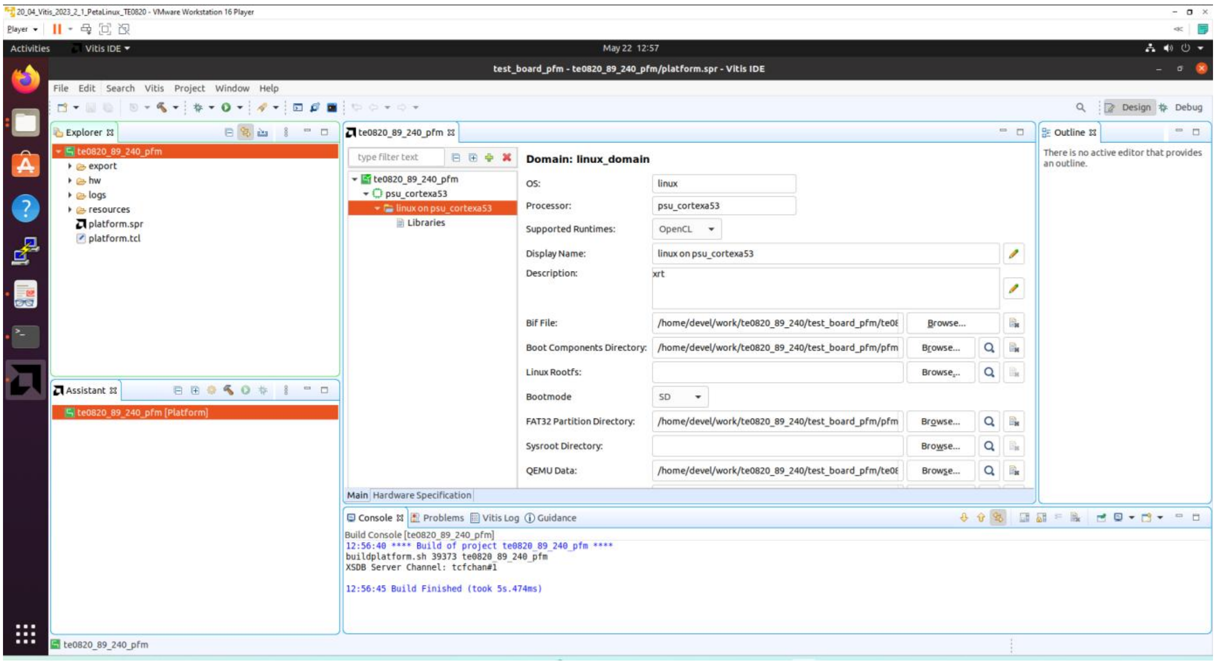
In `FAT32 Partition Directory` select:

```
~/work/te0820_23_240/test_board_pfm/pfm/sd_dir
```

In Vitis IDE `Explorer` section, click on `te0820_23_240_pfm` to highlight it.

Right-click on the highlighted `te0820_23_240_pfm` and select `build project` in the open submenu. Platform is compiled in few seconds.
Close the Vitis tool by selection: `File -> Exit`.

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



Vits extensible platform `te0820_23_240_pfm` has been created in the directory:

```
~/work/te0820_23_240/test_board_pfm/te0820_23_240_pfm/export/te0820_23_240_
pfm
```

5 Platform Usage

5.1 Read Platform Info

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

```
platforminfo
~/work/te0820_23_240/test_board_pfm/te0820_23_240_pfm/export/te0820_23_
240_pfm/te0820_23_240_pfm.xpfm

==========================
Basic Platform Information
==========================
Platform:           te0820_105_240_pfm
File:
/home/devel/work/te0820_105_240/test_board_pfm/te0820_105_240_pfm/expor
t/te0820_105_240_pfm/te0820_105_240_pfm.xpfm
Description:
te0820_105_240_pfm


=====================================
Hardware Platform (Shell) Information
=====================================
Vendor:                           vendor
Board:                            zusys
```


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

```
Name:                              zusys
Version:                           1.0
Generated Version:                 2023.2.1
Hardware:                          1
Software Emulation:                1
Hardware Emulation:                0
Hardware Emulation Platform:       0
FPGA Family:                       zynquplus
FPGA Device:                       xczu4cg
Board Vendor:                      trenz.biz
Board Name:                        trenz.biz:te0820_4cg_1e:2.0
Board Part:                        xczu4cg-sfvc784-1-e


=================
Clock Information
=================
  Default Clock Index: 4
  Clock Index:         1
    Frequency:         100.000000
  Clock Index:         2
    Frequency:         200.000000
  Clock Index:         3
    Frequency:         400.000000
  Clock Index:         4
    Frequency:         240.000000


==================
Memory Information
==================
  Bus SP Tag: HP0
  Bus SP Tag: HP1
  Bus SP Tag: HP2
  Bus SP Tag: HP3
  Bus SP Tag: HPC0
  Bus SP Tag: HPC1


=============================
Software Platform Information
=============================
Number of Runtimes:            1
Default System Configuration:  te0820_105_240_pfm
System Configurations:
  System Config Name:                      te0820_105_240_pfm
  System Config Description:               te0820_105_240_pfm
  System Config Default Processor Group:   linux_domain
  System Config Default Boot Image:        standard
  System Config Is QEMU Supported:         1
```


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

```
  System Config Processor Groups:
    Processor Group Name:      linux on psu_cortexa53
    Processor Group CPU Type:  cortex-a53
    Processor Group OS Name:   linux
  System Config Boot Images:
    Boot Image Name:           standard
    Boot Image Type:
    Boot Image BIF:            te0820_105_240_pfm/boot/linux.bif
    Boot Image Data:           te0820_105_240_pfm/linux_domain/image
    Boot Image Boot Mode:      sd
    Boot Image RootFileSystem:
    Boot Image Mount Path:     /mnt
    Boot Image Read Me:        te0820_105_240_pfm/boot/generic.readme
    Boot Image QEMU Args:
te0820_105_240_pfm/qemu/pmu_args.txt:te0820_105_240_pfm/qemu/qemu_args.
txt
    Boot Image QEMU Boot:
    Boot Image QEMU Dev Tree:
Supported Runtimes:
  Runtime: OpenCL
```

5.2 Create and Compile Vector Addition Example

Create new directory `test_board_test_vadd` to test Vitis extendable flow example “vector addition”

```
~/work/te0820_23_240/test_board_test_vadd
```

Current directory structure:

```
~/work/te0820_23_240/test_board
~/work/te0820_23_240/test_board_pfm
~/work/te0820_23_240/test_board_test_vadd
```

Change working directory:

```
$cd ~/work/te0820_23_240/test_board_test_vadd
```

In Ubuntu terminal, start `vitis –classic` verson of Vitis by:

```
$ vitis --classic –workspace . &
```

In Vitis IDE Launcher, select your working directory

```
~/work/te0820_23_240/test_board_test_vadd
```

Click on `Launch` to launch Vitis.

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

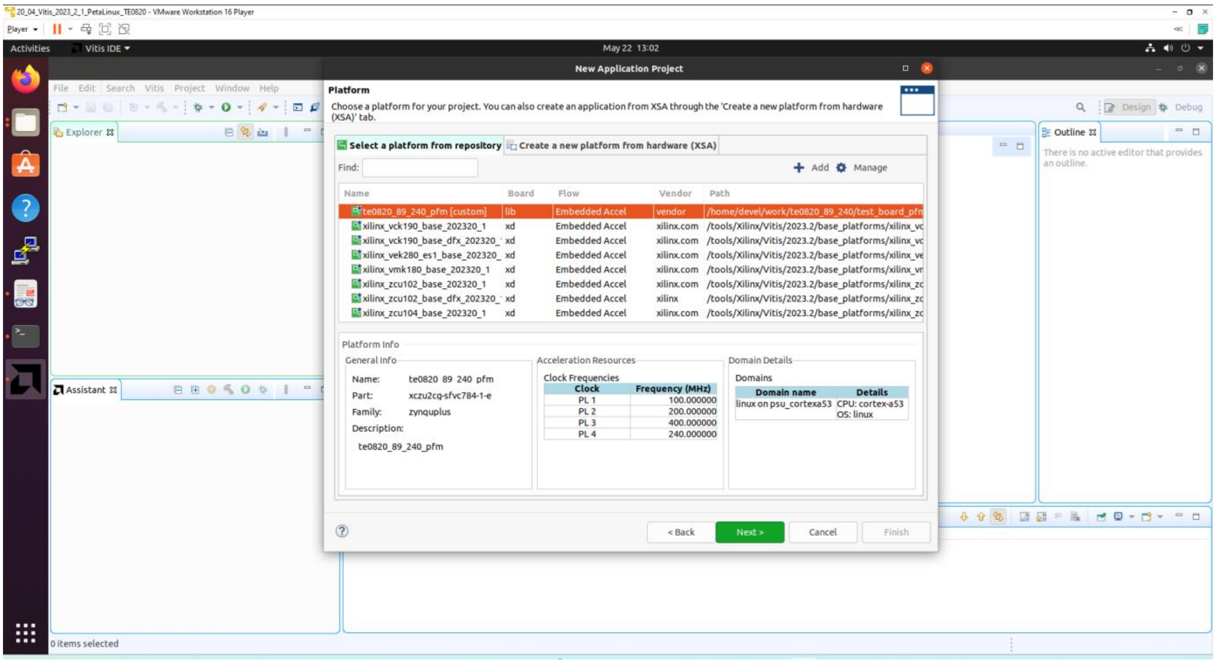
Skip welcome page if shown.

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

Click on `[+ Add]` icon and select the custom extensible platform `te0820_23_240_pfm[custom]` in the directory:

```
~/work/te0820_23_240/test_board_pfm/te0820_23_240_pfm/export/te0820_23_240_pfm
```

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



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/te0820_23_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-linux
```

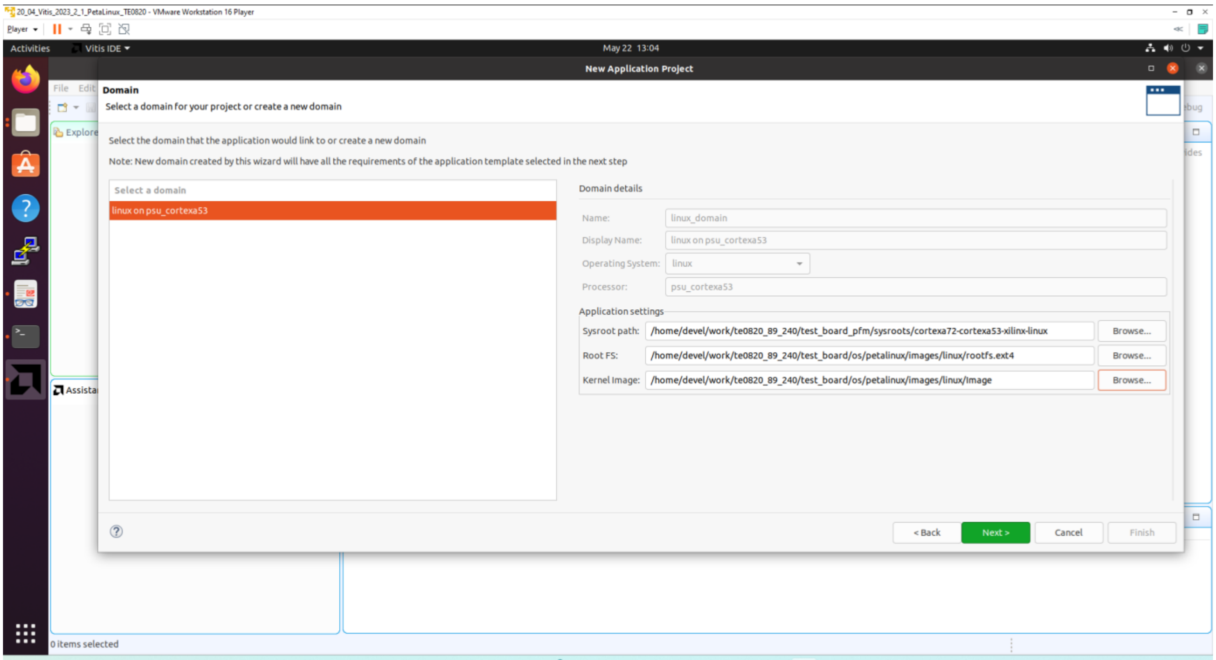
Root FS:

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux/rootfs.ext4
```

Kernel Image:

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux/Image
```


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



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.

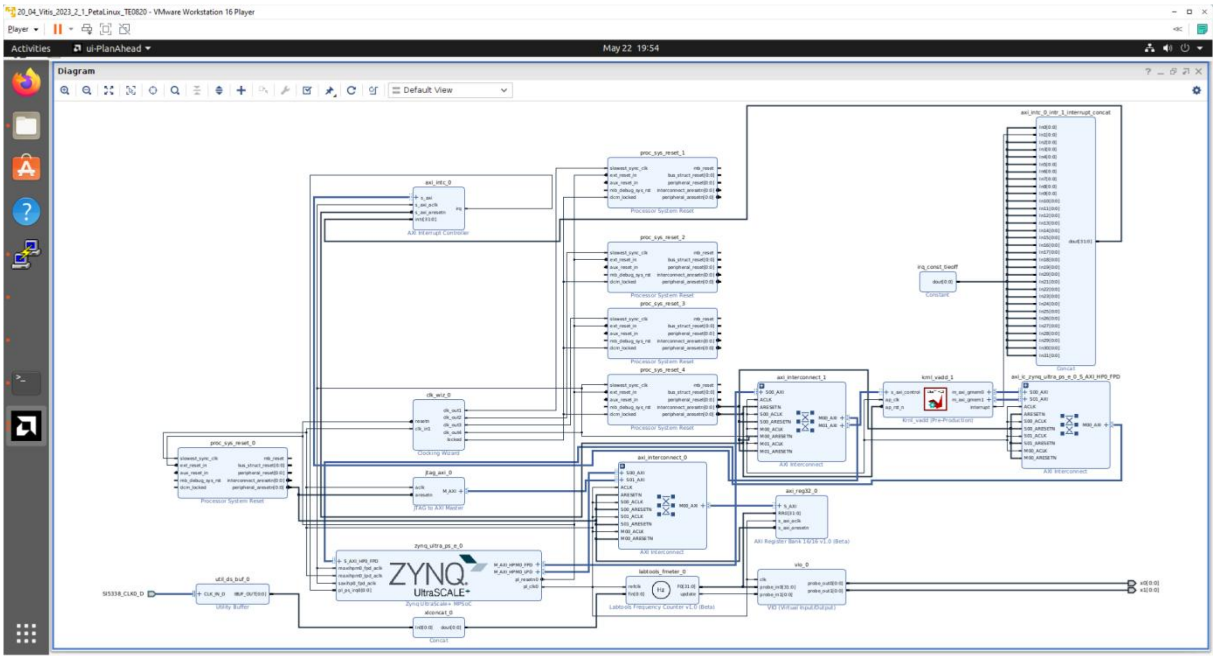
In test_vadd window menu "Active build configuration" switch from `SW Emulation` to `Hardware`.

In "Explorer" section of Vitis IDE, click on: `test_vadd_system[te0820_23_240_pfm]` to select it.

Right Click on: `test_vadd_system[te0820_23_240_pfm]` and select in the opened sub-menu: `Build project`

Vitis will compile. This step can take some time.

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



Created extended HW with integrated vadd IP block can be open and analysed in Vivado 2023.2.

5.3 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/te0820_23_240/test_board_test_vadd/test_vadd_system/Hardware/pac
kage/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.
https://win32diskimager.org/

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.

In PC, find the assigned serial line COM port number for the USB terminal. In case of Win 10 or Win 11, 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:

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

```
sh-5.1# cd /media/sd-mmcblk1p1/

sh-5.1# ./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-5.1#
```

The Vitis application has been compiled to HW and evaluated on custom system with extensible custom `te0820_23_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.
The terminal can be closed.
TE0701-06 carrier board can be disconnected from power.

5.4 Display on X11 terminal

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`)

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

```
sh-5.1# cd /media/sd-mmcblk1p1/
sh-5.1# ./test_vadd krnl_vadd.xclbin
```

The application test_vadd should run with this output:

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

```
INFO: Reading krnl_vadd.xclbin
Loading: 'krnl_vadd.xclbin'
Trying to program device[0]: edge
Device[0]: program successful!
TEST PASSED
sh-5.1#
```

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-5.1# 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.
Close the PC USB terminal application.
The TE0701-06 carrier board can be disconnected from power.

6 Vitis AI 3.0 DPUCZDX8G_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-AI/tree/3.0/dpu

Page description contains table with supported targets. Use the line if theis table dedicated to the AMD DPUCZDX8G 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 Zynq® UltraScale+™ platforms.

Click on the Download link in the column: Reference Design

This will result in download of file:

```
~/Downloads/DPUCZDX8G_VAI_v3.0.tar.gz
```

It contains directory

```
~/Downloads/DPUCZDX8G_VAI_v3.0
```

Copy this directory to the directory:

```
~/work/DPUCZDX8G_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.

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

We have to make these template project files visible for the Vitis tool. Copy the directory:

```
~/work/DPUCZDX8G_VAI_v3.0
```

to the directory

```
/home/<user>/.Xilinx/Vitis/2023.2/vitis_examples/DPUCZDX8G_VAI_v3.0
```

In case of `<user>=devel`

```
/home/devel/.Xilinx/Vitis/2023.2/vitis_examples/DPUCZDX8G_VAI_v3.0
```

Vitis will see this platform template, now.

6.1 Create and Build Vitis Design

Create new directory `test_board_dpu_trd` to test Vitis extendable flow example `dpu trd`

```
~/work/te0820_23_240/test_board_dpu_trd
```

Current directory structure:

```
~/work/te0820_23_240/test_board
~/work/te0820_23_240/test_board_pfm
~/work/te0820_23_240/test_board_test_vadd
~/work/te0820_23_240/test_board_dpu_trd
```

Change working directory:

```
$cd ~/work/te0820_23_240/test_board_dpu_trd
```

In Ubuntu terminal, start vitis –classic version of Vitis tool by:

```
$ vitis --classic --workspace . &
```

In Vitis IDE Launcher, select your working directory

```
~/work/te0820_23_240/test_board_dpu_trd
```

Click on `Launch` to start `Vitis`.

6.2 Configure Project for the Vitis Extensible Flow with DPU

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

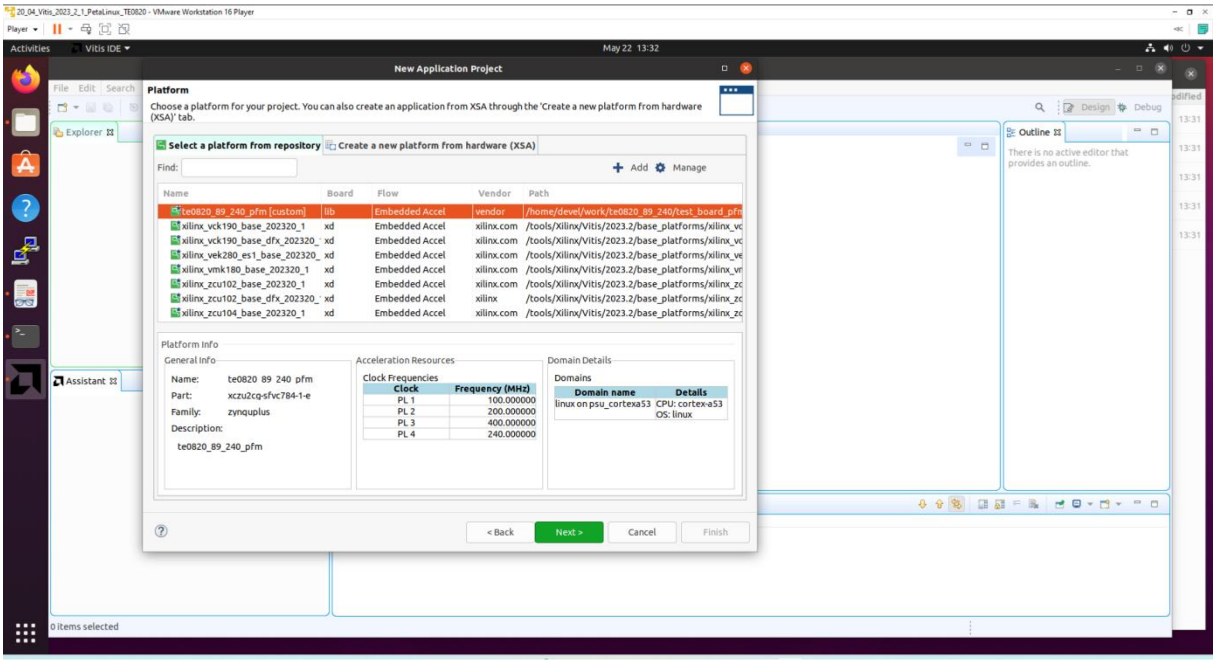
Skip welcome page, if it is shown.

Click on `[+ Add]` icon and select the custom extensible platform `te0820_23_240_pfm[custom]` in the directory:

```
~/work/te0820_23_240/test_board_pfm/te0820_23_240_pfm/export/te0820_23_240_
pfm
```


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

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



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/te0820_23_240/test_board_pfm/sysroots/cortexa72-cortexa53-xilinx-linux
```

"Root FS":

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux/rootfs.ext4
```

"Kernel Image":

```
~/work/te0820_23_240/test_board/os/petalinux/images/linux/Image
```

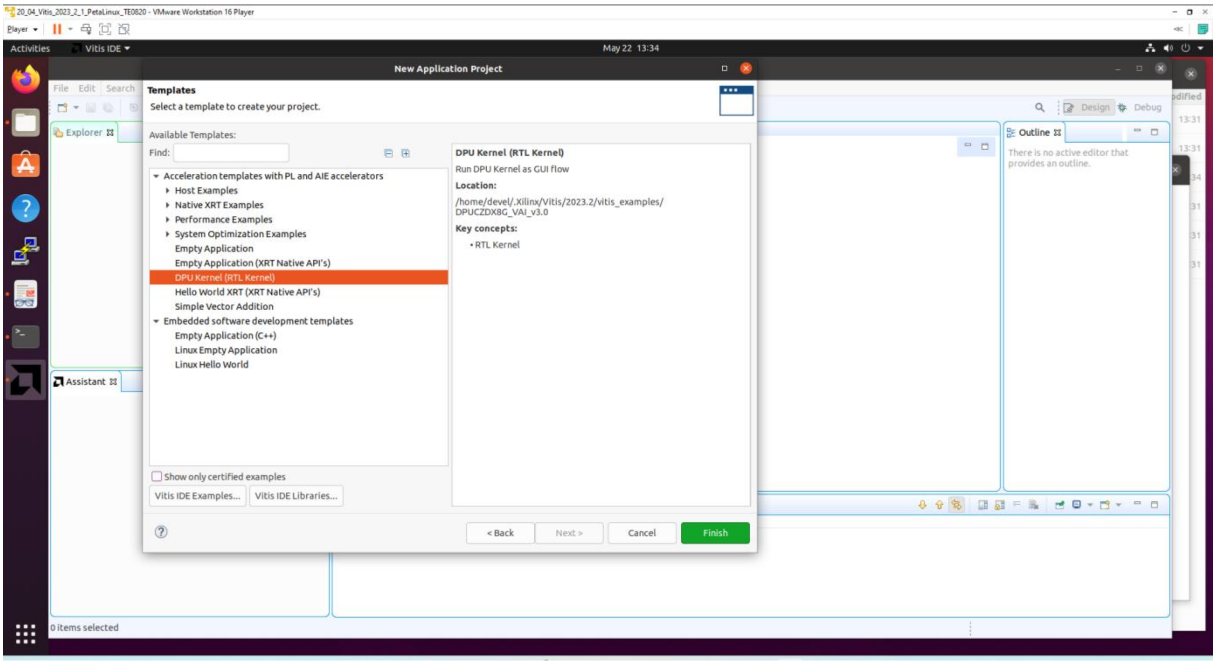
Click Next.

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

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

Select: `DPU Kernel (RTL Kernel)`

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



Click Finish
New project template is created.

In dpu_trd window menu `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.

In case of module with ID=23: TE0820-03-04EV-1EA, device xczu2cg-sfvc784-1-e keep in file `dpu_conf.vh` DPU size B4096 .

To fit the AMD DPUCZDX8G in configuration B4096, the use of URAM blocks must be enabled. URAM blocks are present only in the `xczu04` devices. Modify:

```
`define URAM_DISABLE
```

To

```
`define URAM_ENABLE
```

Go to `dpu_trd_system_hw_link` and double click on `dpu_trd_system_hw_link.prj`

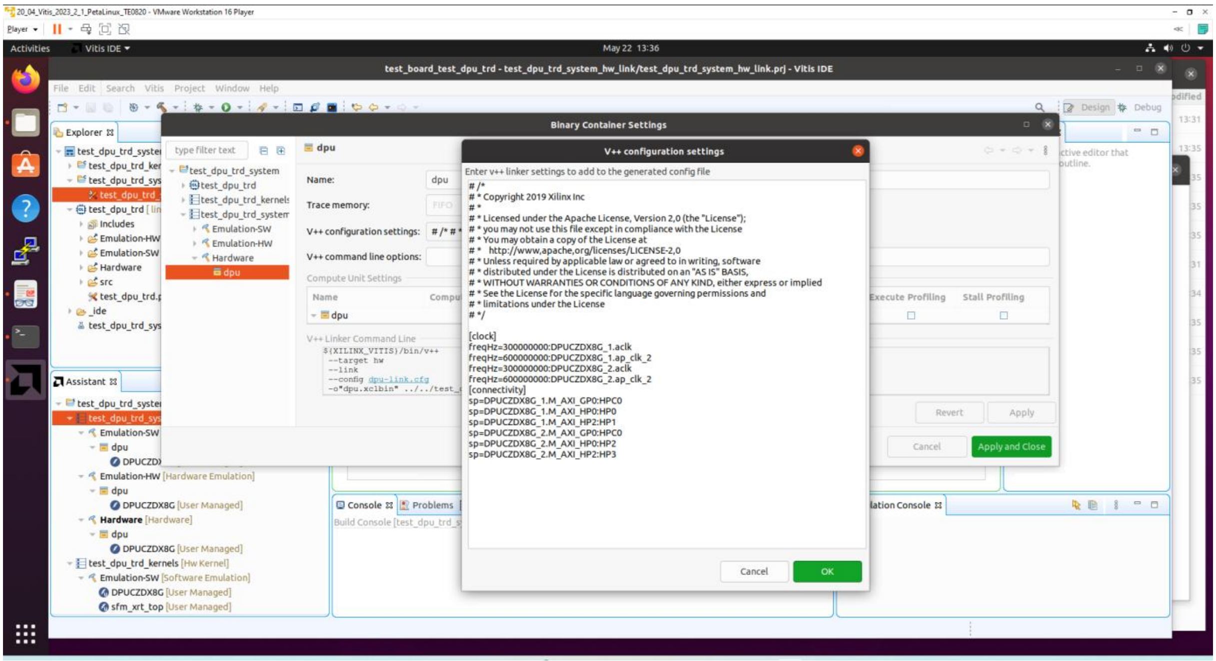
Remove `sfm_xrt_top` kernel from binary container by right clicking on it and choosing remove.

Reduce number of DPU kernels to one.

6.3 Configure Connection of DPU kernel

On the same tab right click on `dpu` and choose `Edit V++ Options`

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



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

```
[clock]
freqHz=200000000:DPUCZDX8G_1.aclk
freqHz=400000000: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
```

To accelerate compilation, you can add to V++ command line options these commands for Vitis:

```
--hls.jobs 12 --vivado.synth.jobs 12 --vivado.impl.jobs 12
```

This sets the maximal number of threads for HLS, Vivado Synthesis, and Vivado Implementation to 12 (if you have 12 cores reserved for Ubuntu OS in ypur virtual machine).

6.4 Build the test_dpu_trd Project

In “`Explorer`” section of Vitis IDE, click on:

```
dpu_trd_system[te0820_23_240_pfm]
```

to select it.

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

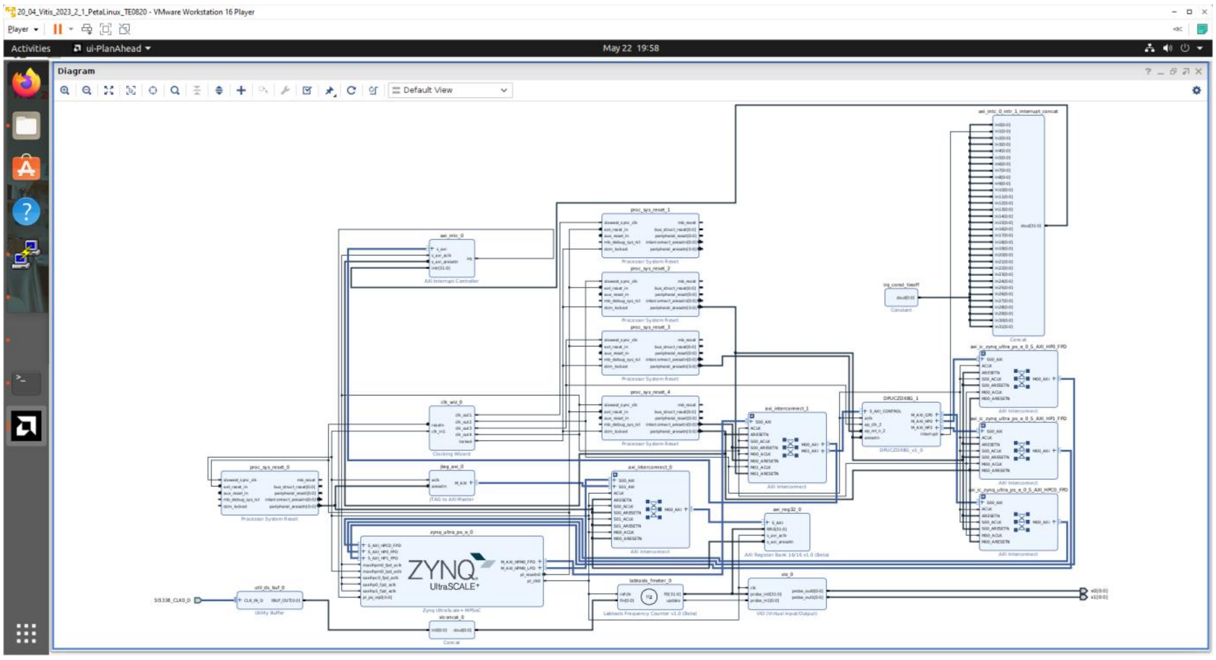
Right Click on:

```
dpu_trd_system[te0820_23_240_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 2023.2



7 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/te0820_23_240/test_board_dpu_trd/dpu_trd_system/Hardware/package
```

In Windows 10 (or Windows 11) PC, install program `Win32DiskImager` for this task. Win32 Disk Imager can write raw disk image to removable devices.
https://win32diskimager.org/

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 Run Compiled Example Application for Vector Addition.

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

7.1 Resize EXT4 Partition

Check ext4 partition size by:

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

Resize partition

```
root@Trenz:~# resize-part /dev/mmcblk1p2
```

Check ext4 partition size again, you should see:

```
root@Trenz:~# df . -h
Filesystem                Size      Used  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 path to `dpu.xclbin` :

```
sh-5.1# root@petalinux:~# export
XLNX_VART_FIRMWARE=/run/media/mmcblk1p1/dpu.xclbin
```

7.2 Test the Integrated DPUCZDX8G

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

```
root@Trenz:~# xdputil query
```

Command and reply in case of module with ID=23 (DPU configuration B4096):

```
root@Trenz:~# xdputil query
{
    "DPU IP Spec":{
        "DPU Core Count":1,
        "IP version":"v4.1.0",
        "generation timestamp":"2023-02-21 21-30-00",
        "git commit id":"7d32c41",
        "git commit time":2023022121,
        "regmap":"1to1 version"
    },
    "VAI Version":{
        "libvart-runner.so":"Xilinx vart-runner Version: 3.5.0-
b7953a2a9f60e23efdfced5c186328dd1449665c  2024-04-18-10:03:55 ",
        "libvitis_ai_library-dpu_task.so":"Advanced Micro Devices
vitis_ai_library dpu_task Version: 3.5.0-
```


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

```
b7953a2a9f60e23efdfced5c186328dd1449665c  2023-06-29 03:20:28 [UTC] ",
        "libxir.so":"Xilinx xir Version: xir-
b7953a2a9f60e23efdfced5c186328dd1449665c 2024-04-18-09:27:26",
        "target_factory":"target-factory.3.5.0
b7953a2a9f60e23efdfced5c186328dd1449665c"
    },
    "kernels":[
        {
           "AIE Frequency (Hz)":0,
            "DPU Arch":"DPUCZDX8G_ISA1_B4096",
            "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":"0xaaab129ceb30",
            "cu_idx":0,
            "cu_mask":1,
            "cu_name":"DPUCZDX8G:DPUCZDX8G_1",
            "device_id":0,
            "fingerprint":"0x101000056010407",
            "name":"DPU Core 0"

        }
    ]
}
}
root@Trenz:~#
```

7.3 Test resnet50_pt model

The AMD DPUCZDX8G in configuration B4096 can be tested with AI 3.0 resnet50_pt model precompiled for the B4096 configuration.

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

Copy the precompiled AI 3.0 resnet50_pt model files accompanying this application note:

From:

```
B4096\app\model\md5sum.txt
B4096\app\model\resnet50_pt.prototxt
B4096\app\model\resnet50_pt.xmodel
```

to target system directory

```
~/app/model/
```

Change the directory to `~/app/model/`

```
cd ~/app/model/
```

Note: If you integrate DPU with B0512, B1024 or B1600 configuration, use files from the evaluation package prepared for that configuration.

Test resnet50_pt model by command

```
sh-5.1# xdputil benchmark resnet50_pt.xmodel 1
```

Result of test

```
root@Trenz:~/app/model# xdputil benchmark resnet50_pt.xmodel 1
WARNING: Logging before InitGoogleLogging() is written to STDERR
I20241205 09:51:12.122829  1198 test_dpu_runner_mt.cpp:477] shuffle
results for batch...
I20241205 09:51:12.124051  1198 performance_test.hpp:73] 0% ...
I20241205 09:51:18.124511  1198 performance_test.hpp:76] 10% ...
I20241205 09:51:24.124938  1198 performance_test.hpp:76] 20% ...
I20241205 09:51:30.125094  1198 performance_test.hpp:76] 30% ...
I20241205 09:51:36.125262  1198 performance_test.hpp:76] 40% ...
I20241205 09:51:42.125427  1198 performance_test.hpp:76] 50% ...
I20241205 09:51:48.125581  1198 performance_test.hpp:76] 60% ...
I20241205 09:51:54.126062  1198 performance_test.hpp:76] 70% ...
I20241205 09:52:00.126227  1198 performance_test.hpp:76] 80% ...
I20241205 09:52:06.126374  1198 performance_test.hpp:76] 90% ...
I20241205 09:52:12.126545  1198 performance_test.hpp:76] 100% ...
I20241205 09:52:12.126606  1198 performance_test.hpp:79] stop and
waiting for all threads terminated....
I20241205 09:52:12.138233  1198 performance_test.hpp:85] thread-0
processes 3554 frames
I20241205 09:52:12.138262  1198 performance_test.hpp:93] it takes 11226
us for shutdown
I20241205 09:52:12.138506  1198 performance_test.hpp:94] FPS= 59.2193
number_of_frames= 3554 time= 60.0142 seconds.
I20241205 09:52:12.138561  1198 performance_test.hpp:96] BYEBYE
Test PASS.
root@Trenz:~/app/model#
```

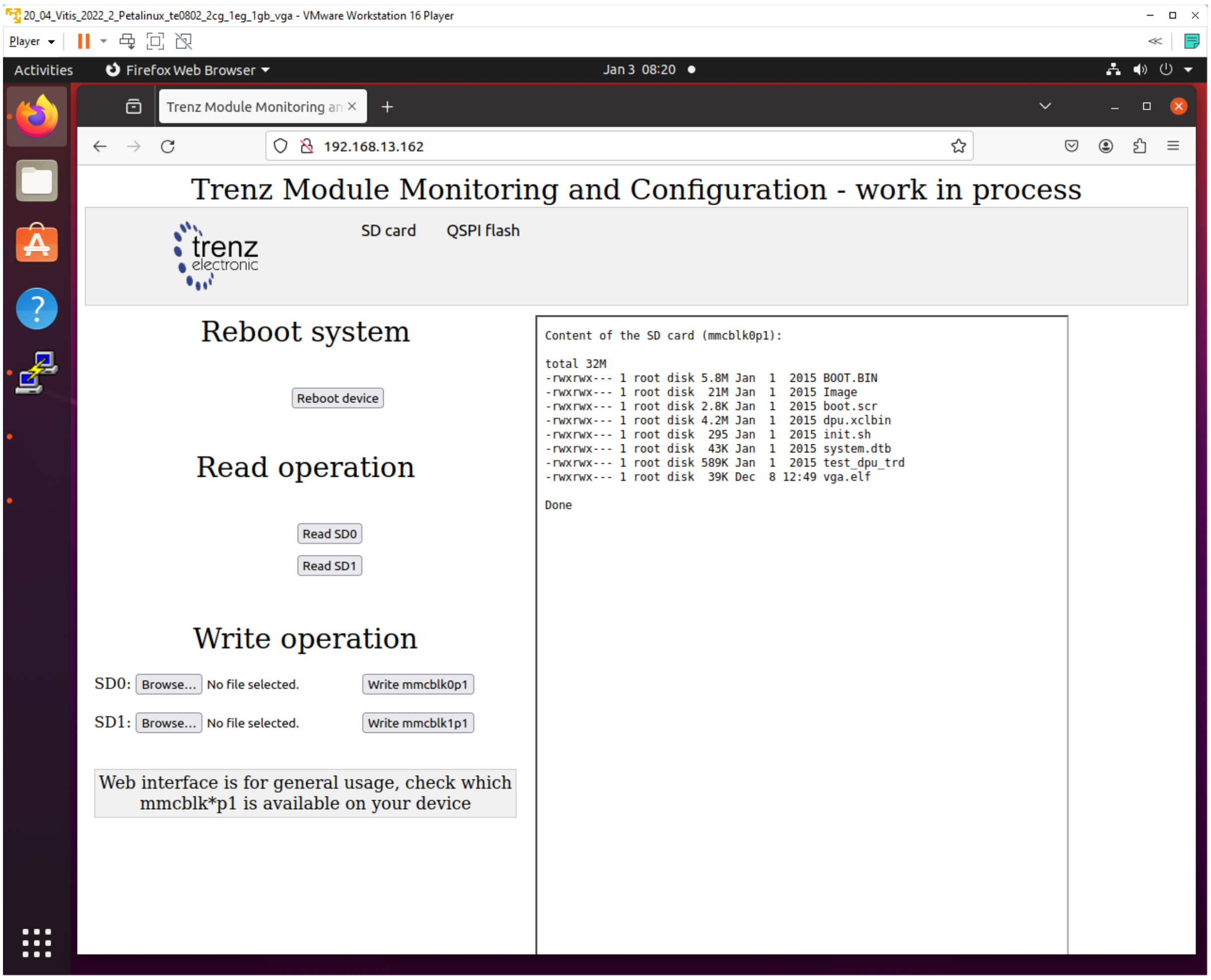

© 2024 ÚTIA AV ČR, v.v.i.
All disclosure and/or reproduction rights reserved

Benchmark indicates **59.2 FPS** for inference of `resnet50_pt` model.
Power consumption of the system during benchmark is increased to **13.0 W**.
Compilation of AI 3.0 models for the AMD DPUCZDX8G is described in separate application note and evaluation package [10]:
Compilation of AI 3.0 models for Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8G.

7.4 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 `mmcblk1p1` .

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.

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

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

7.5 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:

```
root@Trenz:~# rxvt &
root@Trenz:~# rxvt &
```

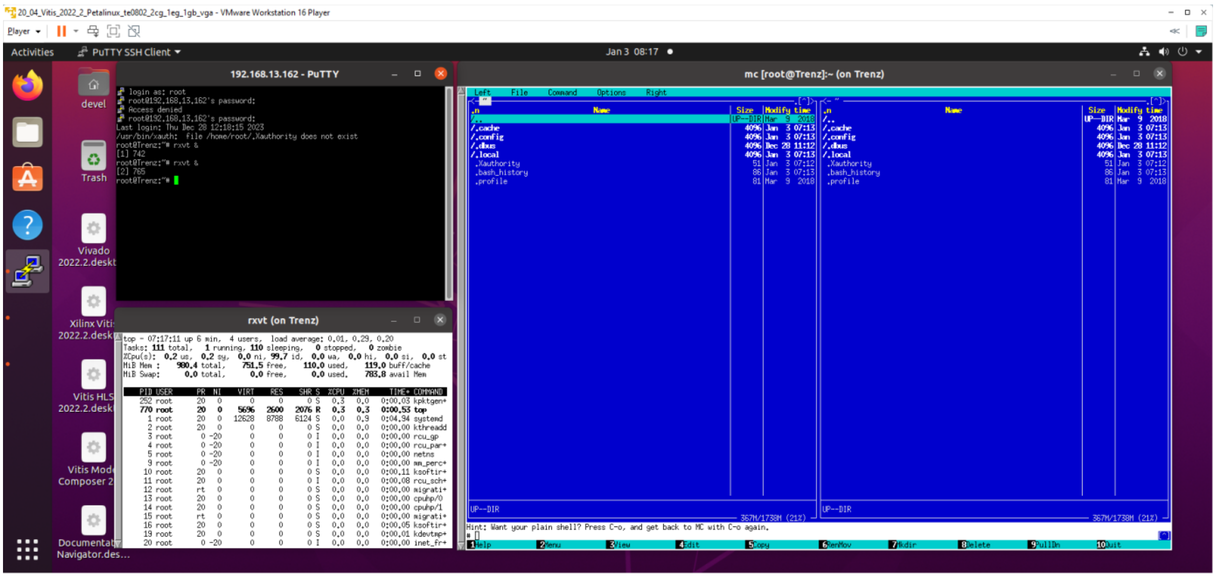
In first rxvt terminal emulator window start utility

```
root@Trenz:~# top
```

In second rxvt terminal emulator start

```
root@Trenz:~# 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.

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

7.6 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:

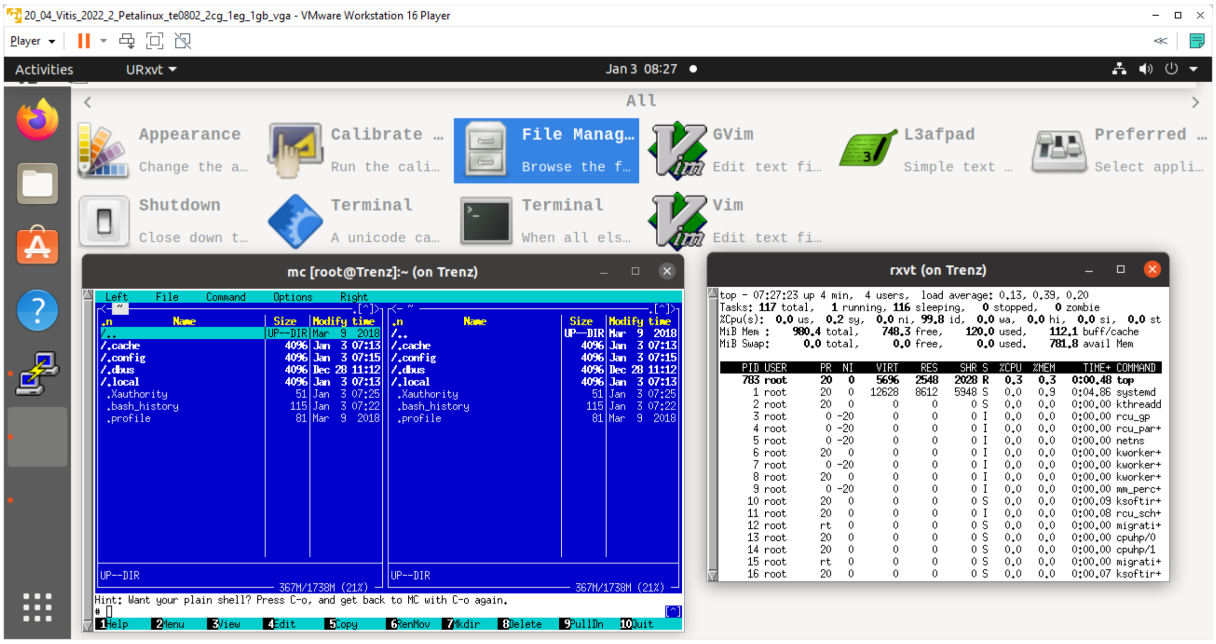
```
root@Trenz:~# x-session-manager &
```

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:

```
sh-5.1# rxvt &
sh-5.1# 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.

7.7 Display Test Pattern and Test USB Camera

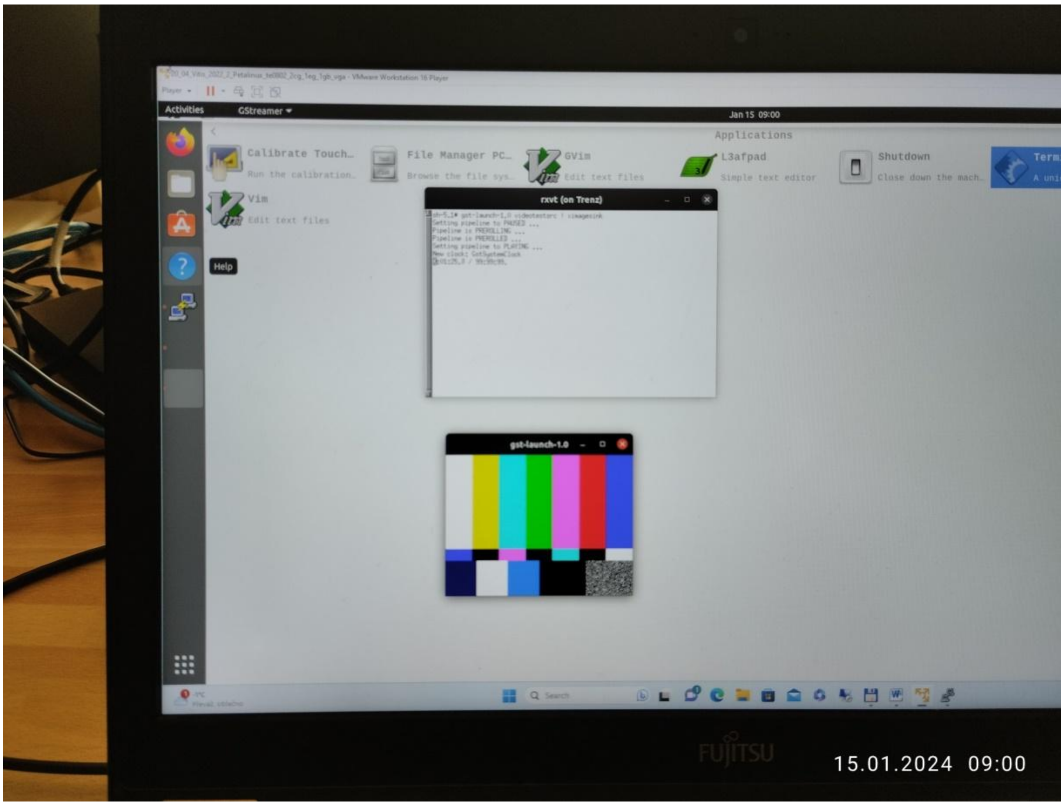
Complete video chain can be tested with output to the X11 desktop.

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

To display the test pattern, use this gstreamer command:

```
sh-5.1# gst-launch-1.0 videotestsrc ! ximagesink
```

Video output is directed to the remote X11desktop.



Test pattern is displayed on remote PC X11 desktop

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

7.8 TE0820-03-04EV-1EA Module ID=23, TE0701-06, DPU (B4096)

Vitis AI 3.0 exampes	Performance input from camera e2e (-t 1) [FPS]	Power w input from camera (-t 1) [W]	Performance input from file e2e (-t 3) [FPS]	Power with input from file e2e (-t 3) [W]	GigaOps input from file e2e (-t 3) [Gops]
Face detection Model: pt_face-mask-detection_512_512_0.67G_3.0	30.0	10.0	113	9.8	75.7
Vehicle make Model: pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
Vehicle type Model: pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
Vehicle color Model: pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
General classification Model: pt_resnet50_imagenet_224_224_8.2G_3.0	30.0	11.9	59.6	13.0	488.7
General classification Model: pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	30.0	11.5	69.2	12.7	401.3
General classification Model: pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	30.0	11.2	73.8	12.4	361.6
General classification Model: pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	30.0	11.0	81.1	12.2	332.5
General classification Model: pt_resnet50_imagenet_224_224_0.6_3.3G_3.0	30.0	10.7	91.1	11.9	300.6
General classification Model: pt_resnet50_imagenet_224_224_0.7_2.5G_3.0	30.0	10.6	99.6	11.5	249.0

Measurement conditions:

- TE0820-03-04EV-1EA module (4EV-1E device, 2GB DDR4), with 12V FAN on TE0701-06 carrier board
- DPU in B4096 configuration
- USB WWW colour camera logi 720p, Logitech, 1280x720p30, 30 FPS
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug

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

8 References

[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/STM32CubeH5/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

[7]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for STM32H573I-DK V1.4.0 web server. (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=21_STM32H753_DK_V1_4_0
This application and evaluation package is based on the STM32CubeH5 ver 1.4 Firmware Examples for STM32H5xx Series Application based on NetXDuo: Nx_WebServer. https://www.st.com/en/development-tools/stm32cubeide.html

[8] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for TE0821 Modules in Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8G (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=24_TE0821_AI_3_5

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

[9]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Support for TE0820 Modules in Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8V (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=25_TE0820_AI_3_5
[10]
Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Compilation of AI 3.0 models for Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8V. (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=26_TE_AI_3_5

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