

# Vitis Platform Creation and Integrating with DPU

In this document we are going to have steps on:

- Creating Vitis Platform
- Creating DPU overlay on top of Vitis Platform

## A. Using the given Vitis Platform and going for DPU overlay?

1. Download "DPUCZDX8G\_VAI\_v3.0" from here([DPUCZDX8G\\_VAI\\_v3.0.tar.gz](https://www.xilinx.com/member/forms/download/xef.html?filename=xilinx-dpuczd8g-vai-v3.0.tar.gz)) and extract the tar file in this directory: **Kria-Prophesee-Event-VitisAI/Vivado-Vitis/Vitis-Platform-Overlay** directory

- At here </Kria-Prophesee-Event-VitisAI/Vivado-Vitis/Vitis-Platform-Overlay/DPU-Overlay> we have provided the "**dpu\_conf.vh**" and make file.
- Now you have to update the "**dpu\_conf.vh**" inside "DPUCZDX8G\_VAI\_v3.0/prj/Vitis". This dpu\_conf has URAM enabled configuration for consuming the available URAM on Kria K26 FPGA of KV260.
- Also replace the "makefile" at your directory "DPUCZDX8G\_VAI\_v3.0/prj/Vitis", this makefile has updated the "single DPU configuration". As KV260 only support single DPU of 4096 we are doing this. and run the make script.

For reference: follow the build log attached here( **Kria-Prophesee-Vitis-DPU-Design-log-j10.sh** ) with or follow the readme.md from "DPUCZDX8G\_VAI\_v3.0.tar.gz/DPUCZDX8G\_VAI\_v3.0/prj/Vitis".

2. Now , in the terminal, you have to do:

### i. Source XRT and Vitis:

Source XRT: source /opt/xilinx/xrt/setup.sh and

Source Vitis : source ~/tools/Xilinx/2022\_2/Vivado/2022.2/settings64.sh .

If you dont have XRT then you can setup via github link: or can download the Ubuntu version specific one from here:

- ii. Download the "zynqmp-common" from petalinux page for 2022.2:  
[https://www.xilinx.com/member/forms/download/xef.html?filename=xilinx-zynqmp-common-v2022.2\\_10141622.tar.gz](https://www.xilinx.com/member/forms/download/xef.html?filename=xilinx-zynqmp-common-v2022.2_10141622.tar.gz)

and do export its directory: export  
EDGE\_COMMON\_SW=/home/logictronix03/Downloads/kk/June9-2024-kv260-platform/kv260\_vitis\_platform/xilinx-zynqmp-common-v2022.2

- iii. Do export the vitis platform directory:

```
export SDX_PLATFORM=/home/logictronix03/Downloads/kk/June9-2024-kv260-  
platform/kv260-ppse-hardware-platform/kv260-pppse-dpu-vitis-flow-  
J10/kv260_ppse_custom/export/kv260_ppse_custom/kv260_ppse_custom.xpfm
```

- iv. Then run: **make all KERNEL=DPU DEVICE=KV260**

Based on the PC config (CPU/RAM) this build may take from 15min to an hour or so.

At the end of build completion, you will get SD\_card not a directory like error , you can skip that. cp ./binary\_/link/vivado/vpl/prj/prj/sources\_1/bd/hw\_handoff/.hwh  
./binary\_/sd\_card cp: target './binary\_/sd\_card' is not a directory make: \*\*\*  
[Makefile:112: package] Error 1

Above error occurred cause we are not going to create sd\_card image from this make process. As KV260 follows "FPGA manger or DFX Manager or XMUTI utility" to load the "firmware(BIN, XCLBIN, JSON, DTBO) in Kria-Ubuntu we dont need "SD\_Boot Image".

## **B. Creating the Vitis Platform**

For creating Vitis Platform , you have to first create the Vivado hardware platform as mentioned here: [Vivado-hardware-platform](#)

For this hardware creation step you can follow this document: [https://github.com/Xilinx/Vitis-Tutorials/blob/2022.2/Vitis\\_Platform\\_Creation/Design\\_Tutorials/01-Edge-KV260/README.md](https://github.com/Xilinx/Vitis-Tutorials/blob/2022.2/Vitis_Platform_Creation/Design_Tutorials/01-Edge-KV260/README.md)

For creating “step 1- Vivado Hardware Design and Generate XSA” you have to have “Prophesee MIPI IP based VIVADO block design” you can get the Prophesee IP by requesting them via email. They have released IP cores for “MIPI event camera pipeline for VIVADO” in Apache2 licensing format, so easy access can be get.

If you got the Prophesee MIPI pipeline or IP core for “CCAM5” then you can also run this Tcl script in your VIVADO 2022.2 terminal : [kv260\\_top\\_hardware\\_plaform\\_with\\_mipi.tcl](#)

You can also download the readily usable “Kria Prophesee MIPI” based “Vitis Platform” from here: [Vivado-Vitis/Vitis-Platform-Overlay/Vitis-Platform](#)