AMD Tech Day

AI#3 - Vitis AI DPU: Explore a Vivado based flow targeting the Kria KV260

Dimitrios Kolosov, Senior Field Application Engineer - FPGA Specialist

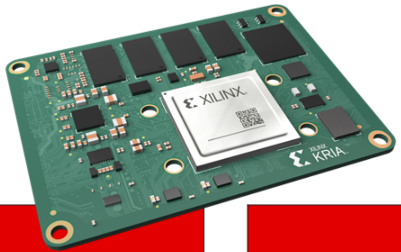
8th November, 2023

Agenda

- Overviews
- AMD Vivado Steps
- AMD PetaLinux Steps
- KV260 Deployment
- Q&A

Overviews

K26 SOM Overview
Based on the Zynq® UltraScale+™ MPSoC Architecture



COMPUTE	
Application Processor	64-bit Quad-Core Arm® Cortex®-A53
Real-Time Processor	32-bit Dual-Core Arm Cortex-R5F
Graphics Processor	Arm Mali™-400MP2
Programmable Logic	256K System Logic Cells
Deep Learning Processor	4K INT8 (upgradable to INT4)
Video Codec (H.264/H.265)	Up to 32 Streams (total resolution ≤ 4Kp60)
Memory	26.6Mb On-Chip SRAM
Security	IEC62443 Security w/HW Root-of-Trust

INTERFACES	
Camera	11 x4 Full MIPI or sub-LVDS Interfaces 1 x4 SLVS-EC Interfaces
USB	4x USB 2.0 / 3.0
Multi-Media	DisplayPort, HDMI
Network	1Gb up to 40Gb Ethernet (w/GigE Vision)
Memory Interface	4GB 64-bit DDR4
Transceivers	4x 12.5Gb/s, 4x 6Gb/s
Mechanical	77 x 60 x 11mm w/ dual 240-pin connectors

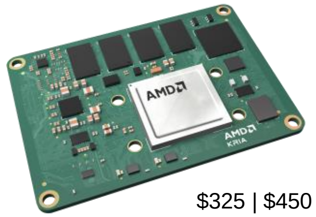
1.4 TOPs

AVNET SILICA

Comparing Kria™ K24 vs. K26 SOM

$250 | $350
(C-grade | I-grade)

CONNECTOR COMPATIBLE



$325 | $450
(C-grade | I-grade)

Cost-optimized SOM for lower power, smaller form-factor & cost sensitive industrial applications

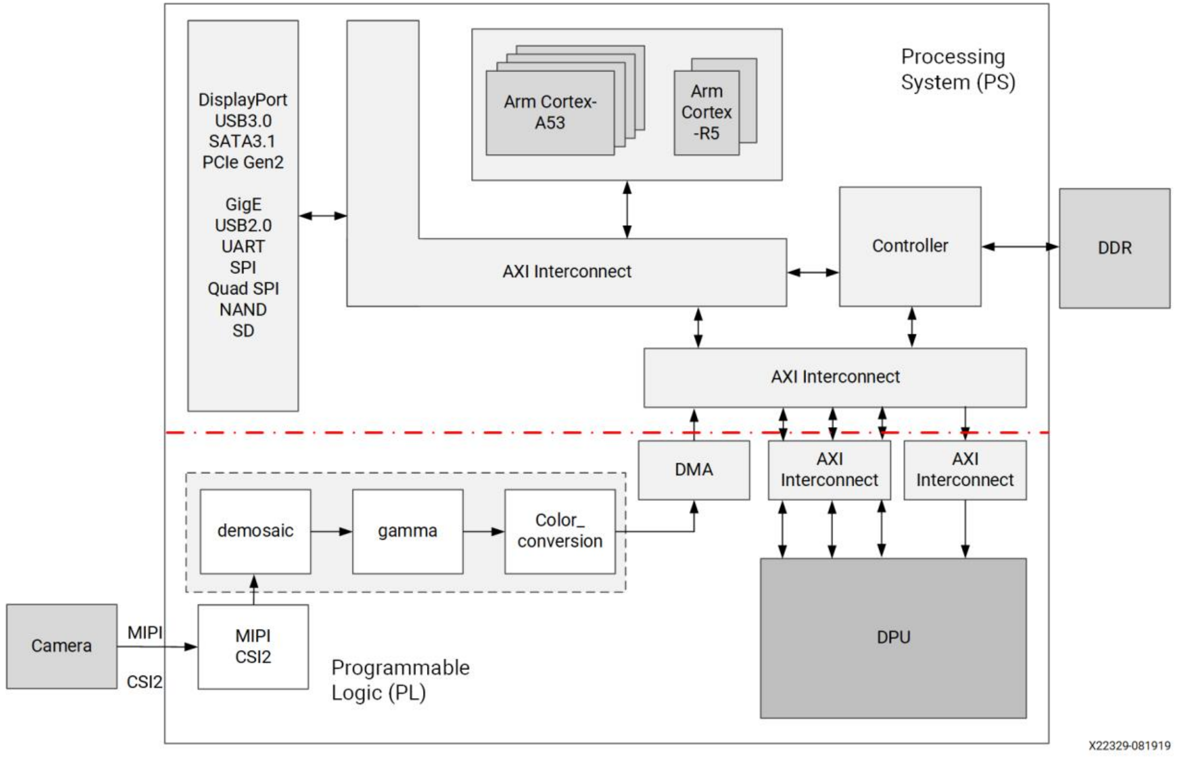
Mid-range SOM for Vision AI and Robotics applications requiring higher performance per watt

	K24 SOM		K26 SOM
SILICON (SYS LOGIC CELLS)	XCK24 InFO (154K)	SILICON	XCK26 (256K)
SOM I/O ACCESS	1x 240-Pin Connector, 1x 40-Pin Connector	SOM I/O	2x 240-Pin Connectors
FORM FACTOR	60 x 42mm	46% SMALL	60 x 77mm
MEMORY	2GB LPDDR4[1], 32 GB	DDR, eMMC	4GB DDR4, 16 GB
POWER[2]	2.5W	51% LESS	5.1W
STARTER KITS	KD240 DRIVES	DEV KITS	KV260 VISION AI, KR260 ROBOTICS

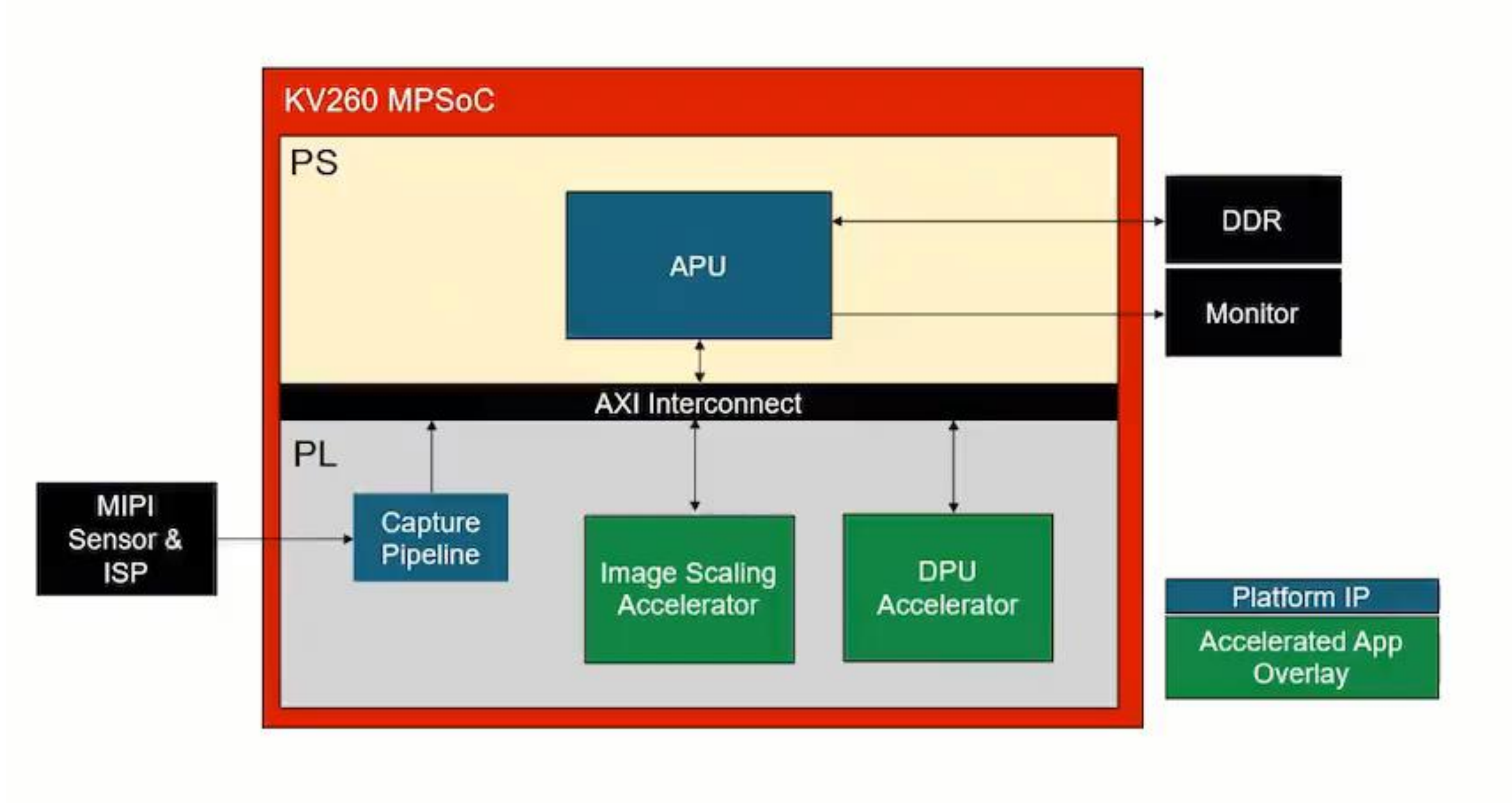
[1] ECC support available on K24 SOM I-grade
[2] Measured power while loading application specific bitstream on the SOM-based starter kit

AVNET SILICA

Typical AI Example System



Example of Dataflow



Zynq™ Ultrascale+™ DPU Compatibility Matrix

Vitis AI Release Version	DPUCZDX8G IP Version	Software Tools Version	Linux Kernel Version Tested
v3.5	4.1 (not updated*)	Vivado / Vitis / PetaLinux 2023.1	6.1
v3.0	4.1	Vivado / Vitis / PetaLinux 2022.2	5.15
v2.5	4.0	Vivado / Vitis / PetaLinux 2022.1	5.15
v2.0	3.4	Vivado / Vitis / PetaLinux 2021.2	5.10
v1.4	3.3	Vivado / Vitis / PetaLinux 2021.1	5.10
v1.3	3.3	Vivado / Vitis / PetaLinux 2020.2	5.4

Vitis AI Release Version	DPUCZDX8G IP Version	Software Tools Version	Linux Kernel Version Tested
v1.2	3.2	Vivado / Vitis / PetaLinux 2020.1	5.4
v1.1	3.2	Vivado / Vitis / PetaLinux 2019.2	4.19
v1.0	3.1	Vivado / Vitis / PetaLinux 2019.1	4.19
N/A (DNNDK)	3.0	Vivado / Vitis / PetaLinux 2019.1	4.19
N/A (DNNDK)	2.0	Vivado / Vitis / PetaLinux 2018.2	4.14
First Release (DNNDK)	1.0	Vivado / Vitis / PetaLinux 2018.1	4.14

AMD Vivado Steps

Goals

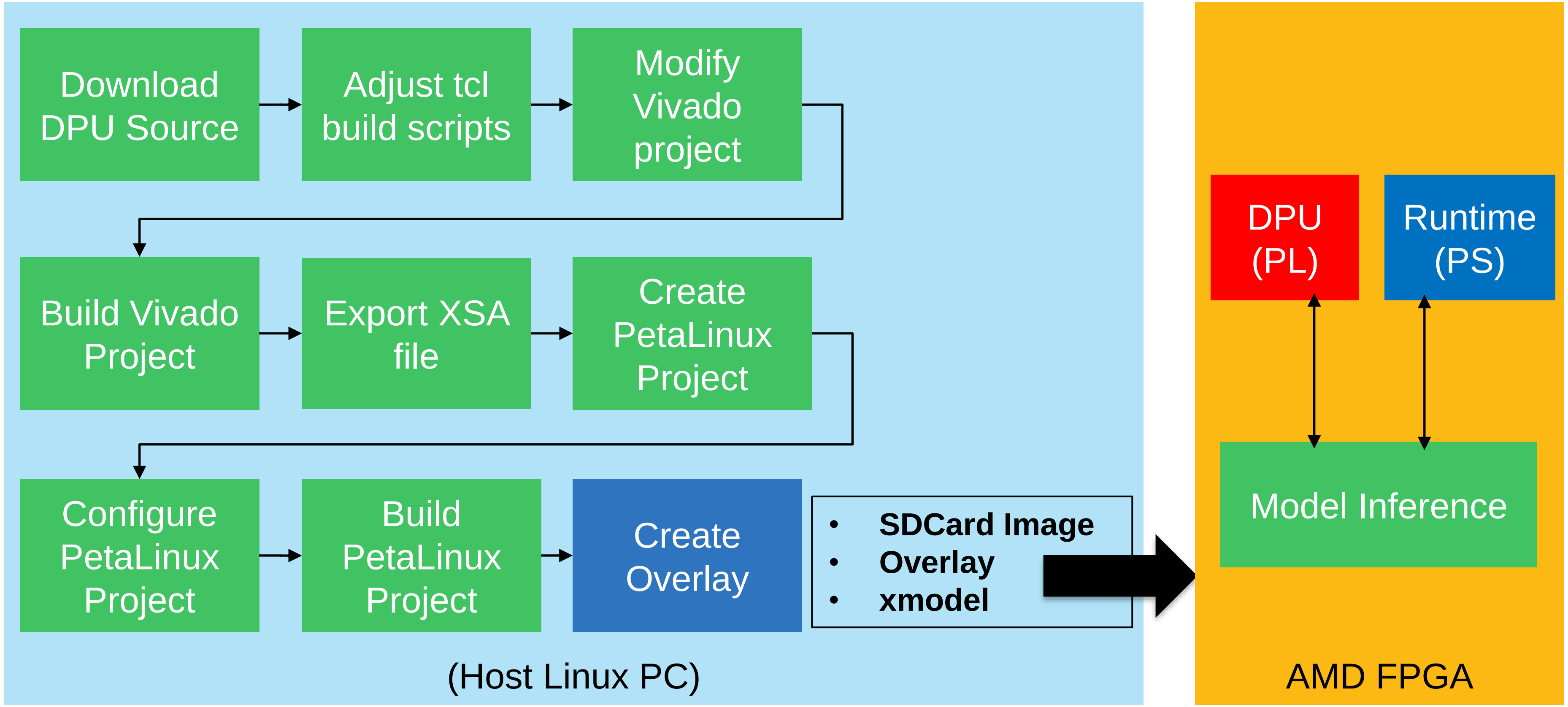
- **Download, adjust and execute AMD-Xilinx tcl scripts for custom Vivado project with following DPU settings (vs pre-build petalinux with DPU images):**

DPU Clock:	325MHz (300MHz originally)
DSP Clock:	650MHz (600MHz originally)
DPU Variant:	B4096
DPU Cores:	x1
UltraRAM:	50/64

- **Create and configure PetaLinux project**
- **Create overlay**
- **Integrate and run custom overlay on KV260 Starter Kit**

Original ref design targets ZCU102 development board

AMD Vivado Flow (KV260)

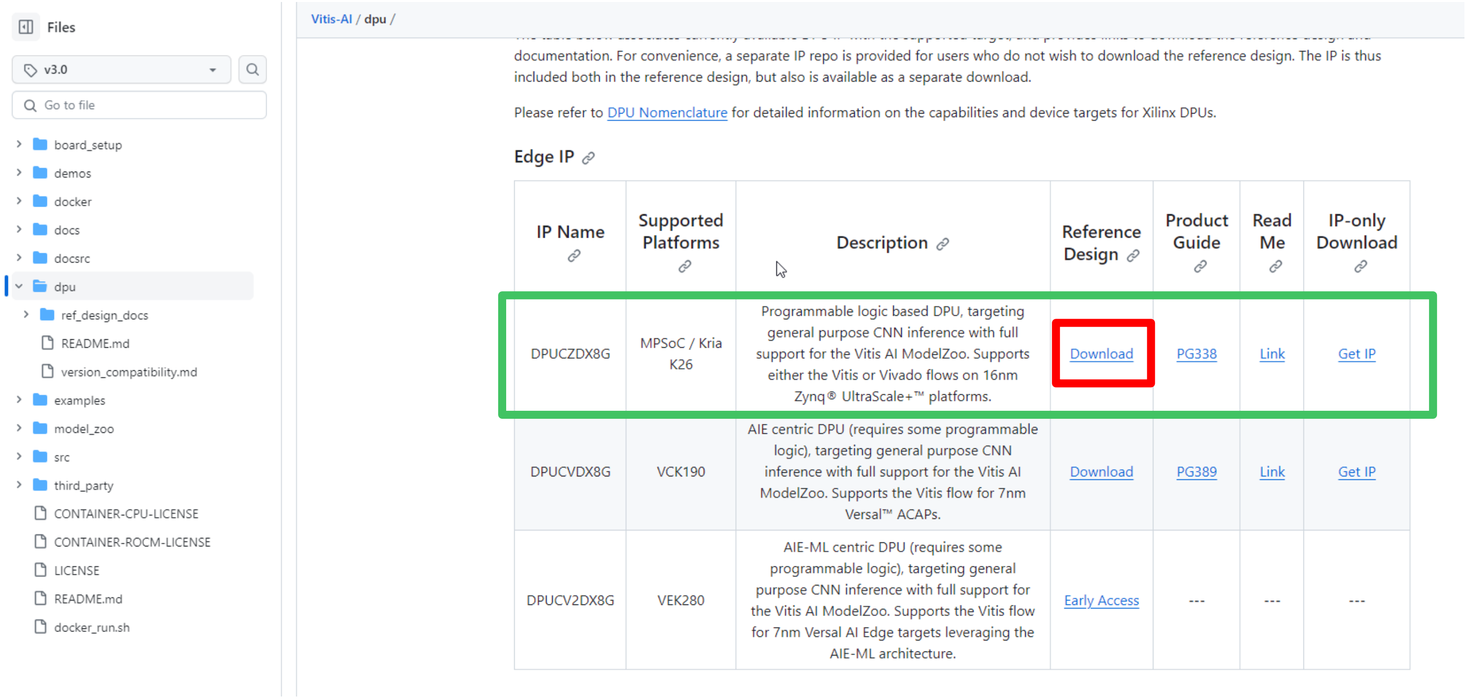


Requirements:

- Vivado 2022.2 (Windows or Linux OS)
- Petalinux 2022.2 (Linux OS)
- KV260 Starter Kit BSP v2022.2 from PetaLinux Download page (link)
- DPUCZDX8G_VAI_v3.0 (v4.1) source files from Vitis-AI GitHub (link)

Download DPU IP

Vitis-AI/dpu at v3.0 · Xilinx/Vitis-AI · GitHub



documentation. For convenience, a separate IP repo is provided for users who do not wish to download the reference design. The IP is thus included both in the reference design, but also is available as a separate download.

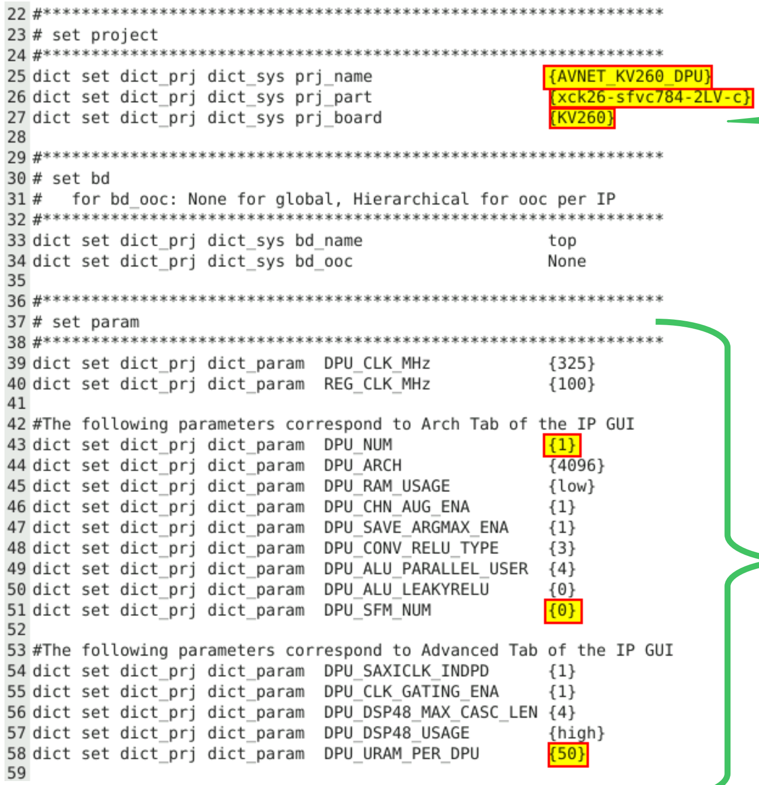
Please refer to DPU Nomenclature for detailed information on the capabilities and device targets for Xilinx DPUs.

Edge IP

IP Name	Supported Platforms	Description	Reference Design	Product Guide	Read Me	IP-only Download
DPUCZDX8G	MPSoC / Kria K26	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.	Download	PG338	Link	Get IP
DPUCVDX8G	VCK190	AIE centric DPU (requires some programmable logic), targeting general purpose CNN inference with full support for the Vitis AI ModelZoo. Supports the Vitis flow for 7nm Versal™ ACAPs.	Download	PG389	Link	Get IP
DPUCV2DX8G	VEK280	AIE-ML centric DPU (requires some programmable logic), targeting general purpose CNN inference with full support for the Vitis AI ModelZoo. Supports the Vitis flow for 7nm Versal AI Edge targets leveraging the AIE-ML architecture.	Early Access	---	---	---

Adjust tcl build scripts

Edit ~/DPUCZDX8G_VAI_v3.0/prj/Vivado/hw/scripts/trd_prj.tcl



```
#****************************************************************
# set project
#****************************************************************
dict set dict_prj dict_sys prj_name                   {AVNET_KV260_DPU}
dict set dict_prj dict_sys prj_part                   {xck26-sfvc784-2LV-c}
dict set dict_prj dict_sys prj_board                  {KV260}

#****************************************************************
# set bd
#   for bd_ooc: None for global, Hierarchical for ooc per IP
#****************************************************************
dict set dict_prj dict_sys bd_name                    top
dict set dict_prj dict_sys bd_ooc                     None

#****************************************************************
# set param
#****************************************************************
dict set dict_prj dict_param  DPU_CLK_MHz             {325}
dict set dict_prj dict_param  REG_CLK_MHz             {100}

#The following parameters correspond to Arch Tab of the IP GUI
dict set dict_prj dict_param  DPU_NUM                 {1}
dict set dict_prj dict_param  DPU_ARCH                {4096}
dict set dict_prj dict_param  DPU_RAM_USAGE           {low}
dict set dict_prj dict_param  DPU_CHN_AUG_ENA         {1}
dict set dict_prj dict_param  DPU_SAVE_ARGMAX_ENA     {1}
dict set dict_prj dict_param  DPU_CONV_RELU_TYPE      {3}
dict set dict_prj dict_param  DPU_ALU_PARALLEL_USER   {4}
dict set dict_prj dict_param  DPU_ALU_LEAKYRELU       {0}
dict set dict_prj dict_param  DPU_SFM_NUM             {0}

#The following parameters correspond to Advanced Tab of the IP GUI
dict set dict_prj dict_param  DPU_SAXICLK_INDPD       {1}
dict set dict_prj dict_param  DPU_CLK_GATING_ENA      {1}
dict set dict_prj dict_param  DPU_DSP48_MAX_CASC_LEN  {4}
dict set dict_prj dict_param  DPU_DSP48_USAGE         {high}
dict set dict_prj dict_param  DPU_URAM_PER_DPU        {50}

```

Project settings

DPU settings

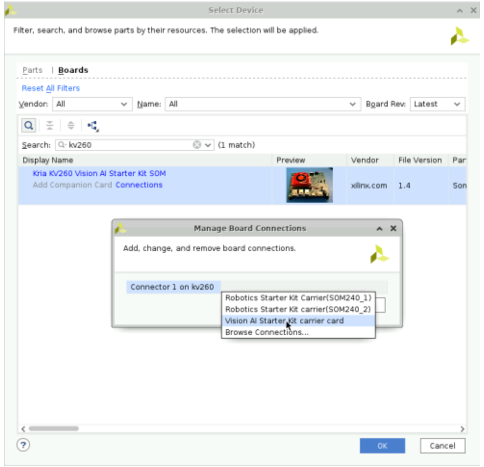
Modify Vivado settings

Create Vivado project via tcl script:

/opt/Xilinx/Vivado/2022.2/bin/vivado -mode tcl -source trd_prj.tcl

Vivado Project modifications:

- **Update project settings**



Select Device

Filter, search, and browse parts by their resources. The selection will be applied.

Parts | **Boards**

Reset All Filters

Vendor: All Name: All Board Rev: Latest

Search: kv260 (1 match)

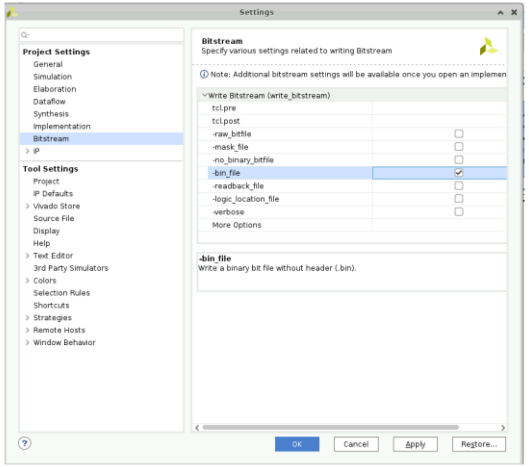
Display Name	Preview	Vendor	File Version	Par
Kria KV260 Vision AI Starter Kit SOM Add Companion Card Connections		xilinx.com	1.4	Son

Manage Board Connections

Add, change, and remove board connections.

Connector 1 on kv260

- Robotics Starter Kit Carrier(SOM240_1)
- Robotics Starter Kit carrier(SOM240_2)
- Vision AI Starter Kit carrier card
- Browse Connections...



Settings

Project Settings
- General
- Simulation
- Elaboration
- Dataflow
- Synthesis
- Implementation
- Bitstream
- IP

Tool Settings
- Project
- IP Defaults
- Vivado Store
- Source File
- Display
- Help
- Text Editor
- 3rd Party Simulators
- Colors
- Selection Rules
- Shortcuts
- Strategies
- Remote Hosts
- Window Behavior

Bitstream
Specify various settings related to writing Bitstream

Note: Additional bitstream settings will be available once you open an implemen

Write Bitstream (write_bitstream)

tcl.pre	
tcl.post	
-raw_bitfile	☐
-mask_file	☐
-no_binary_bitfile	☐
-bin_file	✓
-readback_file	☐
-logic_location_file	☐
-verbose	☐
More Options	

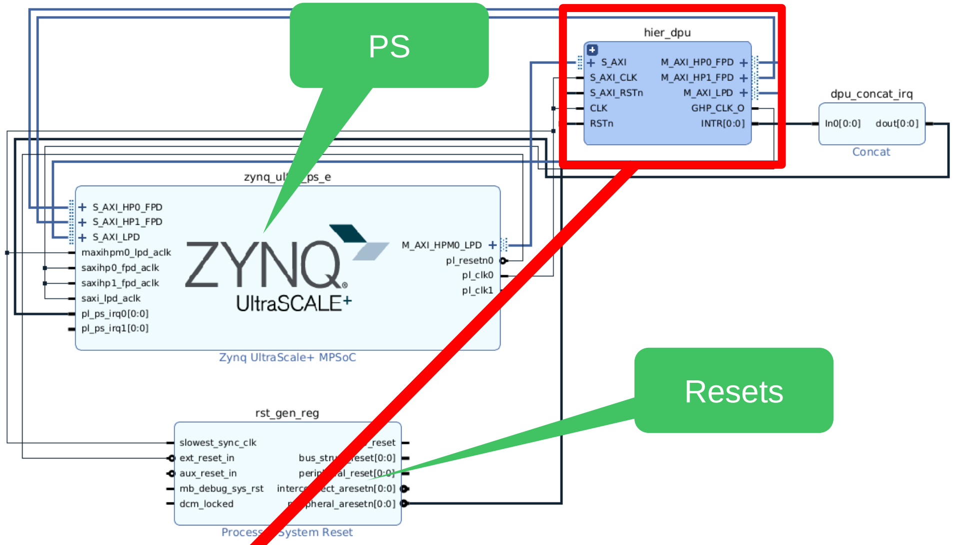
-bin_file
Write a binary bit file without header (.bin).

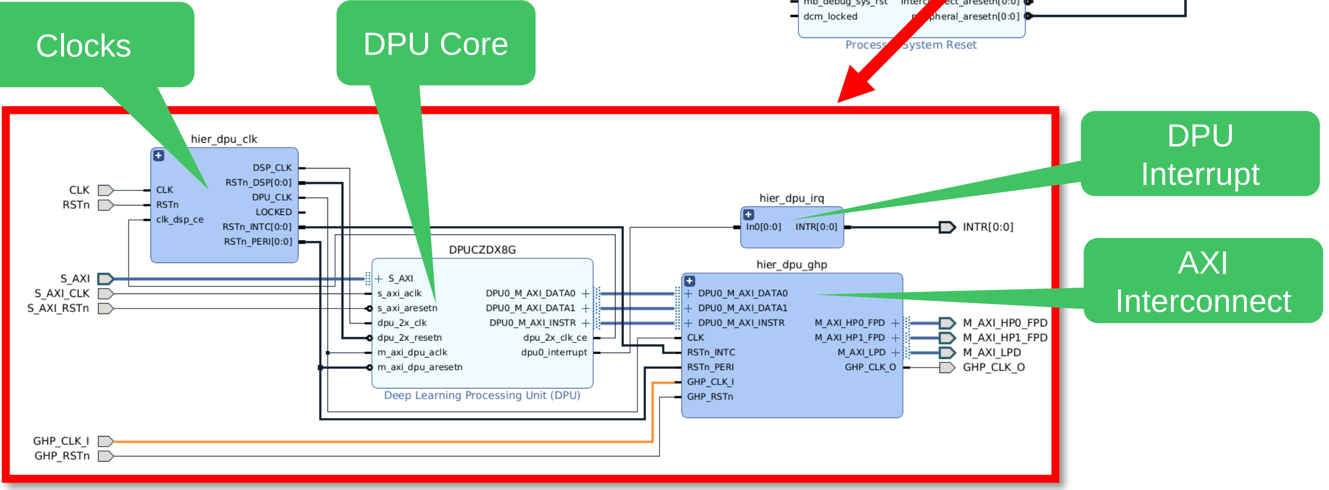
- **Update block diagram IPs**
- **Update Implementation Strategy:** Performance_Explore

Vivado Project Overview

Core Components

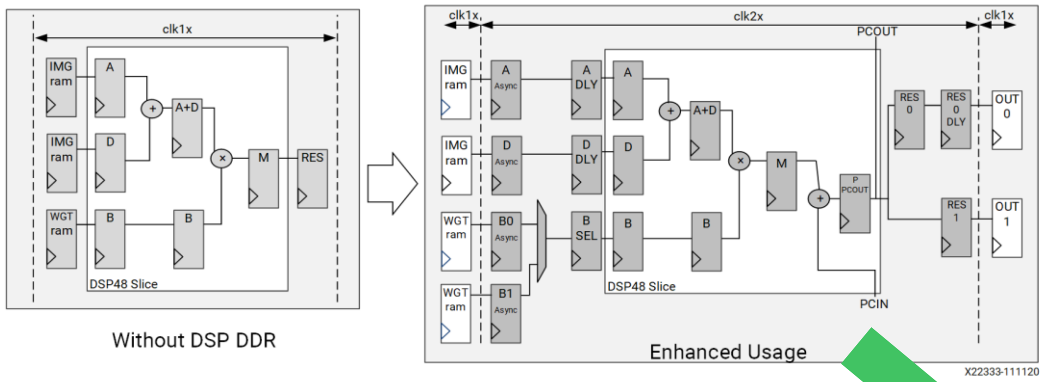
- PS
- DPU (DPUCZDX8G)
- Clocking Wizard (AXI, DPU, 2x_DPU)
- Processor System Resets
- AXI Interconnect

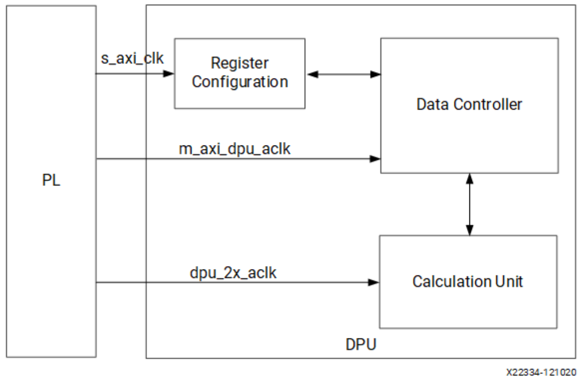




AVNET SILICA

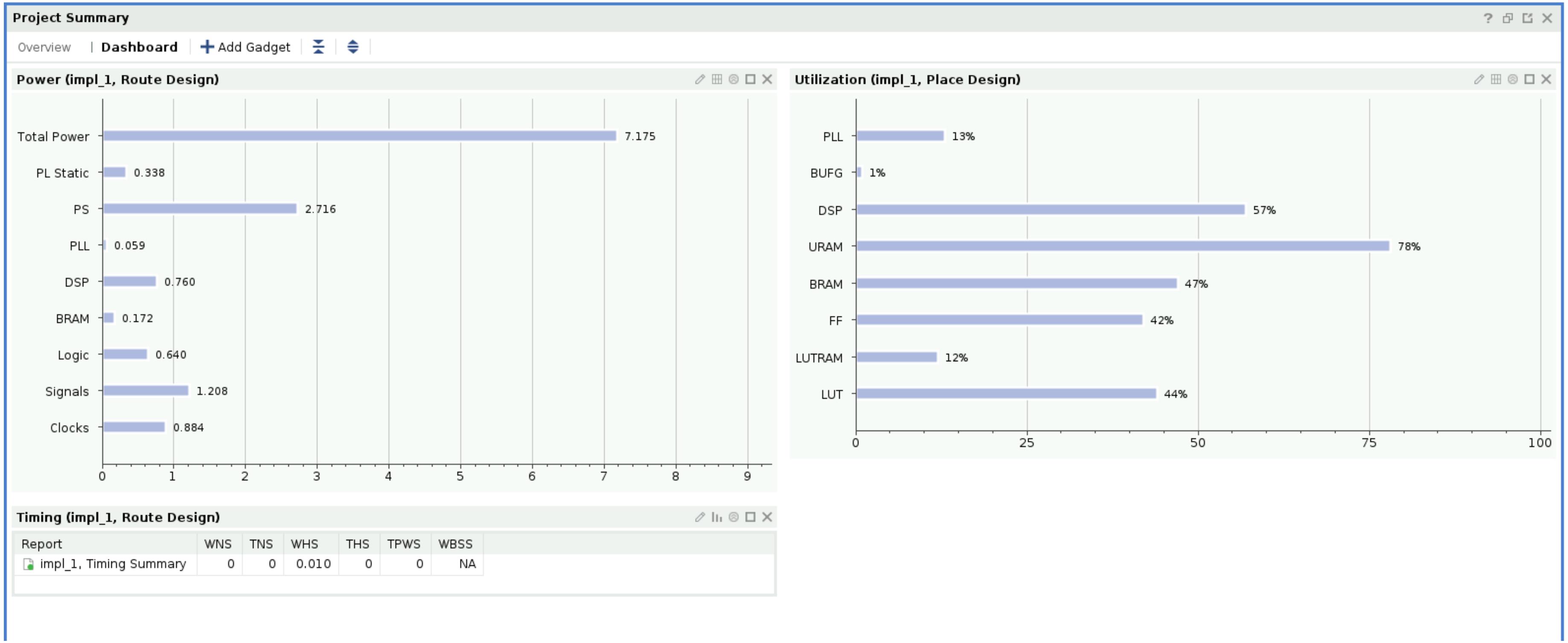
DPUCZDX8G Enhanced Usage





A DSP Double Data Rate (DDR) technique is used to boost the performance.

Vivado Project Compilation Results

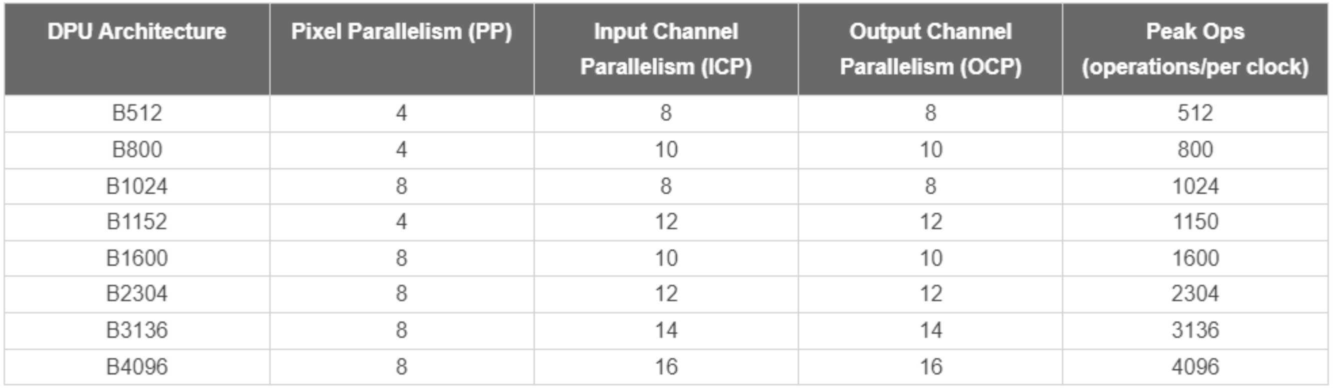


Report	WNS	TNS	WHS	THS	TPWS	WBSS
impl_1, Timing Summary	0	0	0.010	0	0	NA

Things to explore?

- **Experimenting various DPU / clock frequency customizations**

Table: Parallelism for Different Convolution Architectures



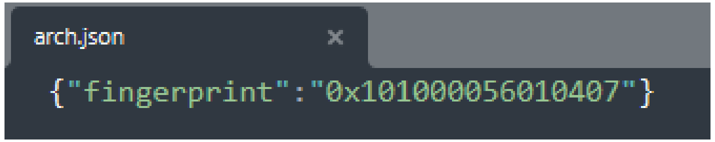
DPU Architecture	Pixel Parallelism (PP)	Input Channel Parallelism (ICP)	Output Channel Parallelism (OCP)	Peak Ops (operations/per clock)
B512	4	8	8	512
B800	4	10	10	800
B1024	8	8	8	1024
B1152	4	12	12	1150
B1600	8	10	10	1600
B2304	8	12	12	2304
B3136	8	14	14	3136
B4096	8	16	16	4096

GOPS = PP*ICP*OCP*2 * Freq

- **Resource Utilizations**
- **Power Reports**

Vivado required output files

1. **arch.json file**
   Required to compile models in Vitis-AI Docker, targeting this DPU configuration, found in: ../DPUCZDX8G_VAI_v3.0/prj/Vivado/hw/srcs/top/ip/top_DPUCZDX8G_0/arch.json



```
{"fingerprint":"0x101000056010407"}
```

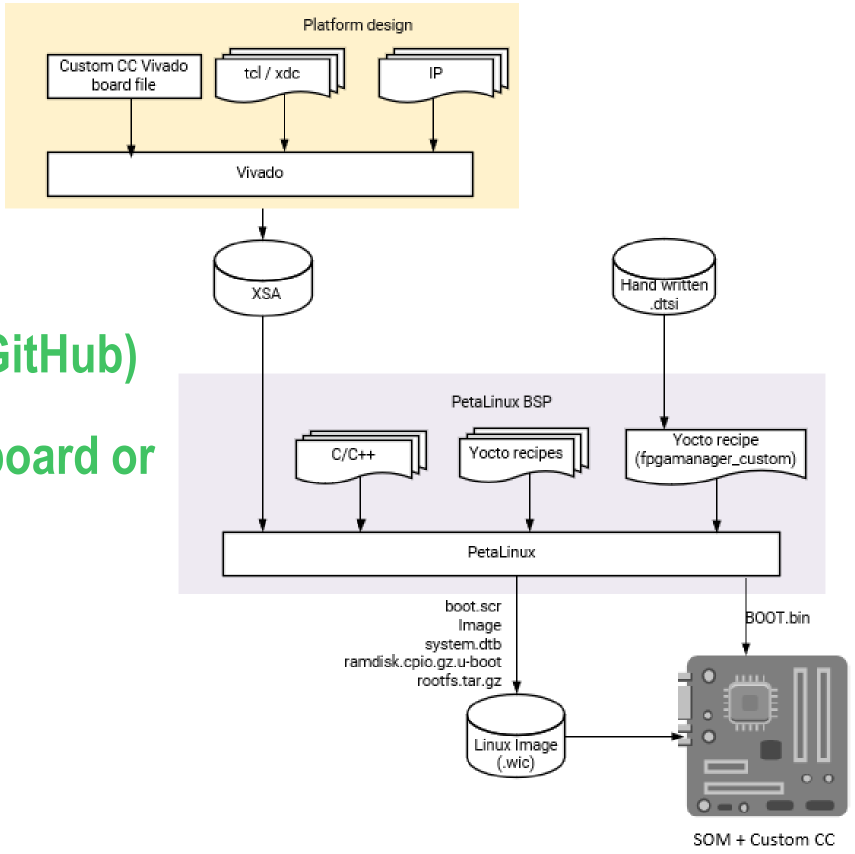
2. ***.xsa file**
   Hardware information of the Vivado design, to be used to create the PetaLinux project in following steps., found in: ../DPUCZDX8G_VAI_v3.0/prj/Vivado/hw/prj/*.xsa

AMD PetaLinux Steps

PetaLinux Flow

1. **v2022.2 includes PetaLinux build script**
2. **Previous versions, included steps (e.g. GitHub)**
3. **Script needs to be adjusted for custom board or in this case KV260 Starter Kit**
   (adjusted script “kv260_bsp_help.sh” can be provided by Avnet Silica)

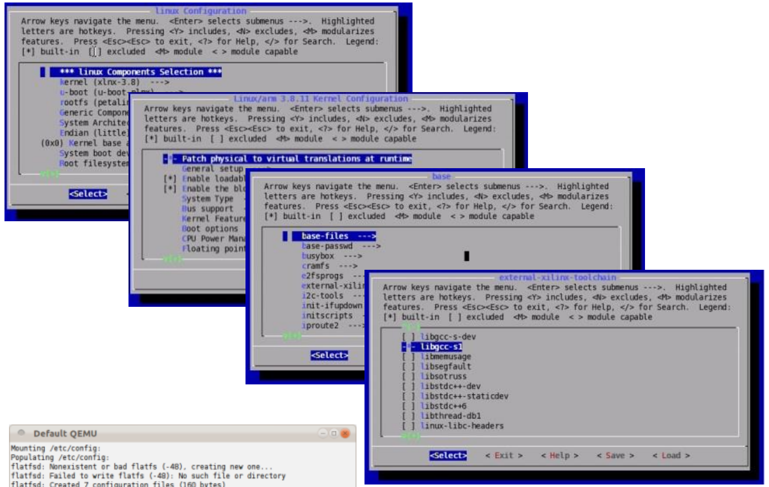
Summary of changes in next slides



PetaLinux Development Flow in short

Customize Linux in five steps:

- **`petalinux-create`**
  - Create a new project or components
- **`petalinux-config`**
  - Configure the project or specified components
  - Kernel, file system, libraries, debug options
  - Get hardware description from the Vivado® IDE
- **`petalinux-build`**
  - Build project or specified components
  - Kernel, file system, project-specific components
- **`petalinux-boot`**
  - Import FSBL and bitstream
  - Build image.ub, BOOT.BIN, etc.
  - Deploy to QEMU or hardware
- **`petalinux-package`**
  - Package custom image formats, firmware, prebuilt images & BSP for distribution



```
linux Configuration
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
[*] built-in  [ ] excluded  <M> module  < > module capable

      *** linux Components Selection ***
      kernel (xlnx-3.8)  --->
      u-boot (u-boot-
      rootfs (petali
      Generic Compone
      System Archite
      Endian (little)
(0x0) Kernel base a
      System boot dev
      Root filesystem
```

```
Linux/arm 3.8.11 Kernel Configuration
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
[*] built-in  [ ] excluded  <M> module  < > module capable

    -*- Patch physical to virtual translations at runtime
        General setup
    [*] Enable loadab
    [*] Enable the bl
        System Type
        Bus support
        Kernel Featur
        Boot options
        CPU Power Mana
        Floating point
```

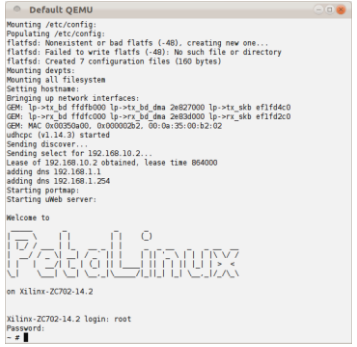
```
base
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
[*] built-in  [ ] excluded  <M> module  < > module capable

      base-files  --->
      base-passwd  --->
      busybox  --->
      cramfs  --->
      e2fsprogs  --
      external-xilin
      i2c-tools  --
      init-ifupdown
      initscripts
      iproute2  ---
```

```
external-xilinx-toolchain
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend:
[*] built-in  [ ] excluded  <M> module  < > module capable

      [ ] libgcc-s-dev
      -*- libgcc-s1
      [ ] libmemusage
      [ ] libsegfault
      [ ] libsotruss
      [ ] libstdc++-dev
      [ ] libstdc++-staticdev
      [ ] libstdc++6
      [ ] libthread-db1
      [ ] linux-libc-headers

      <Select>    < Exit >    < Help >    < Save >    < Load >
```



Default QEMU

```
Mounting /etc/config:
Populating /etc/config:
flatfsd: Nonexistent or bad flatfs (-48), creating new one...
flatfsd: Failed to write flatfs (-48): No such file or directory
flatfsd: Created 7 configuration files (160 bytes)
Mounting devpts:
Mounting all filesystem
Setting hostname:
Bringing up network interfaces:
GEM: lp->tx_bd ffdfb000 lp->tx_bd_dma 2e827000 lp->tx_skb ef1fd4c0
GEM: lp->rx_bd ffdfc000 lp->rx_bd_dma 2e83d000 lp->rx_skb ef1fd2c0
GEM: MAC 0x00350a00, 0x000002b2, 00:0a:35:00:b2:02
udhcpc (v1.14.3) started
Sending discover...
Sending select for 192.168.10.2...
Lease of 192.168.10.2 obtained, lease time 864000
adding dns 192.168.1.1
adding dns 192.168.1.254
Starting portmap:
Starting uWeb server:

Welcome to

PetaLinux

on Xilinx-ZC702-14.2


Xilinx-ZC702-14.2 login: root
Password:
~ #
```

Create PetaLinux Project

1. Create PetaLinux Project based on BSP

```
petalinux-create -t project -s $bsp -n $prj_name
```

2. Update PetaLinux project with custom XSA

```
petalinux-config --get-hw-description=${xsa_dir} --silentconfig
```

Core PetaLinux Customizations for KV260

1. petalinux-config
   # Enable FPGA Manager
2. petalinux-config -c kernel
   # Enable DPU Driver
3. petalinux-config -c rootfs
   # Disable ZOCL and XRT (for use with Vitis Acceleration flow)
   # Enable auto-log in
   # Enable image feature package management

Create overlay for KV260

1. **Device Tree (DT) Creation (in XSCT)**
   createdts -hw <XSA_FILE>.xsa -zocl -platform-name <PLATFORM NAME> -git-branch <xlnx_rel_vXXXX> -overlay -compile -out <OUTPUT_NAME>

2. **Device Tree (DT) Compilation**
   dtc -@ -O dtb -o <OUTPUT>.dtbo <OUTPUT_NAME>/<OUTPUT_NAME>/<PLATFORM NAME>/psu_cortexa53_0/device_tree_domain/bsp/pl.dtsi

3. **Create shell.json file:**
   echo '{ "shell_type" : "XRT_FLAT", "num_slots": "1" }' > shell.json

OVERLAY

- kv260.bit.bin (PL configuration file)
- kv260.dtbo (compiled DT)
- shell.json (overlay shell file)

Build PetaLinux Project

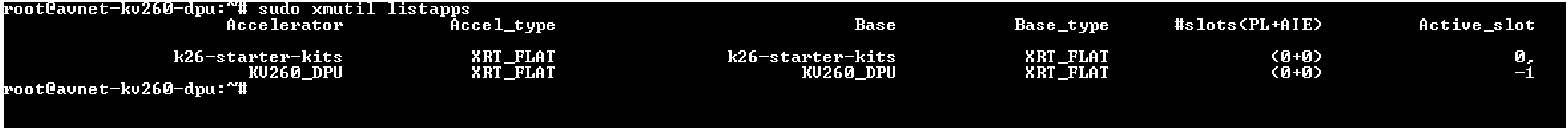
1. **Build PetaLinux**
   petalinux-config -c kernel –silentconfig
   petalinux-config -c rootfs –silentconfig
   petalinux-build

2. **Create SD Card**
   petalinux-package --wic --images-dir images/linux/ --bootfiles "ramdisk.cpio.gz.uboot, boot.scr, Image, system.dtb, system-zynqmp-sck-kv-g-revB.dtb“ --disk-name "mmcblk1" --wic-extra-args "-c gzip"

KV260 Deployment

KV260 – Load Overlay

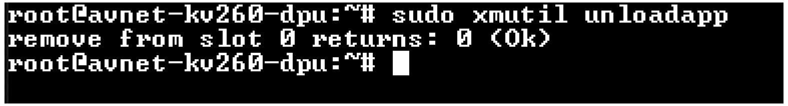
- **Copy overlay folder "KV260_DPU" to KV260 board**
  sudo mv ./KV260_DPU /lib/firmware/xilinx/
- **Check available apps:**
  sudo xmutil listapps



```
root@avnet-kv260-dpu:~# sudo xmutil listapps
                     Accelerator          Accel_type                            Base           Base_type        #slots(PL+AIE)         Active_slot

                k26-starter-kits            XRT_FLAT                k26-starter-kits            XRT_FLAT                (0+0)                  0,
                       KV260_DPU            XRT_FLAT                       KV260_DPU            XRT_FLAT                (0+0)                  -1
root@avnet-kv260-dpu:~#
```

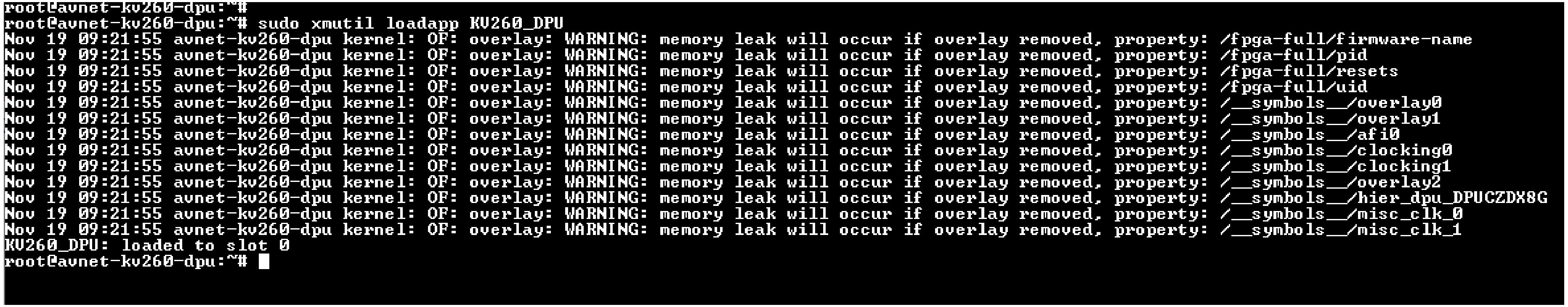
- **Unload current app:**
  sudo xmutil unloadapp



```
root@avnet-kv260-dpu:~# sudo xmutil unloadapp
remove from slot 0 returns: 0 (Ok)
root@avnet-kv260-dpu:~#
```

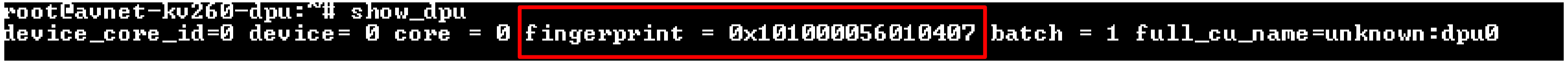
KV260 – Load Overlay

- **Load our custom DPU app:**
  sudo xmutil loadapp KV260_DPU



```
root@avnet-kv260-dpu:~#
root@avnet-kv260-dpu:~# sudo xmutil loadapp KV260_DPU
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/pid
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/uid
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay0
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay1
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afi0
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking1
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay2
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/hier_dpu_DPUCZDX8G
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/misc_clk_0
Nov 19 09:21:55 avnet-kv260-dpu kernel: OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/misc_clk_1
KV260_DPU: loaded to slot 0
root@avnet-kv260-dpu:~#
```

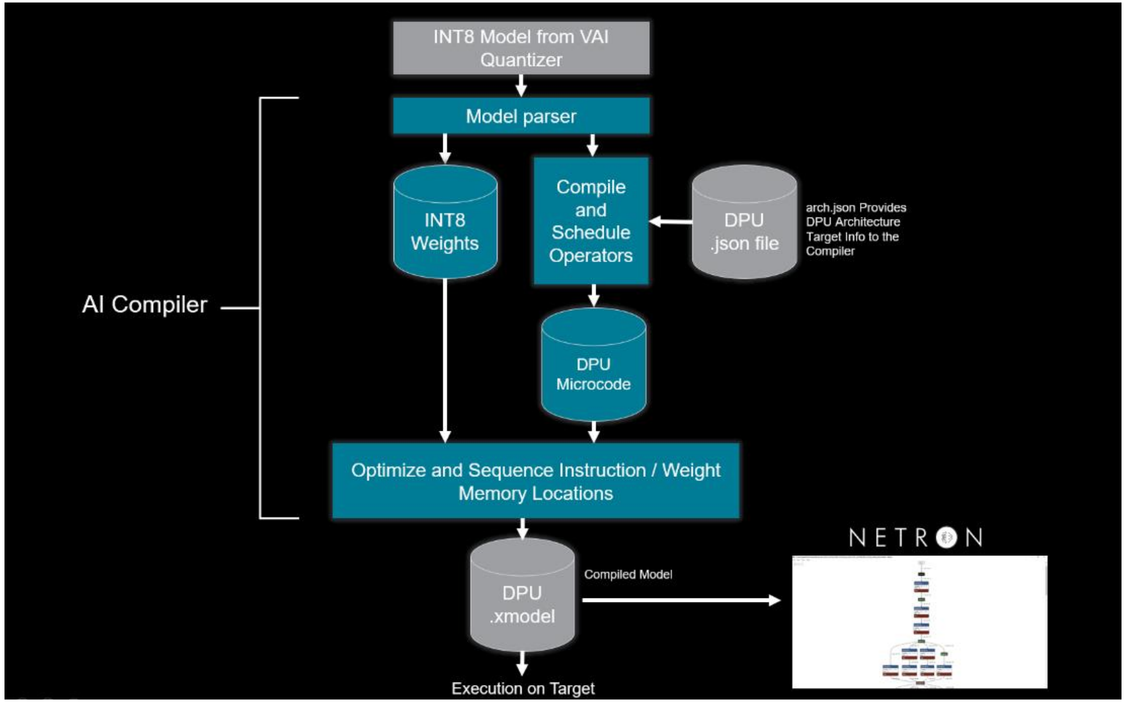
Compile for target AMD Hardware



```
root@avnet-kv260-dpu:~# show_dpu
device_core_id=0 device= 0 core = 0 fingerprint = 0x101000056010407 batch = 1 full_cu_name=unknown:dpu0
```

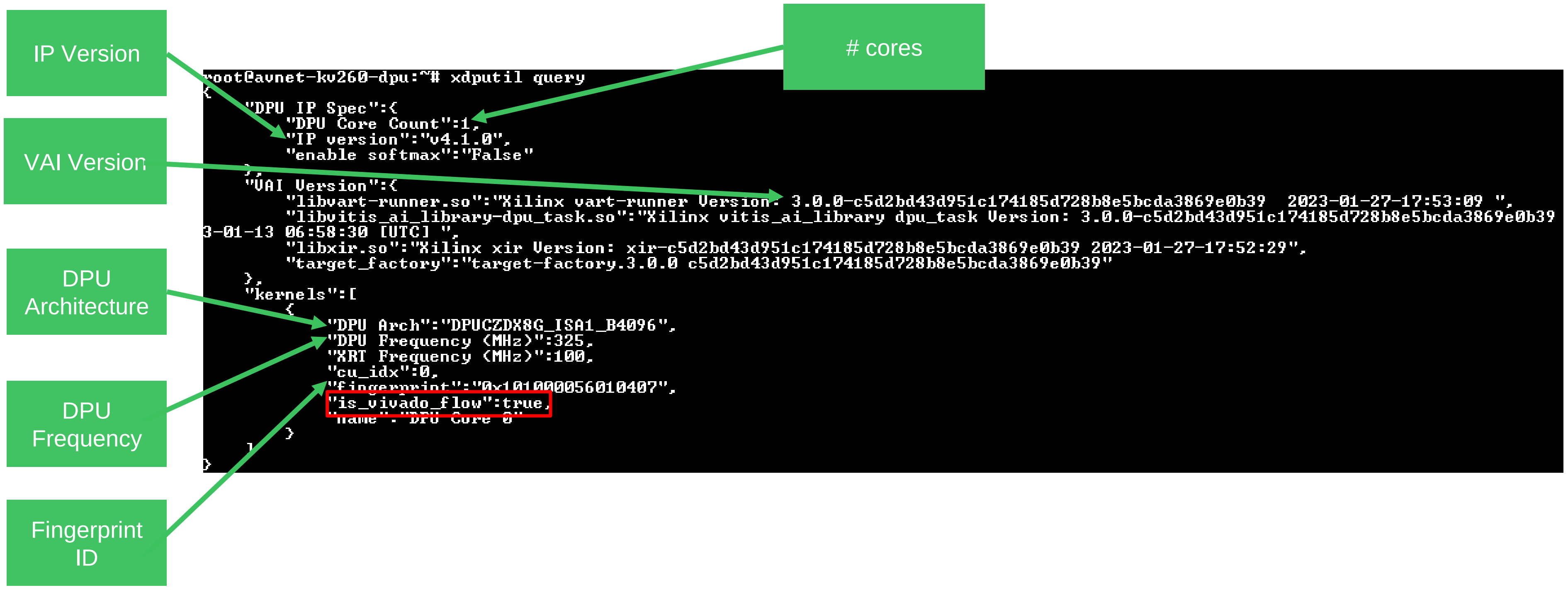
Requirements:

- Quantized Model
- DPU *.json file


INT8 Model from VAI Quantizer
Model parser
INT8 Weights
Compile and Schedule Operators
DPU .json file
arch.json Provides DPU Architecture Target Info to the Compiler
AI Compiler
DPU Microcode
Optimize and Sequence Instruction / Weight Memory Locations
NETRON
DPU .xmodel
Compiled Model
Execution on Target

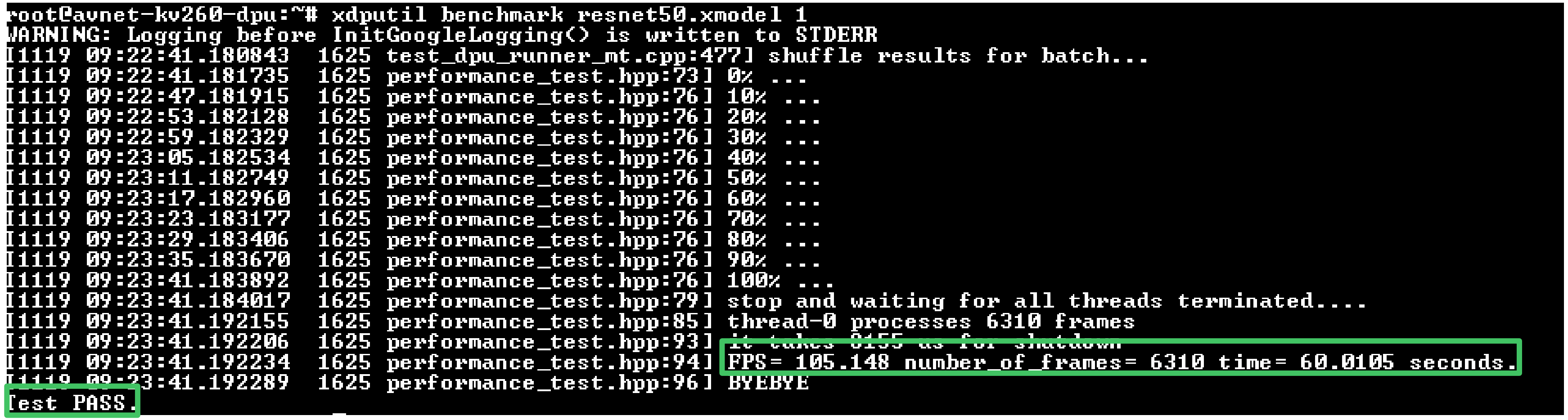
KV260 – Quick DPU Check

Check DPU settings with: **xdputil query**



```
root@avnet-kv260-dpu:~# xdputil query
{
    "DPU IP Spec":{
        "DPU Core Count":1,
        "IP version":"v4.1.0",
        "enable softmax":"False"
    },
    "VAI Version":{
        "libvart-runner.so":"Xilinx vart-runner Version: 3.0.0-c5d2bd43d951c174185d728b8e5bcda3869e0b39  2023-01-27-17:53:09 ",
        "libvitis_ai_library-dpu_task.so":"Xilinx vitis_ai_library dpu_task Version: 3.0.0-c5d2bd43d951c174185d728b8e5bcda3869e0b39
3-01-13 06:58:30 [UTC] ",
        "libxir.so":"Xilinx xir Version: xir-c5d2bd43d951c174185d728b8e5bcda3869e0b39 2023-01-27-17:52:29",
        "target_factory":"target-factory.3.0.0 c5d2bd43d951c174185d728b8e5bcda3869e0b39"
    },
    "kernels":[
        {
            "DPU Arch":"DPUCZDX8G_ISA1_B4096",
            "DPU Frequency (MHz)":325,
            "XRT Frequency (MHz)":100,
            "cu_idx":0,
            "fingerprint":"0x101000056010407",
            "is_vivado_flow":true,
            "name":"DPU Core 0"
        }
    ]
}
```

KV260 – Quick DPU Benchmark



```
root@avnet-kv260-dpu:~# xdputil benchmark resnet50.xmodel 1
WARNING: Logging before InitGoogleLogging() is written to STDERR
I1119 09:22:41.180843  1625 test_dpu_runner_mt.cpp:477] shuffle results for batch...
I1119 09:22:41.181735  1625 performance_test.hpp:73] 0% ...
I1119 09:22:47.181915  1625 performance_test.hpp:76] 10% ...
I1119 09:22:53.182128  1625 performance_test.hpp:76] 20% ...
I1119 09:22:59.182329  1625 performance_test.hpp:76] 30% ...
I1119 09:23:05.182534  1625 performance_test.hpp:76] 40% ...
I1119 09:23:11.182749  1625 performance_test.hpp:76] 50% ...
I1119 09:23:17.182960  1625 performance_test.hpp:76] 60% ...
I1119 09:23:23.183177  1625 performance_test.hpp:76] 70% ...
I1119 09:23:29.183406  1625 performance_test.hpp:76] 80% ...
I1119 09:23:35.183670  1625 performance_test.hpp:76] 90% ...
I1119 09:23:41.183892  1625 performance_test.hpp:76] 100% ...
I1119 09:23:41.184017  1625 performance_test.hpp:79] stop and waiting for all threads terminated....
I1119 09:23:41.192155  1625 performance_test.hpp:85] thread-0 processes 6310 frames
I1119 09:23:41.192206  1625 performance_test.hpp:93] it takes 8155 us for shutdown
I1119 09:23:41.192234  1625 performance_test.hpp:94] FPS= 105.148 number_of_frames= 6310 time= 60.0105 seconds.
I1119 09:23:41.192289  1625 performance_test.hpp:96] BYEBYE
Test PASS.
```

+6% FPS by increasing the clock rate from 300MHz to 325MHz for ResNet50

DPU Clock / DSP Clock (MHz)	FPS
300 / 600	98.9
325 / 650	105.1

Q&A

Typical Questions

Are all the components of Vitis AI free?

Yes!

As of the 3.5 release all components are free!

For releases <3.5, the Vitis AI Optimizer does require a separate license which can be obtained free-of-charge upon request.

Is Vitis AI a separate download?

Yes! Users can get started by cloning the Vitis AI GitHub repository:

GitHub - Xilinx/Vitis-AI: Vitis AI is Xilinx’s development stack for AI inference on Xilinx hardware platforms, including both edge devices and Alveo cards.

What Xilinx Target Device Families and Platforms does Vitis AI Support?

Vitis AI DPUs are available for:

- **Zynq 7000**
- **Zynq Ultrascale+ MPSoC**
- **Versal AI Edge**
- **Versal AI Core**

How does FPGA compare to CPU and GPU acceleration?

FPGA accelerated networks can run up to 90x faster as compared to CPU. FPGA accelerated networks are on par with GPU accelerated networks for throughput critical applications yet provide support for more custom applications.

FPGA accelerated networks are far superior to GPU accelerated networks for latency critical applications such as autonomous driving.

Lot of scope in reducing on a system level power / cost and creating scalable designs.

Is it possible to deploy the DPUCZ using Yocto flows, or even Ubuntu, rather than PetaLinux?

Yes!

What is important to consider is that each release of the Vitis AI tool and the DPUCZ IP is provided with drivers and a runtime that targets a specific Linux kernel release. Misalignment between the target kernel version can pose challenges and may require extensive code changes.

My DPU implementation does not meet my latency/throughput targets. Is there anything else I can do?

Yes!

Besides modifying architecture and/or taking advantage of pruning within Vitis AI, you can also explore FINN.

FINN implements each layer of a neural network separately and creates like this a custom very low latency and high throughput design in the PL.

Thank You