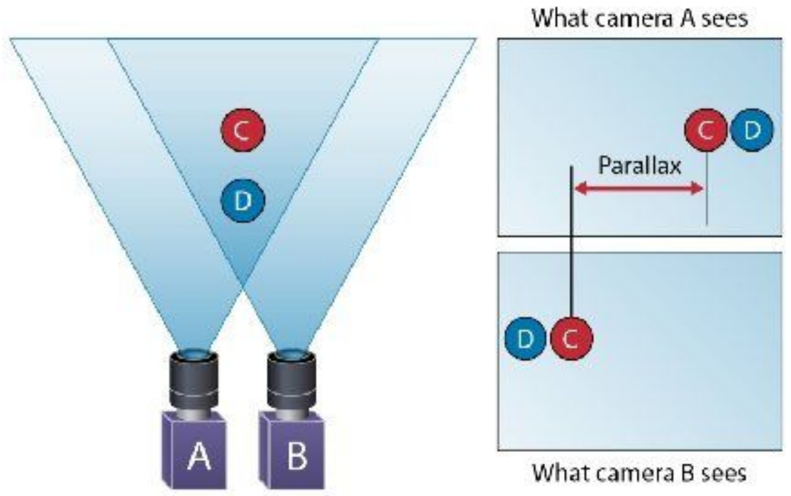
Stereo Vision on FPGA

Final Presentation (12/14)

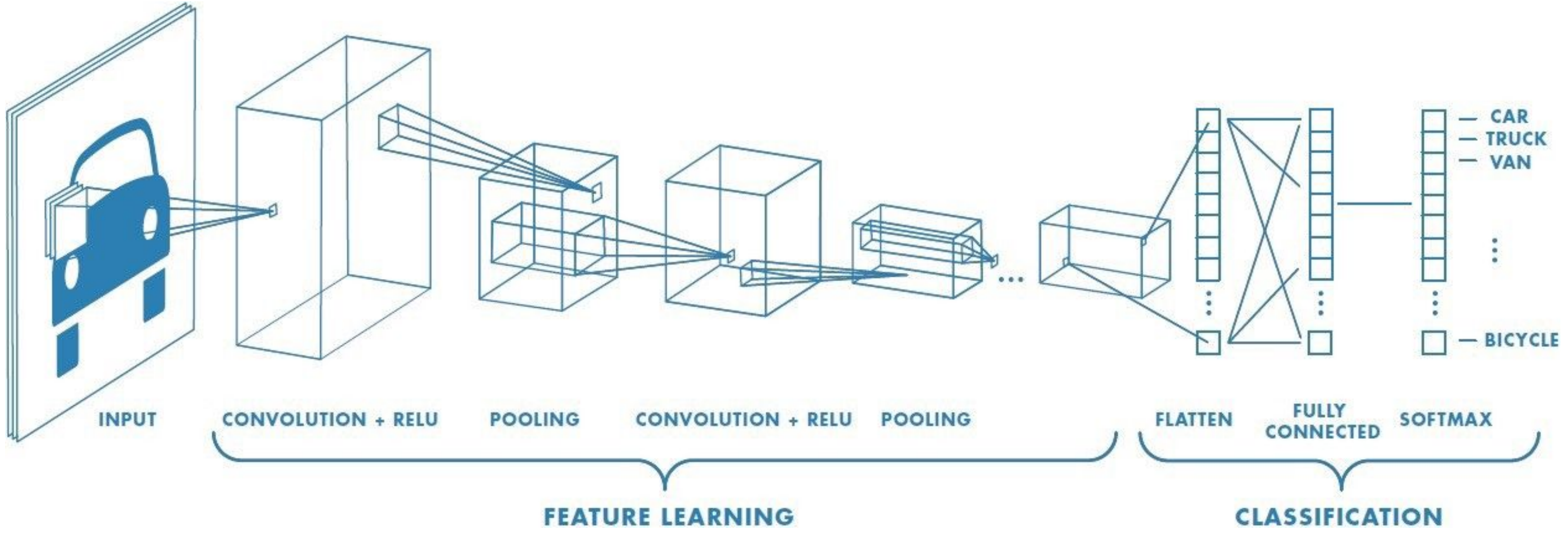
Alejandro Perez-Vicente, David Hernandez

Outline

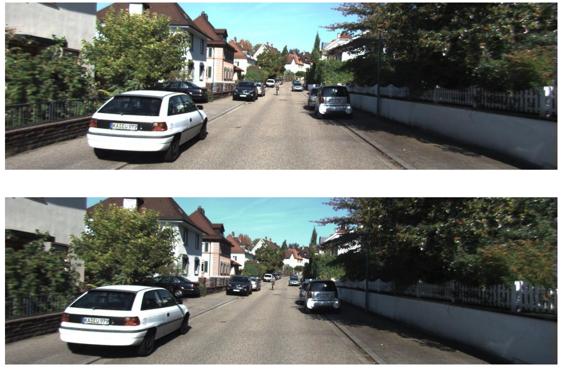
What is Stereo Vision?

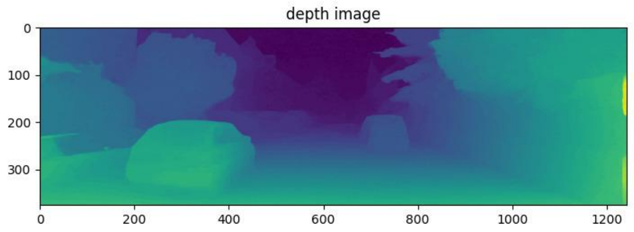


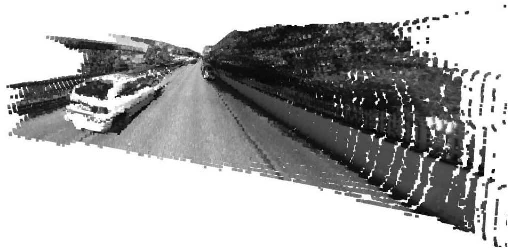
What is a Convolutional Neural Network



Objective





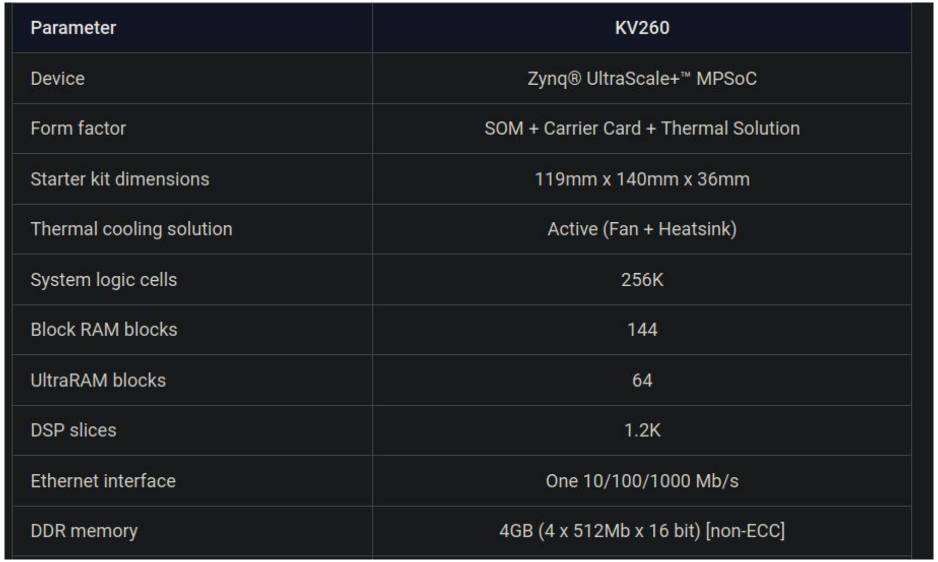


Images from Stereo Camera

Depth Map

Point Cloud

Hardware Equipment



Parameter	KV260
Device	Zynq® UltraScale+™ MPSoC
Form factor	SOM + Carrier Card + Thermal Solution
Starter kit dimensions	119mm x 140mm x 36mm
Thermal cooling solution	Active (Fan + Heatsink)
System logic cells	256K
Block RAM blocks	144
UltraRAM blocks	64
DSP slices	1.2K
Ethernet interface	One 10/100/1000 Mb/s
DDR memory	4GB (4 x 512Mb x 16 bit) [non-ECC]

KV260 Board

Hardware Platform Design Tools

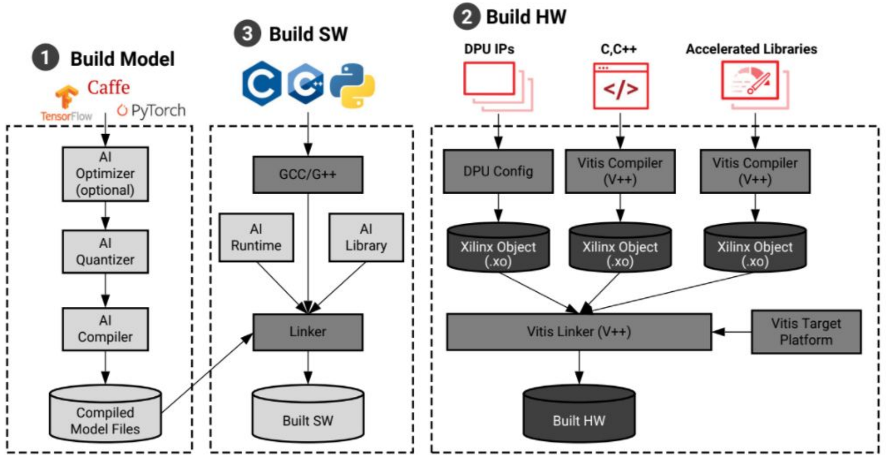
VIVADO™

PetaLinux

XILINX
VITIS™

Vitis AI Integration Workflow

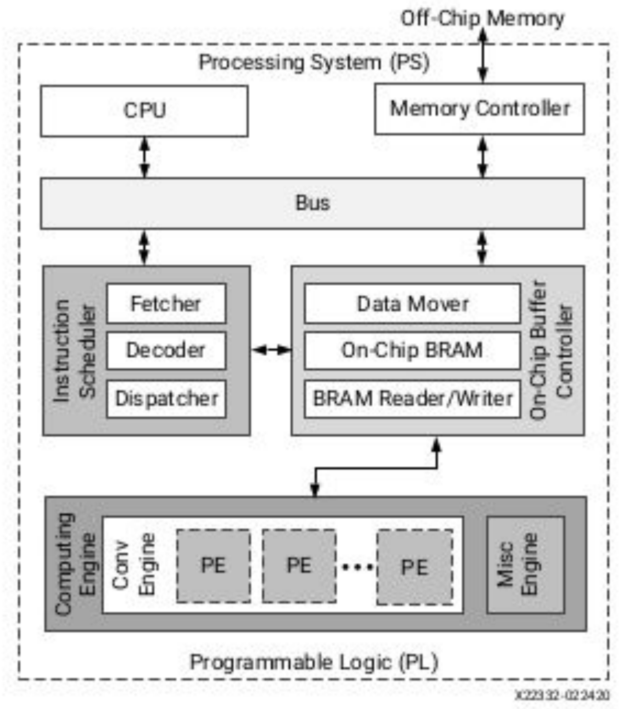
- Define Neural Network Architecture on Tensorflow/Pytorch framework
- Generate C++ code for the AI runtime on the FPGA accelerator by using Vitis AI precompiled libraries.
- Load Vitis DPU IPs, C++ runtimes and accelerated libraries onto the Petalinux custom board Image.



Vitis AI Workflow

Xilinx Deep Learning Processing Unit

- Xilinx offers several IP cores destined for AI/Deep Learning acceleration on FPGA logic.
- The DPU used for this project is optimized for Zynq UltraScale+.



DPU Configurations

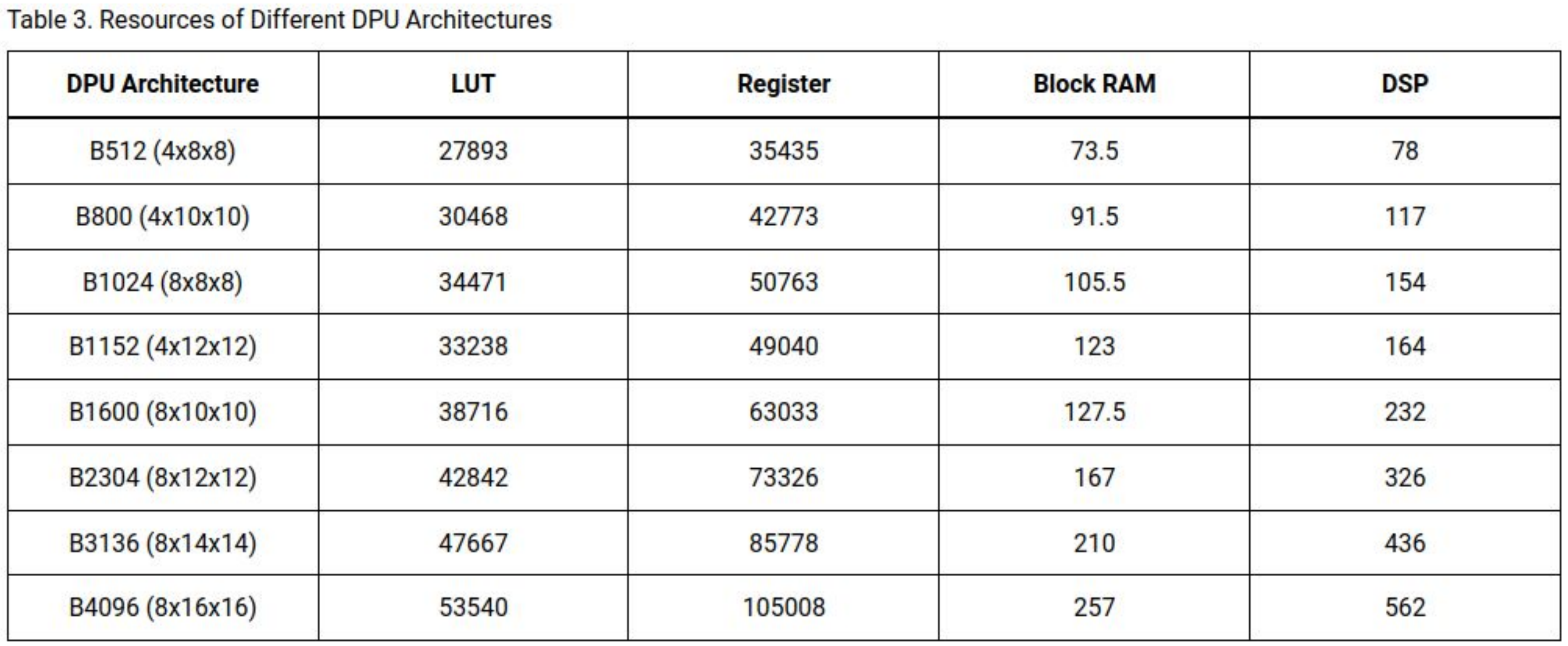


Table 3. Resources of Different DPU Architectures

DPU Architecture	LUT	Register	Block RAM	DSP
B512 (4x8x8)	27893	35435	73.5	78
B800 (4x10x10)	30468	42773	91.5	117
B1024 (8x8x8)	34471	50763	105.5	154
B1152 (4x12x12)	33238	49040	123	164
B1600 (8x10x10)	38716	63033	127.5	232
B2304 (8x12x12)	42842	73326	167	326
B3136 (8x14x14)	47667	85778	210	436
B4096 (8x16x16)	53540	105008	257	562

FADNet

- Fast Accurate Disparity estimation Network
- Mostly formed by 2D based convolutional neural networks
- Operations inside the NN architecture are easy to compile, optimize and deploy on hardware accelerators

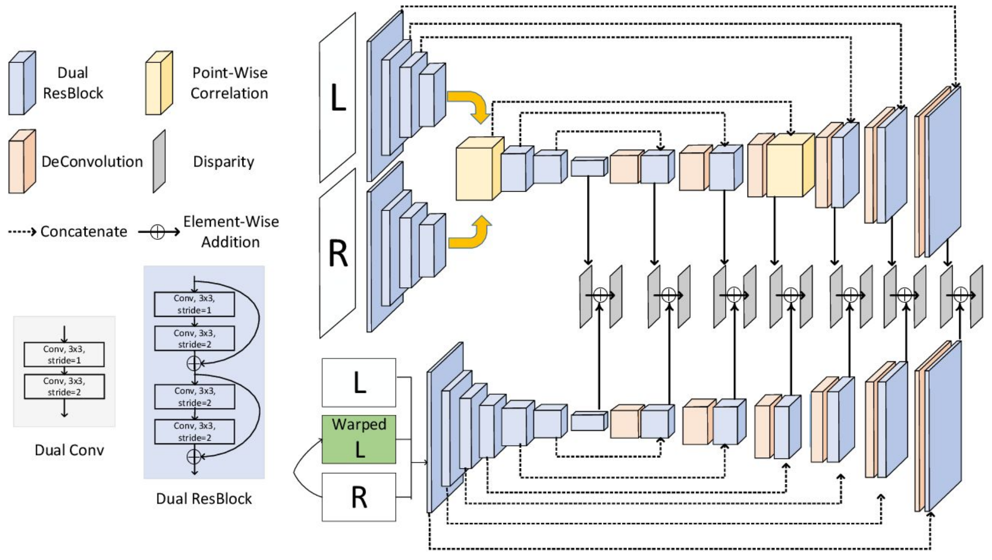
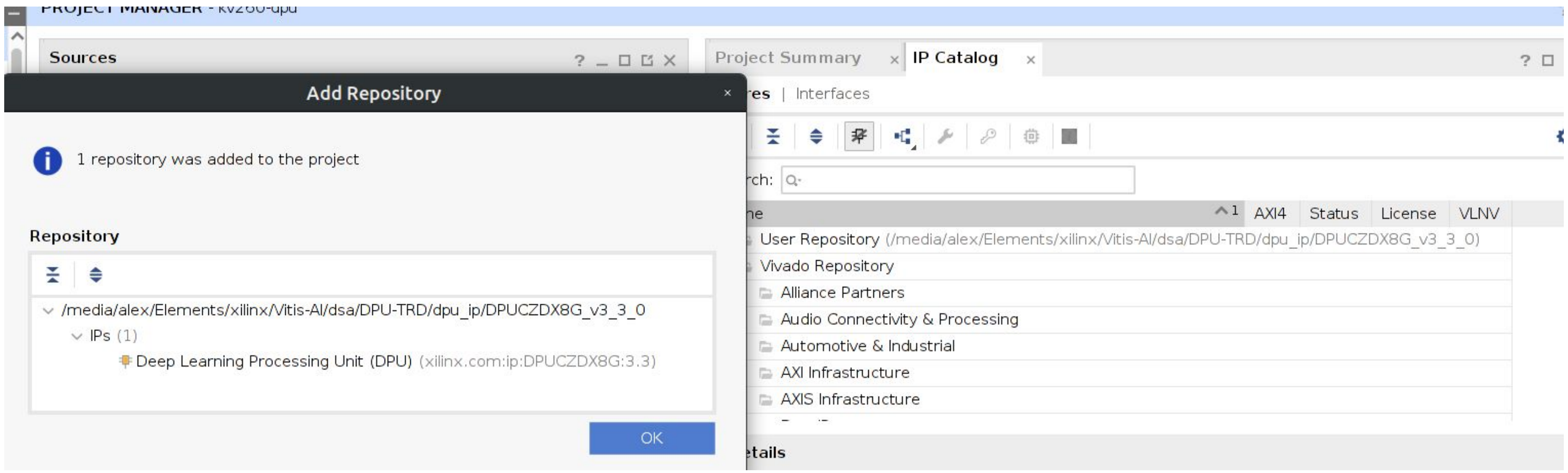


Fig. 2: The model structure of our proposed FADNet

Implementation

Add DPU IP from VITIS-AI DPU-TRD repository to the Vivado design project



1 repository was added to the project

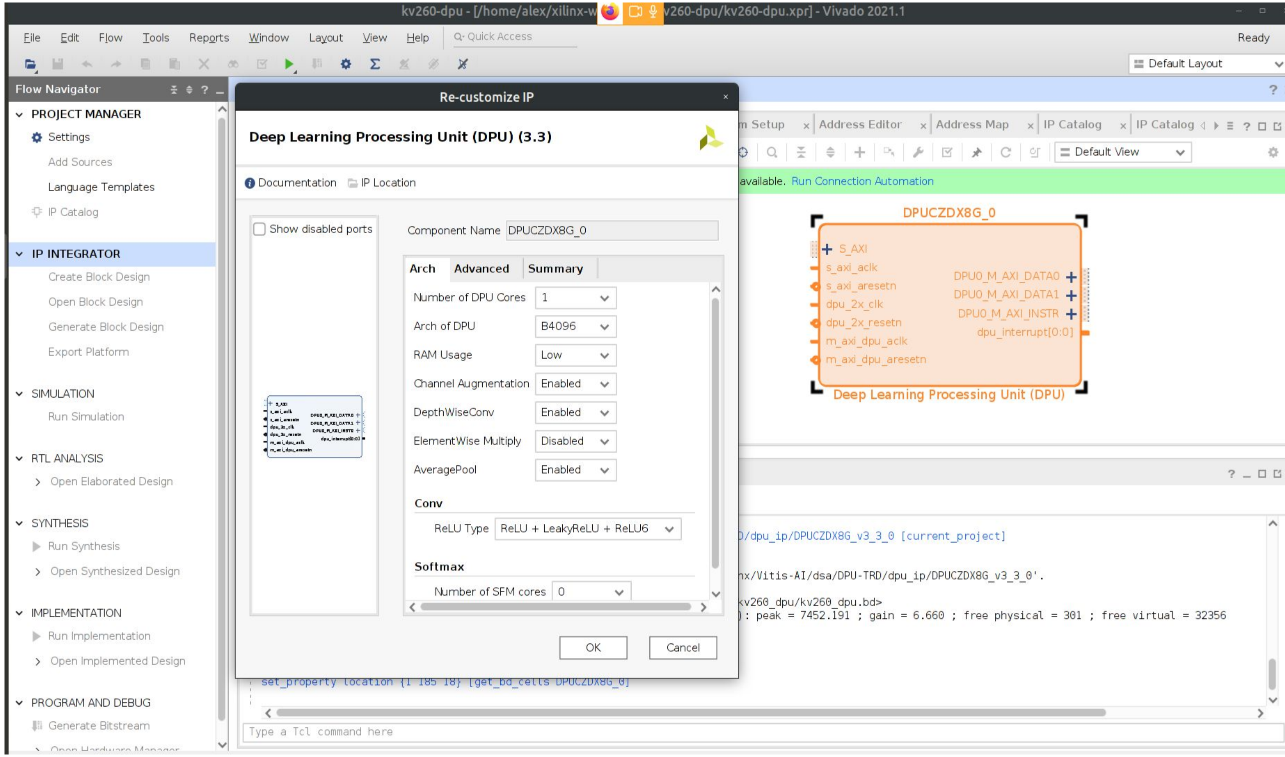
Repository

/media/alex/Elements/xilinx/Vitis-AI/dsa/DPU-TRD/dpu_ip/DPUCZDX8G_v3_3_0
IPs (1)
Deep Learning Processing Unit (DPU) (xilinx.com:ip:DPUCZDX8G:3.3)

User Repository (/media/alex/Elements/xilinx/Vitis-AI/dsa/DPU-TRD/dpu_ip/DPUCZDX8G_v3_3_0)
Vivado Repository
Alliance Partners
Audio Connectivity & Processing
Automotive & Industrial
AXI Infrastructure
AXIS Infrastructure

Implementation

The DPU IP is encrypted so there is no way to look at its intellectual property design. However we can configure it in the design by GUI



Deep Learning Processing Unit (DPU) (3.3)

Component Name: DPUCZDX8G_0

Number of DPU Cores	1
Arch of DPU	B4096
RAM Usage	Low
Channel Augmentation	Enabled
DepthWiseConv	Enabled
ElementWise Multiply	Disabled
AveragePool	Enabled

Conv

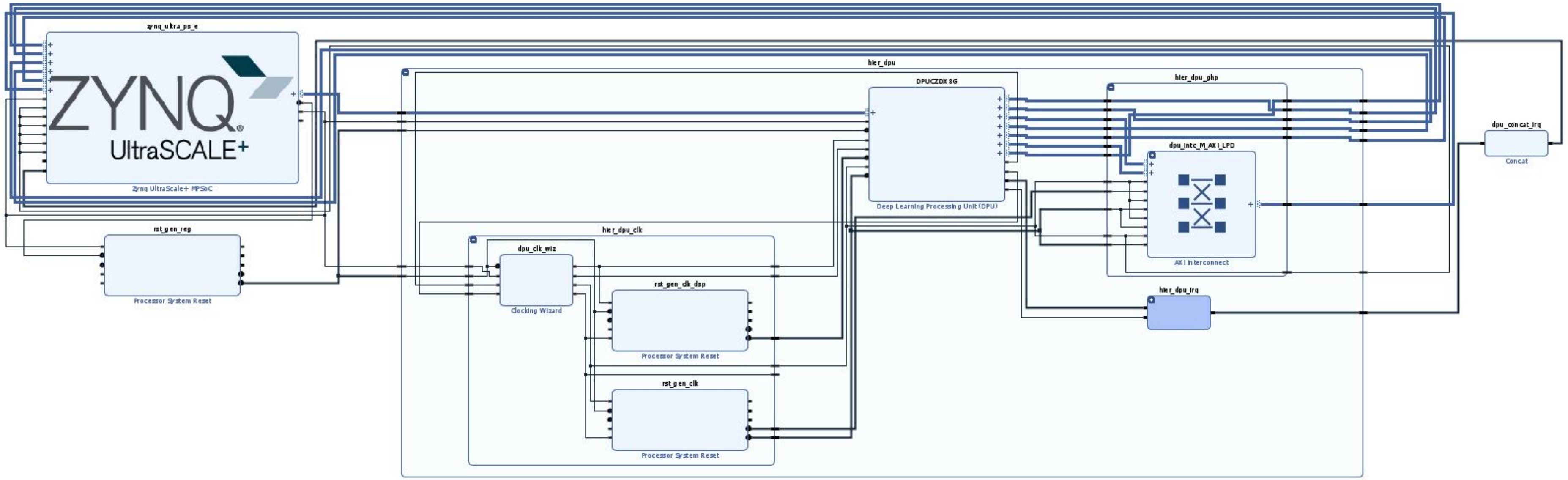
ReLU Type	ReLU + LeakyReLU + ReLU6

Softmax

Number of SFM cores	0

```
D/dpu_ip/DPUCZDX8G_v3_3_0 [current_project]

nx/Vitis-AI/dsa/DPU-TRD/dpu_ip/DPUCZDX8G_v3_3_0'.

kv260_dpu/kv260_dpu.bd>
): peak = 7452.191 ; gain = 6.660 ; free physical = 301 ; free virtual = 32356
set_property location {1 185 18} [get_bd_cells DPUCZDX8G_0]
```

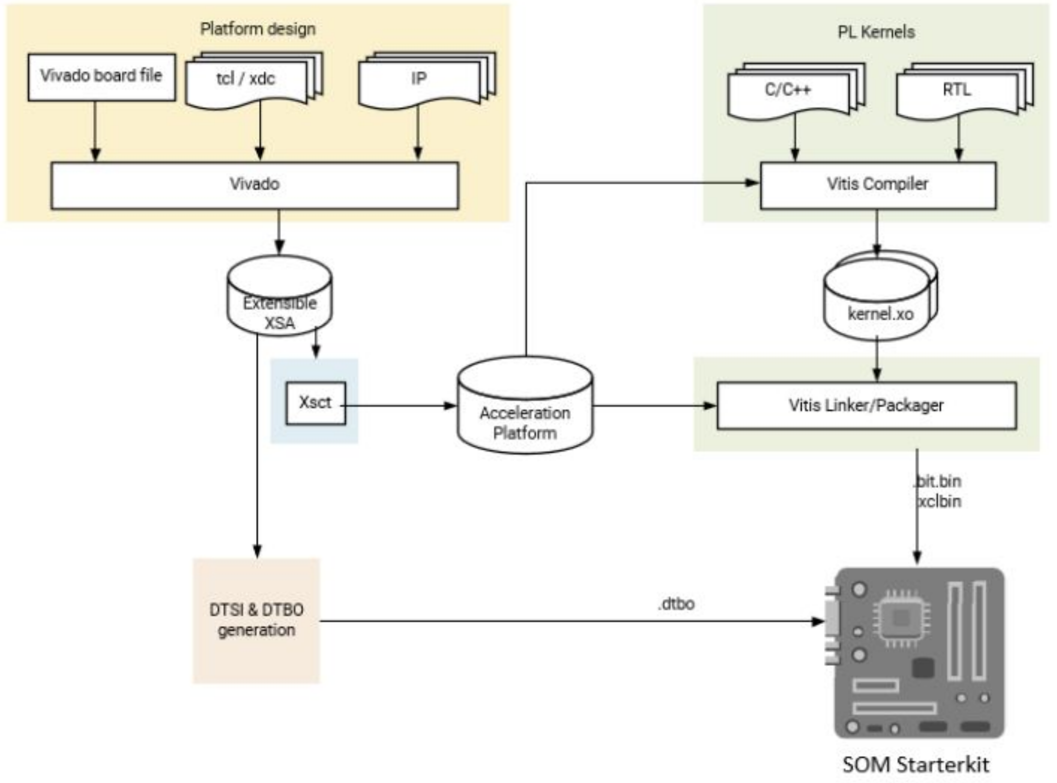
Implementation



Implementation

Synthesize the design and generate a bitstream file, which will be altogether saved on a Xilinx Support Archive or .xsa extension file.

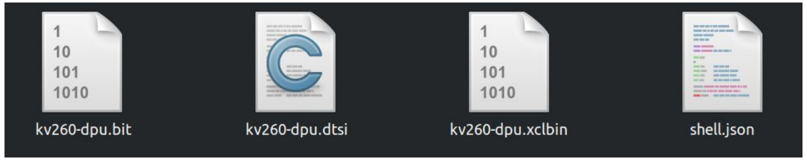
The Vitis platform generator will create a .xclbin binary. Unlike a common bitstream file, the xclbin file is designed so that the whole customized platform can be loaded during run-time without requiring to reboot the whole system.

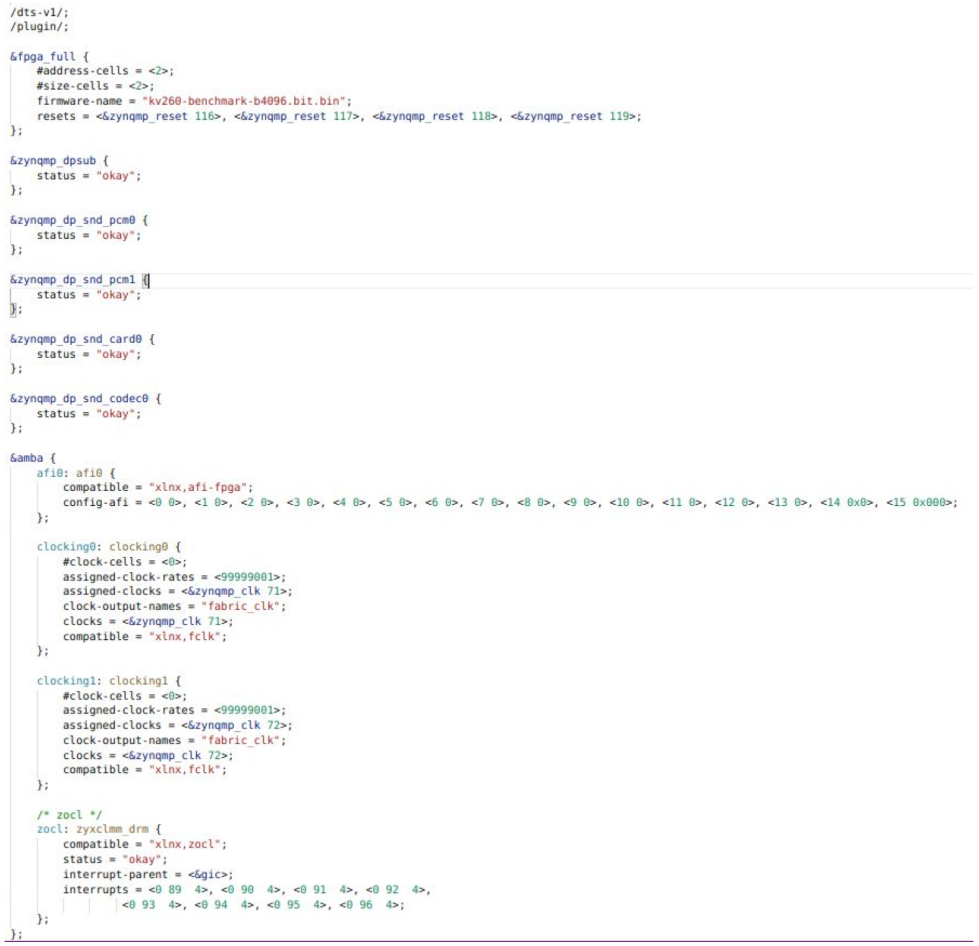


Implementation

Build the petalinux project with the Board Support Package of the kv260 board. This project will compile our hardware platform files

Device tree overlays are special device tree blob fragments that allow you to override specific parts of a device tree on-the-fly, before booting the operating system.

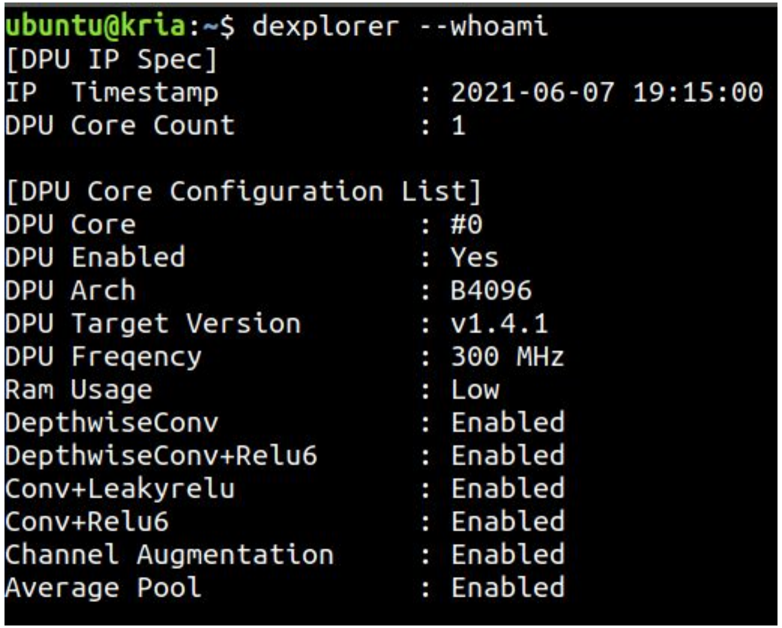




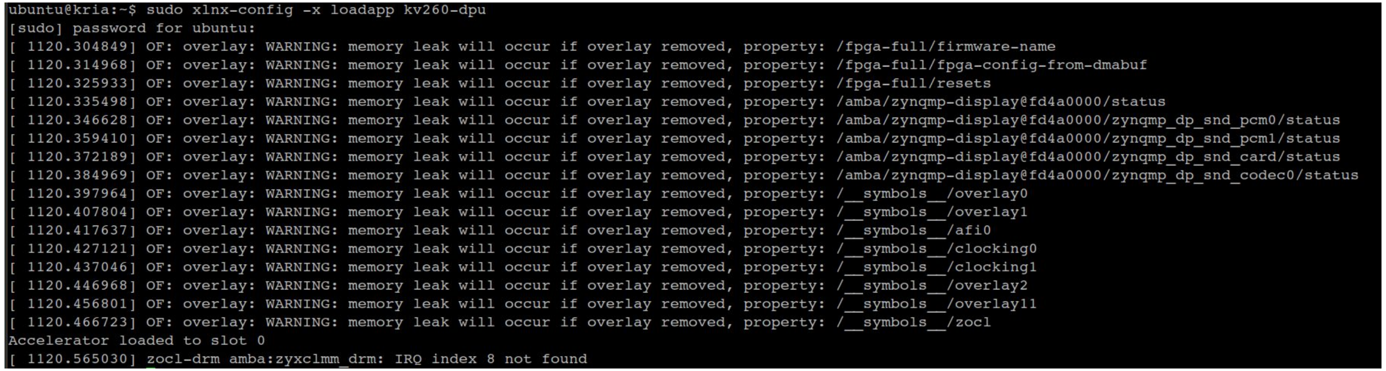
```
/dts-v1/;
/plugin/;

&fpga_full {
    #address-cells = <2>;
    #size-cells = <2>;
    firmware-name = "kv260-benchmark-b4096.bit.bin";
    resets = <&zynqmp_reset 116>, <&zynqmp_reset 117>, <&zynqmp_reset 118>, <&zynqmp_reset 119>;
};

&zynqmp_dpsub {
    status = "okay";
};

&zynqmp_dp_snd_pcm0 {
    status = "okay";
};

&zynqmp_dp_snd_pcm1 {
    status = "okay";
};

&zynqmp_dp_snd_card0 {
    status = "okay";
};

&zynqmp_dp_snd_codec0 {
    status = "okay";
};

&amba {
    afi0: afi0 {
        compatible = "xlnx,afi-fpga";
        config-afi = <0 0>, <1 0>, <2 0>, <3 0>, <4 0>, <5 0>, <6 0>, <7 0>, <8 0>, <9 0>, <10 0>, <11 0>, <12 0>, <13 0>, <14 0x0>, <15 0x000>;
    };

    clocking0: clocking0 {
        #clock-cells = <0>;
        assigned-clock-rates = <99999001>;
        assigned-clocks = <&zynqmp_clk 71>;
        clock-output-names = "fabric_clk";
        clocks = <&zynqmp_clk 71>;
        compatible = "xlnx,fclk";
    };

    clocking1: clocking1 {
        #clock-cells = <0>;
        assigned-clock-rates = <99999001>;
        assigned-clocks = <&zynqmp_clk 72>;
        clock-output-names = "fabric_clk";
        clocks = <&zynqmp_clk 72>;
        compatible = "xlnx,fclk";
    };

    /* zocl */
    zocl: zyxclmm_drm {
        compatible = "xlnx,zocl";
        status = "okay";
        interrupt-parent = <&gic>;
        interrupts = <0 89  4>, <0 90  4>, <0 91  4>, <0 92  4>,
                     <0 93  4>, <0 94  4>, <0 95  4>, <0 96  4>;
    };
};
```

Implementation

The DPUCZDX8G hardware platform must be loaded into the embedded linux for re-configuring the FPGA. After booting and logging into the MPSoC OS, we will use “xlnx-config” application manager. This program helps managing, installing and updating the FPGA firmware during runtime.



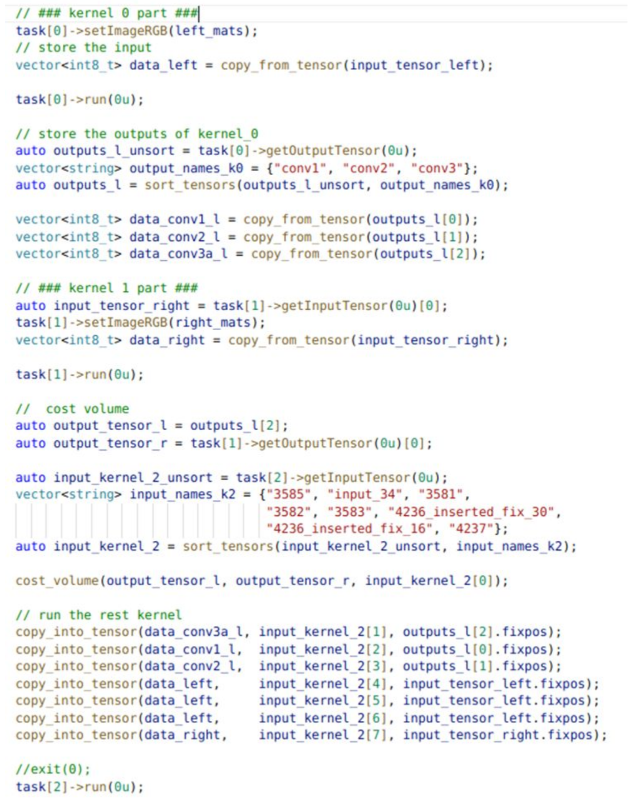
```
ubuntu@kria:~$ dexplorer --whoami
[DPU IP Spec]
IP  Timestamp                   : 2021-06-07 19:15:00
DPU Core Count                  : 1

[DPU Core Configuration List]
DPU Core                        : #0
DPU Enabled                     : Yes
DPU Arch                        : B4096
DPU Target Version              : v1.4.1
DPU Freqency                    : 300 MHz
Ram Usage                       : Low
DepthwiseConv                   : Enabled
DepthwiseConv+Relu6             : Enabled
Conv+Leakyrelu                  : Enabled
Conv+Relu6                      : Enabled
Channel Augmentation            : Enabled
Average Pool                    : Enabled
```



```
ubuntu@kria:~$ sudo xlnx-config -x loadapp kv260-dpu
[sudo] password for ubuntu:
[ 1120.304849] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
[ 1120.314968] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/fpga-config-from-dmabuf
[ 1120.325933] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
[ 1120.335498] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /amba/zynqmp-display@fd4a0000/status
[ 1120.346628] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /amba/zynqmp-display@fd4a0000/zynqmp_dp_snd_pcm0/status
[ 1120.359410] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /amba/zynqmp-display@fd4a0000/zynqmp_dp_snd_pcm1/status
[ 1120.372189] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /amba/zynqmp-display@fd4a0000/zynqmp_dp_snd_card/status
[ 1120.384969] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /amba/zynqmp-display@fd4a0000/zynqmp_dp_snd_codec0/status
[ 1120.397964] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay0
[ 1120.407804] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay1
[ 1120.417637] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afi0
[ 1120.427121] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
[ 1120.437046] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking1
[ 1120.446968] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay2
[ 1120.456801] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay11
[ 1120.466723] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/zocl
Accelerator loaded to slot 0
[ 1120.565030] zocl-drm amba:zyxclmm_drm: IRQ index 8 not found
```

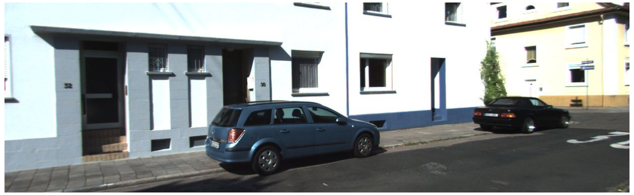
Implementation

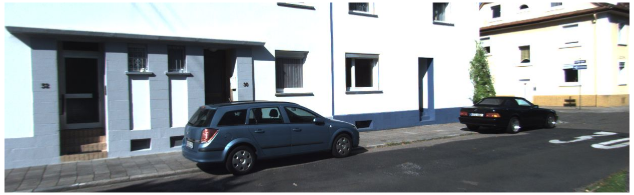
The FadNet processing consists of feeding the pre-processed image tensors across the different kernel tasks.

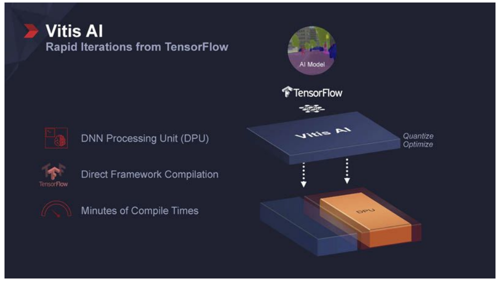


```cpp
// ### kernel 0 part ###
task[0]->setImageRGB(left_mats);
// store the input
vector<int8_t> data_left = copy_from_tensor(input_tensor_left);

task[0]->run(0u);

// store the outputs of kernel_0
auto outputs_l_unsort = task[0]->getOutputTensor(0u);
vector<string> output_names_k0 = {"conv1", "conv2", "conv3"};
auto outputs_l = sort_tensors(outputs_l_unsort, output_names_k0);

vector<int8_t> data_conv1_l = copy_from_tensor(outputs_l[0]);
vector<int8_t> data_conv2_l = copy_from_tensor(outputs_l[1]);
vector<int8_t> data_conv3a_l = copy_from_tensor(outputs_l[2]);

// ### kernel 1 part ###
auto input_tensor_right = task[1]->getInputTensor(0u)[0];
task[1]->setImageRGB(right_mats);
vector<int8_t> data_right = copy_from_tensor(input_tensor_right);

task[1]->run(0u);

//  cost volume
auto output_tensor_l = outputs_l[2];
auto output_tensor_r = task[1]->getOutputTensor(0u)[0];

auto input_kernel_2_unsort = task[2]->getInputTensor(0u);
vector<string> input_names_k2 = {"3585", "input_34", "3581",
                                 "3582", "3583", "4236_inserted_fix_30",
                                 "4236_inserted_fix_16", "4237"};
auto input_kernel_2 = sort_tensors(input_kernel_2_unsort, input_names_k2);

cost_volume(output_tensor_l, output_tensor_r, input_kernel_2[0]);

// run the rest kernel
copy_into_tensor(data_conv3a_l, input_kernel_2[1], outputs_l[2].fixpos);
copy_into_tensor(data_conv1_l,  input_kernel_2[2], outputs_l[0].fixpos);
copy_into_tensor(data_conv2_l,  input_kernel_2[3], outputs_l[1].fixpos);
copy_into_tensor(data_left,     input_kernel_2[4], input_tensor_left.fixpos);
copy_into_tensor(data_left,     input_kernel_2[5], input_tensor_left.fixpos);
copy_into_tensor(data_left,     input_kernel_2[6], input_tensor_left.fixpos);
copy_into_tensor(data_right,    input_kernel_2[7], input_tensor_right.fixpos);

//exit(0);
task[2]->run(0u);
```

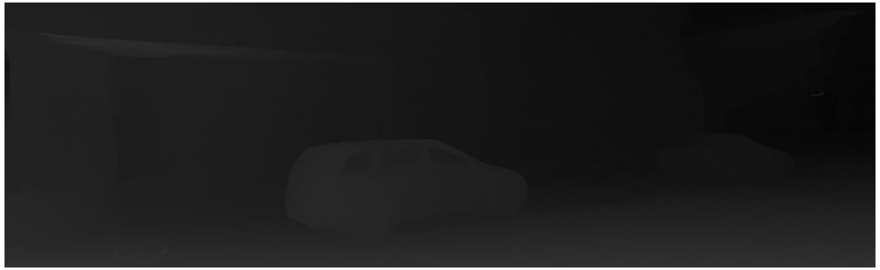
Results





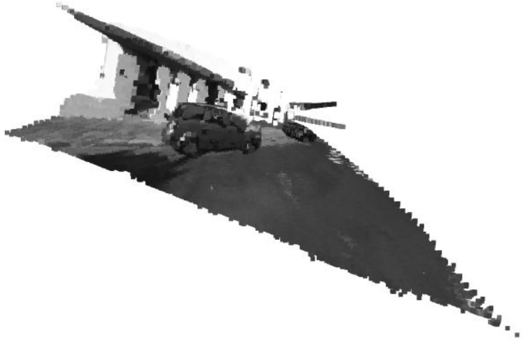


Vitis AI
Rapid Iterations from TensorFlow
AI Model
TensorFlow
Vitis AI
Quantize
Optimize
DNN Processing Unit (DPU)
Direct Framework Compilation
Minutes of Compile Times
DPU



Open3D

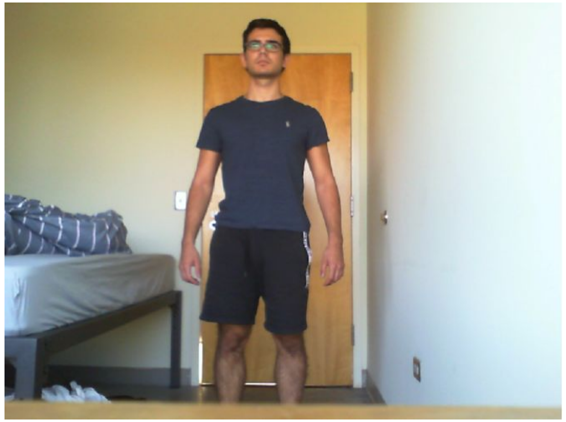
A Modern Library for 3D Data Processing

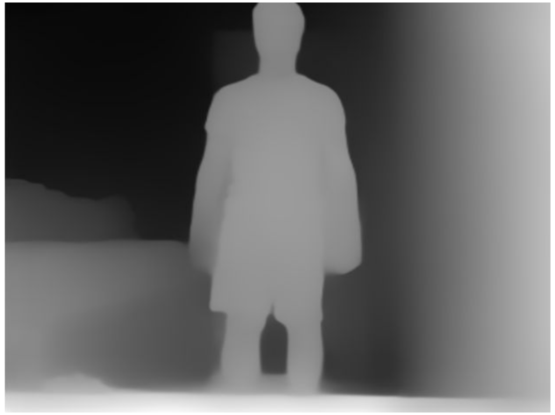


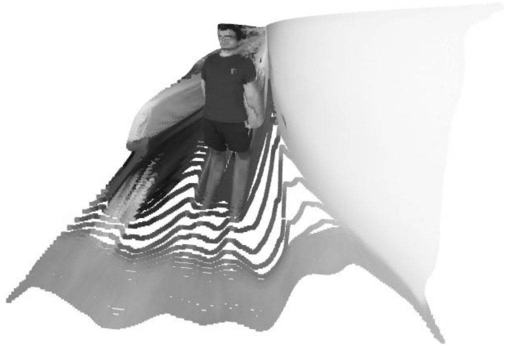
Paving Path Forward

- Create a video pipeline on FPGA to do real time processing on information retrieved
- Exploring Mobile Industry Processor Interface (MIPI) for embedded image cameras
- Triangulation, Monocular Depth, Multi-View reconstruction, segmentation algorithm on higher processing power FPGAs (ZCU104)

Currently working on


2D RGB Image


Depth map estimation


Point Cloud Mesh

Currently working on

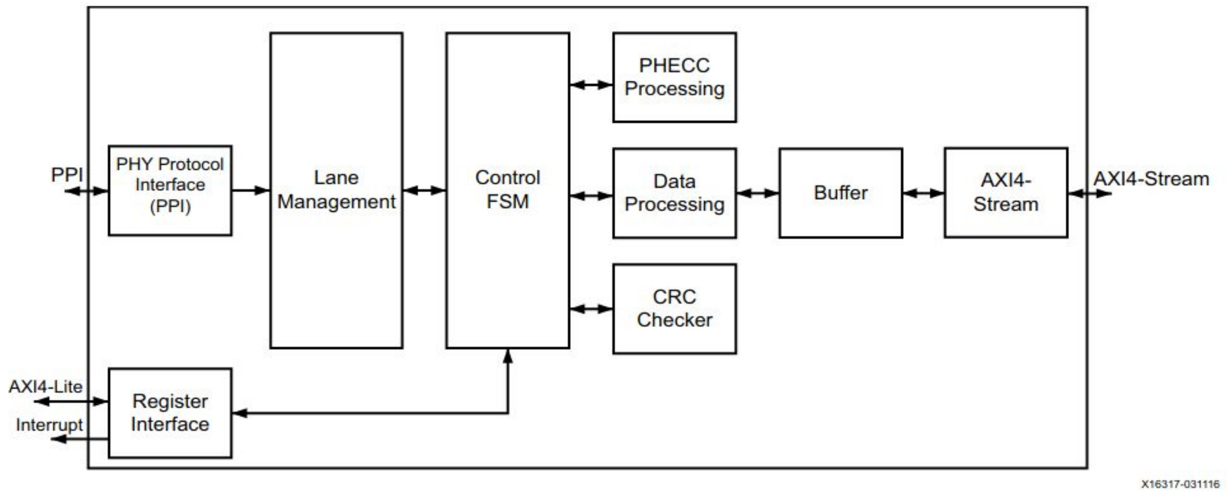


Figure 1-2: **MIPI CSI-2 RX Controller Core**

References

[1] https://www.xilinx.com/developer/products/vivado.html
[2] https://www.xilinx.com/html_docs/xilinx2019_2/vitis_doc/dpu_over.html
[3]https://docs.xilinx.com/r/en-US/ug1393-vitis-application-acceleration/Introduction-to-the-Vitis-Environment-for-Acceleration
[4] https://xilinx.github.io/kria-apps-docs/creating_applications/1.0/build/html/index.html
[5]https://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html
[6]https://www.xilinx.com/html_docs/vitis_ai/1_1/index.html
[7]http://www.cvlibs.net/datasets/kitti/
[8]https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/2057043969/Snaps+-+xlnx-config+Snap+for+Certified+Ubuntu+on+Xilinx+Devices
[9] https://www.xilinx.com/html_docs/xilinx2019_2/vitis_doc/compiling_model.html
[10] https://opencv.org
[11] http://www.open3d.org/