



2019 年 SEU-Xilinx 国际暑期学校团队项目设计文档

(Project Paper Submission Template)

作品名称	基于 DPU 的人脸识别与检测
组员姓名	组长：彭锋 组员：董金陇 闫宇航
房间号及桌号	房间号：717 桌号：20



第一部分

小组成员分工

(各成员任务分配)

	姓名	任务分配
组长	彭锋	Pynq 上实现 DPU 加速, 制作数据集, 用 caffe 框架进行训练
组员 1	董金陇	petalinux 中创建 linux 平台
组员 2	闫宇航	运用 DPU 进行网络算法加速及通过 SDK 构建机器学习应用程序

第二部分

设计概述 /Design Introduction

(请简要描述一下你的设计: 1. 功能说明; 2. 所用到的设备清单)

- 1、在 Ultra96 上实现基于 DPU 的实时人脸识别。

所用网络: Resnet

设备: Ultra96 板套件、USB 摄像头、显示屏

- 2、在 Pynq 上实现基于 DPU 的 YOLO2 算法

所用网络: yolo2

设备: Pynq 开发板套件, OV5640 coms 摄像头

3. 基于自身数据集的 Caffe.model 生成

所用框架: 基于 Linux 的 caffe 环境

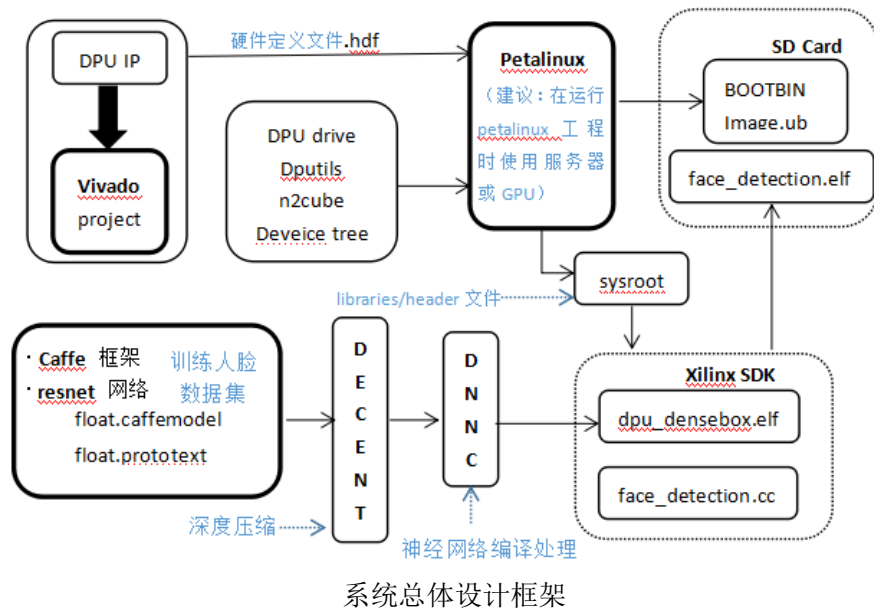
设备: GPU

第三部分

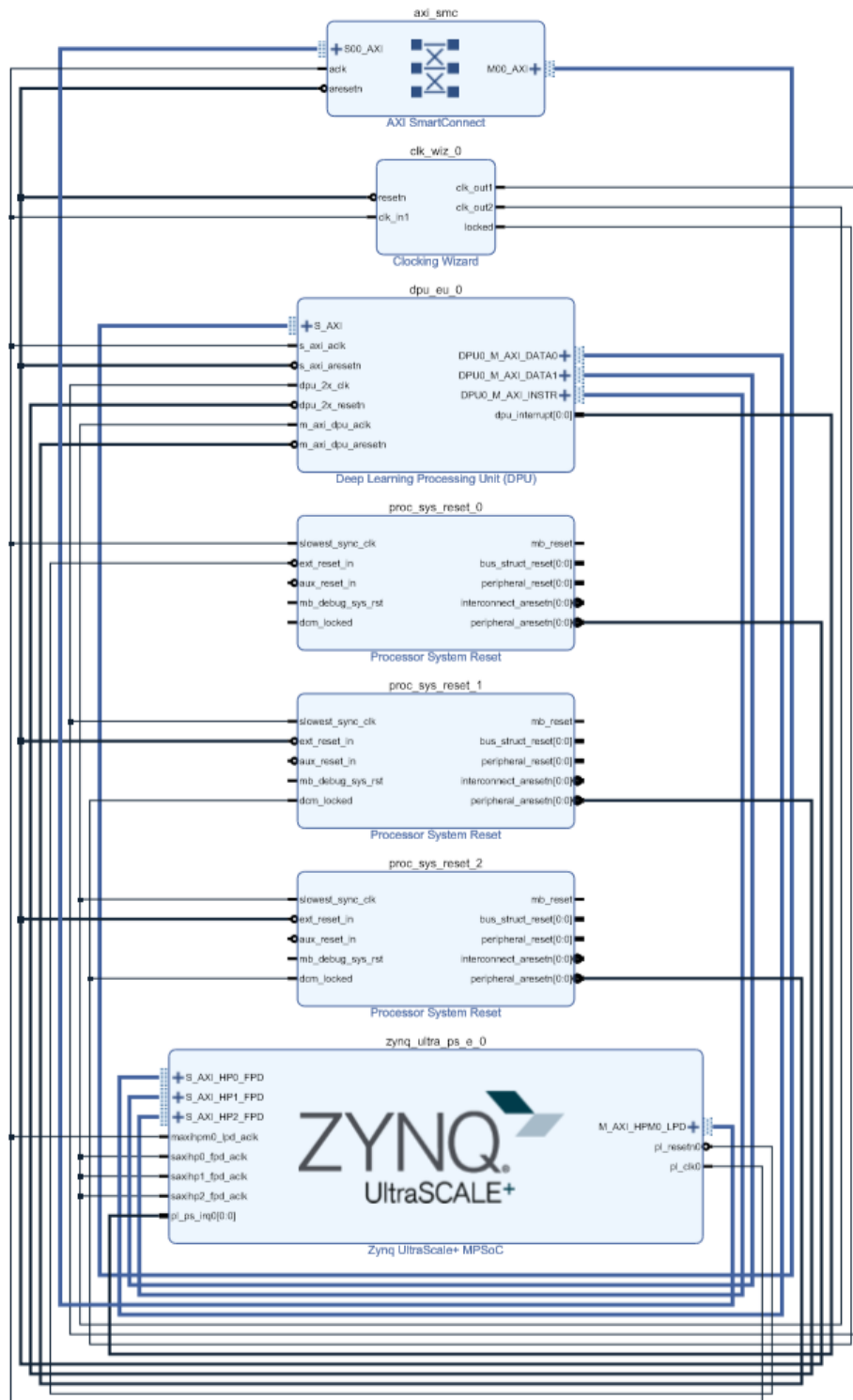
详细设计 /Detailed Design

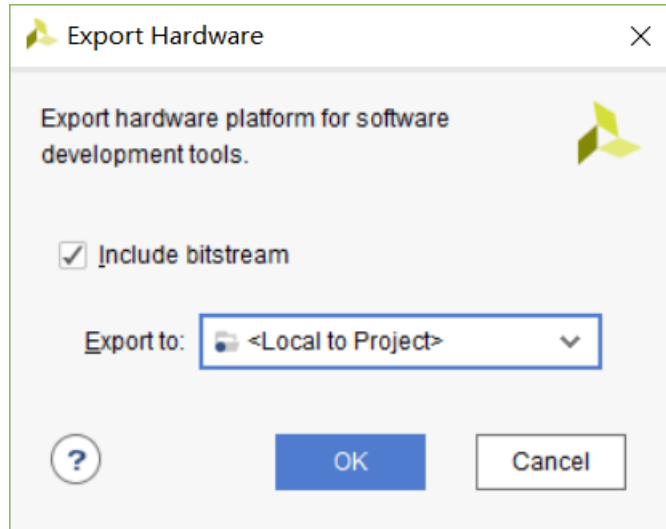
DPU 应用部署

1. 模型压缩 (主机)
2. 模型编译 (主机)
3. DPU 编程开发 (主机或开发板)
4. 混合编译 (开发板)
5. 运行 (开发板)



将 DPU 的 IP 添加进一个 vivado 工程中，将所需 IP 进行连接，将 block_design 生成 HDL_wrapper,进一步生成 bitstream 并且输出 .hdf 文件。

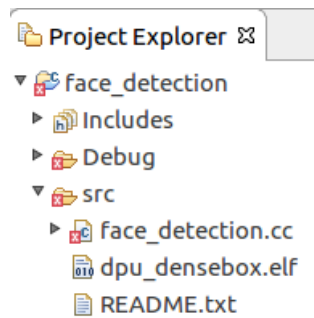




在 petalinux 中生成所需文件及目录。

构建一个应用，将使用 DNNDK 优化的训练模型生成的.elf 文件与.cc 文件共同输入到 SDK 中，添加环境变量到由 petalinux 生成的 sysroot 目录。

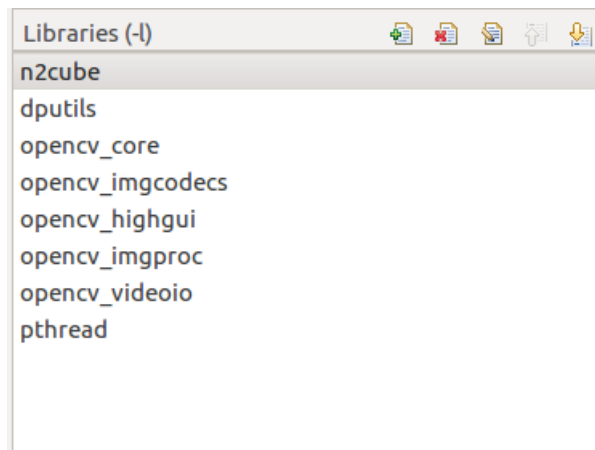
以脸部识别的 demo 为例，首先将 face_detection.cc 和 dpu_densebox.elf（由 dnnc 生成）import 到指定的路径。添加后的文件结构如图所示。



接下来设置系统环境变量到由 petalinux 生成的“aarch64-xilinx-linux”文件夹。

Variable	Value	Origin
CWD	/home/xilinx/Downloads/dpu_integration_lab/sc	BUILD SYSTEM
PWD	/home/xilinx/Downloads/dpu_integration_lab/sc	BUILD SYSTEM
SYSROOT	/home/xilinx/Downloads/aarch64-xilinx-linux	USER: CONFIG

设置 g++连接器和编译器指向 SYSROOT，并在 g++linker 的 libraries 中添加如下 library:

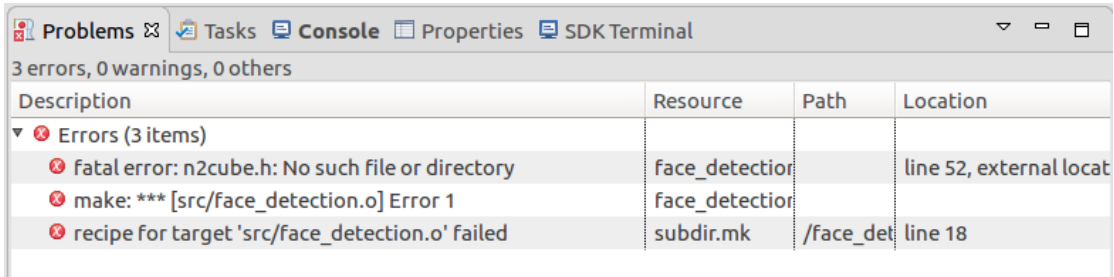


将 linker 的 Other Object 设置为



/home/xilinx/Downloads/dpu_integration_lab/files/face_detection/B1152_1.3.0/dpu_densebox.elf

此时 build project 后会报错，教程中并未提及。

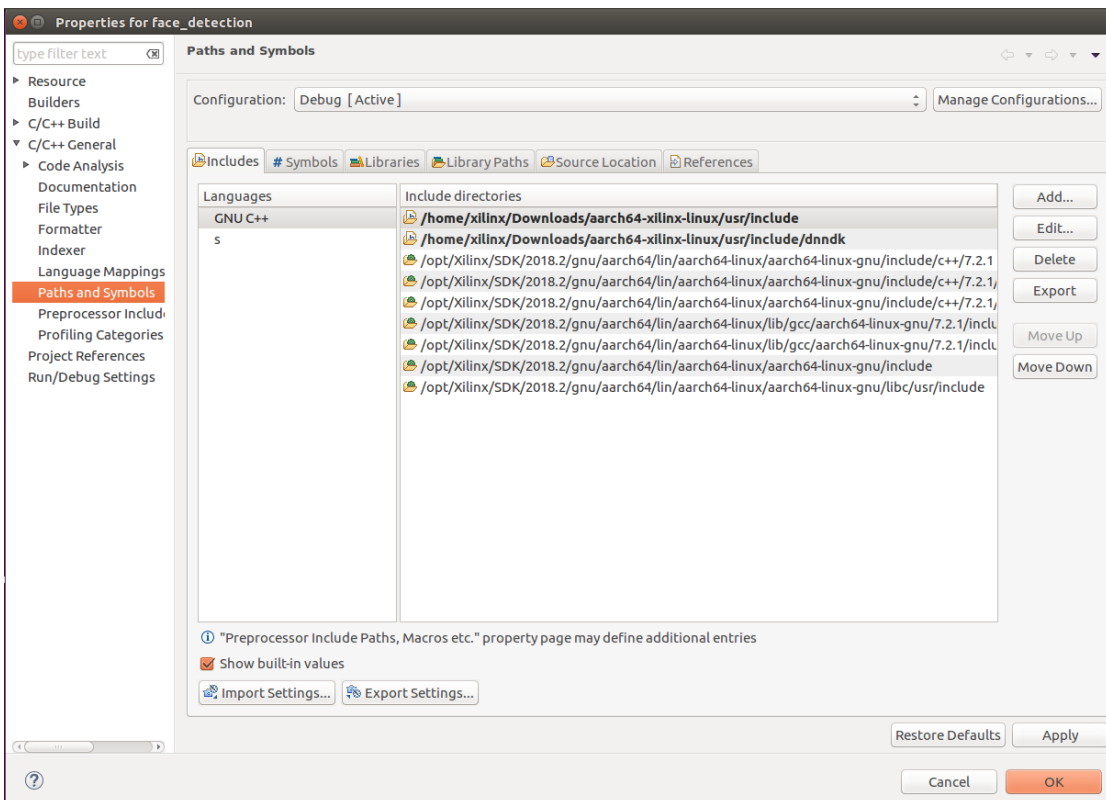


SDK 无法找到 n2cube.h。另外在 face_detection.cc 中，在 include 如下头文件时出现了问题。

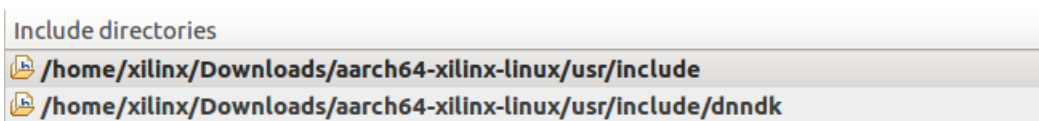
```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/opencv.hpp>

// Header files for DNNDK API
#include <dnndk/dnndk.h>
```

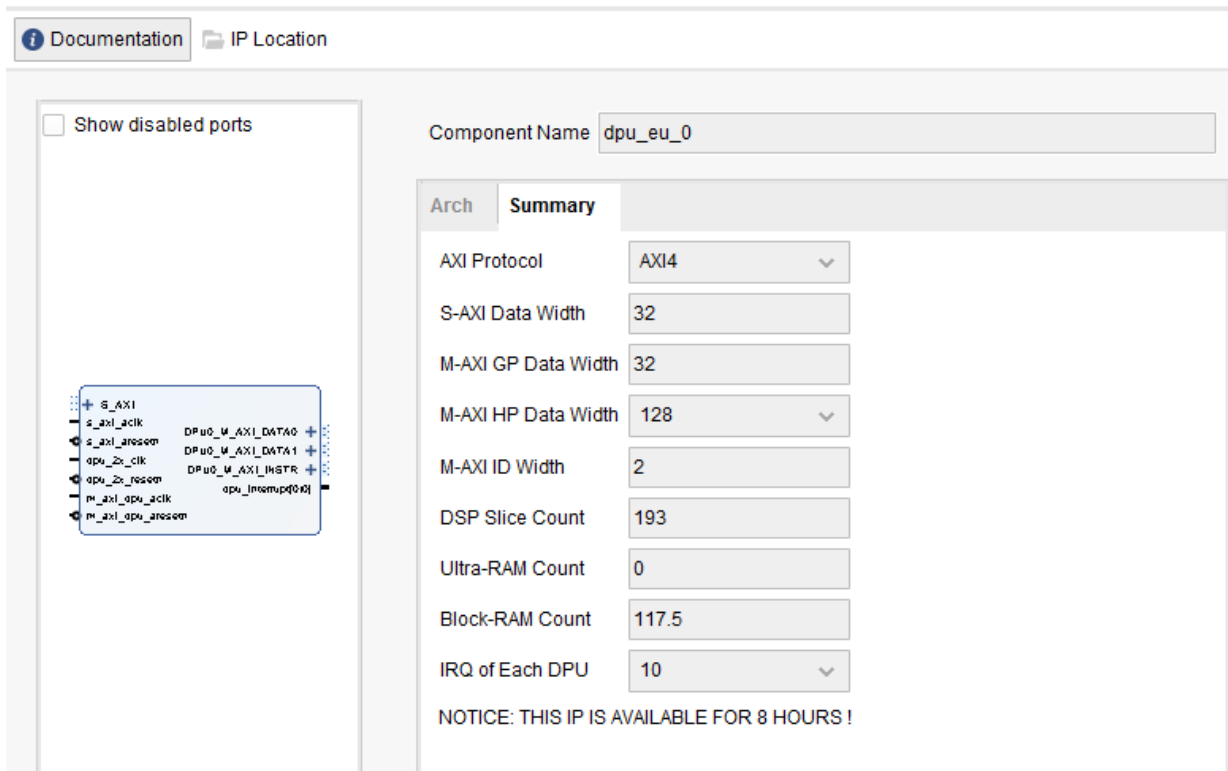
基于此，设置“c/c++ General”中的“Path and Symbols”如图所示。



具体为添加如下两个路径：



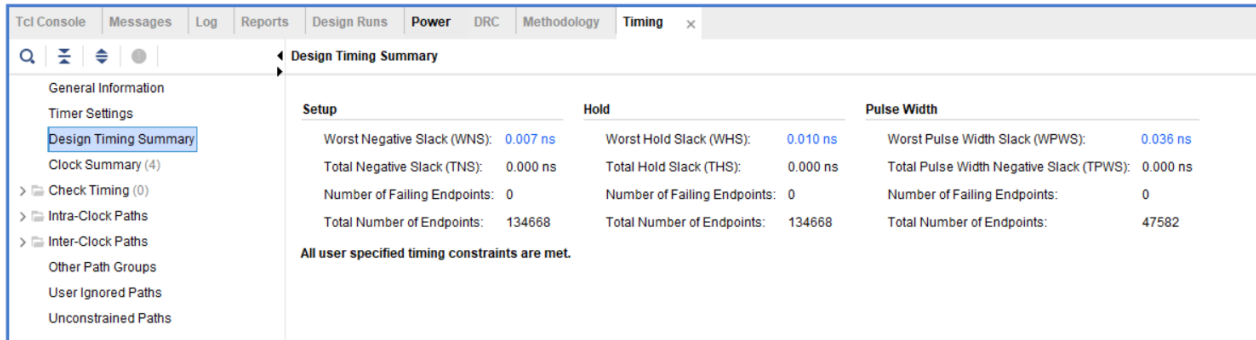
便可消除错误。此时 build project 便可生成 sd 卡中所需的 face_detection.elf 文件。



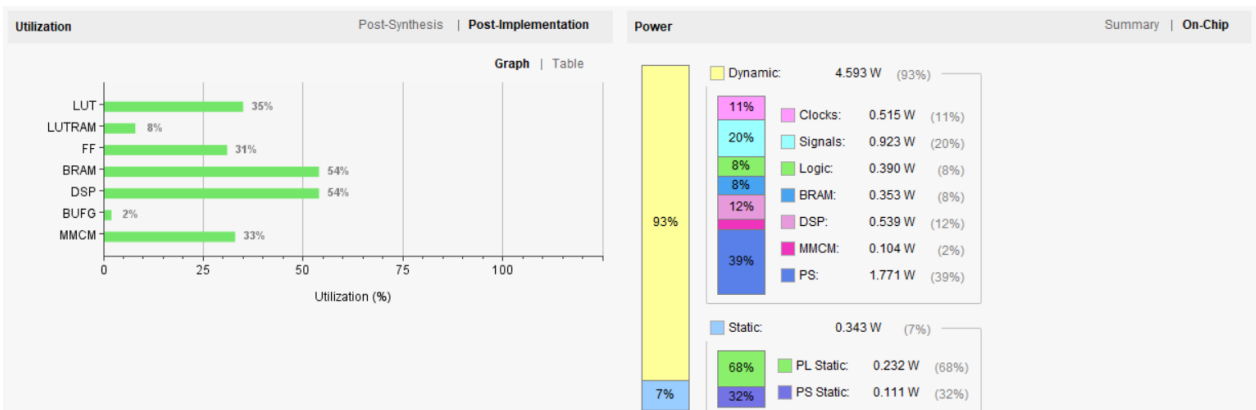


After Running the performance parameters:

1. Timing Summary



2. resource utilization



第五部分

项目总结 /Conclusions

(项目中所用到的知识点, 项目收获及心得)

知识点:

1. xilinx 开发软件的操作

1. Vivado
2. SDK
3. PetaLinux
4. Ubuntu

2. 底层框架的搭建调试

1. vivado IP 调用和参数设置
2. 在 Ubuntu 中下载安装Petalinux, 并调试环境
3. 创建Petalinux 工程, 利用服务器编译

3. 熟悉 DPU 工作模块的功能和流程

1. DEep ComprESSION Tool (DECENT), 深度压缩工具: 提供剪枝和定点化运算。

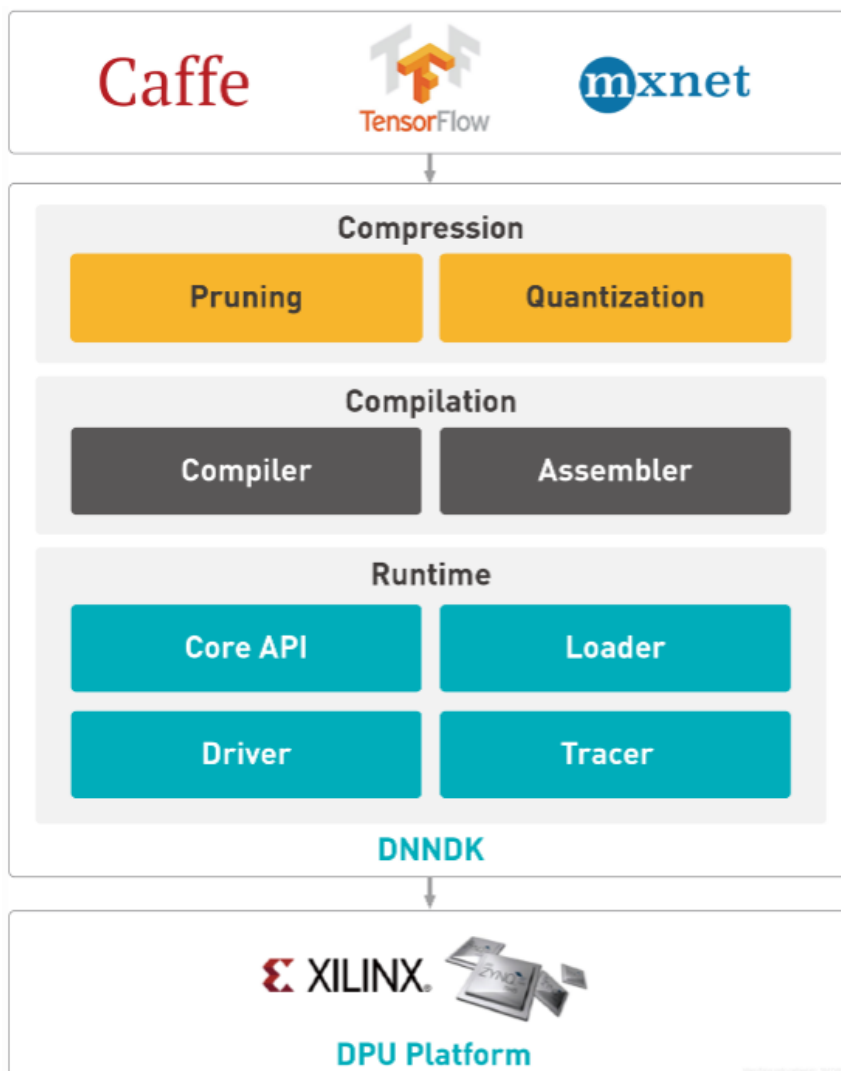
Deep Neural Network Compiler (DNNC), 神经网络编译器: 将网络算法编译到 DPU 平台高效运行。运用 DECENT 后的 caffe 网络模型作为输入。

2. Deep Neural Network Assembler (DNNAS), 神经网络集成器: 将 DPU 指令流会编程标准的 ELF 二进制文件



Neural Network Runtime (N2Cube), 神经网络运行, 运行时支持环境, 神经网络的加载, 资源管理, 调度

3. N2Cube 核心组件包括四个部分 : DPU 驱动程序 (Driver), 加载器 (Loader)、性能分析器 (Profiler) 及编程开发库编程开发库 (Library)。



4. Caffe 框架的使用

第六部分

源代码/Source code

详见 GitHub: <https://github.com/FengPenggo/Create-caffe.model-on-caffe->

第七部分 (技术总结/心得/笔记等分享)

心得: 在 7 天的学习中, 我们组成员对于 FPGA 在软件层面以及 HLS 等方面有了入门式的了解. 过去一直使用 RTL 去做底层的搭建, 现在学会利用嵌入式和 C 语言去完成项目工作, 这扩充了我们的学习方向, 同时 HLS 的强大, 也使得很多工作有了更多的进展. 我们本身都是硕士, 在学习过程中遇到过很多问题, 一直在遇到问题解决问题的路上. 这次课程的培训给了我们非常多的指导, 尤其在和几位导师探讨对于学术方面的见解, 有了很大启发.



技术总结:因为这次我们选择的项目是 DPU,这对于我们来说是一个全新的领域,包括像是 petalinux 的操作,IP 核与底层文件的混合编译.都是第一次接触,所以问题不断,甚至环境搭建也在前期遇到很多问题.在解决问题,实现目标后,这次的课程不仅仅给了我们学习上的指导,更加开阔了我们的视野,不仅仅局限于 RTL 层次的实现,更多的与未来,与深度学习接轨.