哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

深度学习体系结构（实验）

课程实验目标

- 加深对DLA相关理论知识的理解
- 掌握**domain-specific加速器设计**的基本原理、基本方法
- 具备**软硬件协同**的系统分析、系统设计能力

实验概况

- 实验成绩：共**40**分
  - 验收：**19**分
  - 报告：**21**分

	项目名称	学时	验收	报告	总分
实验一	神经网络加速器设计入门	2	2	2	4
实验二	YOLO算法量化加速	4	4	4	8
实验三	基于LSTM的MNIST手写数字识别加速	4	5	7	12
实验四	基于脉动阵列的CNN加速器设计	6	7	9	16

  - **附加题**：**每道题额外加相应分数，加满40分为止**
- 指导书网址：https://hitsz-cslab.gitee.io/dla/

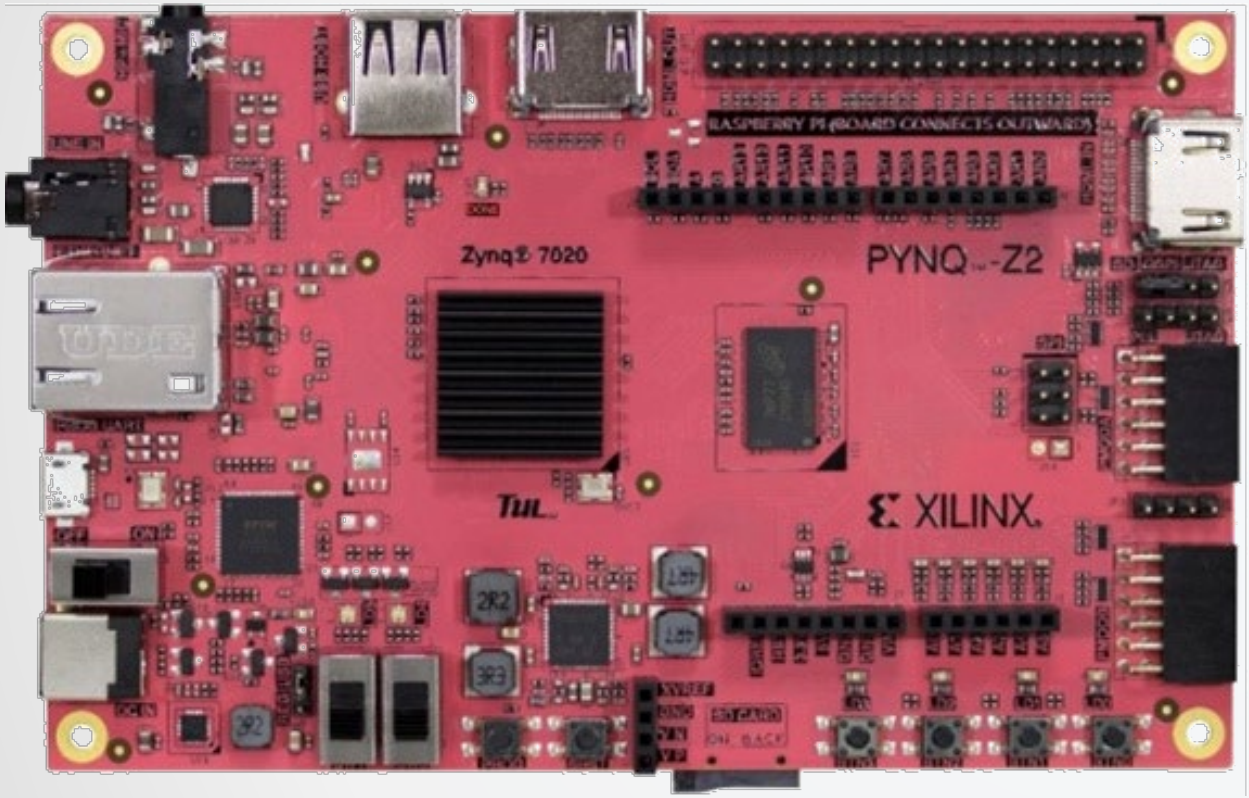
哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

PYNQ实验平台简介

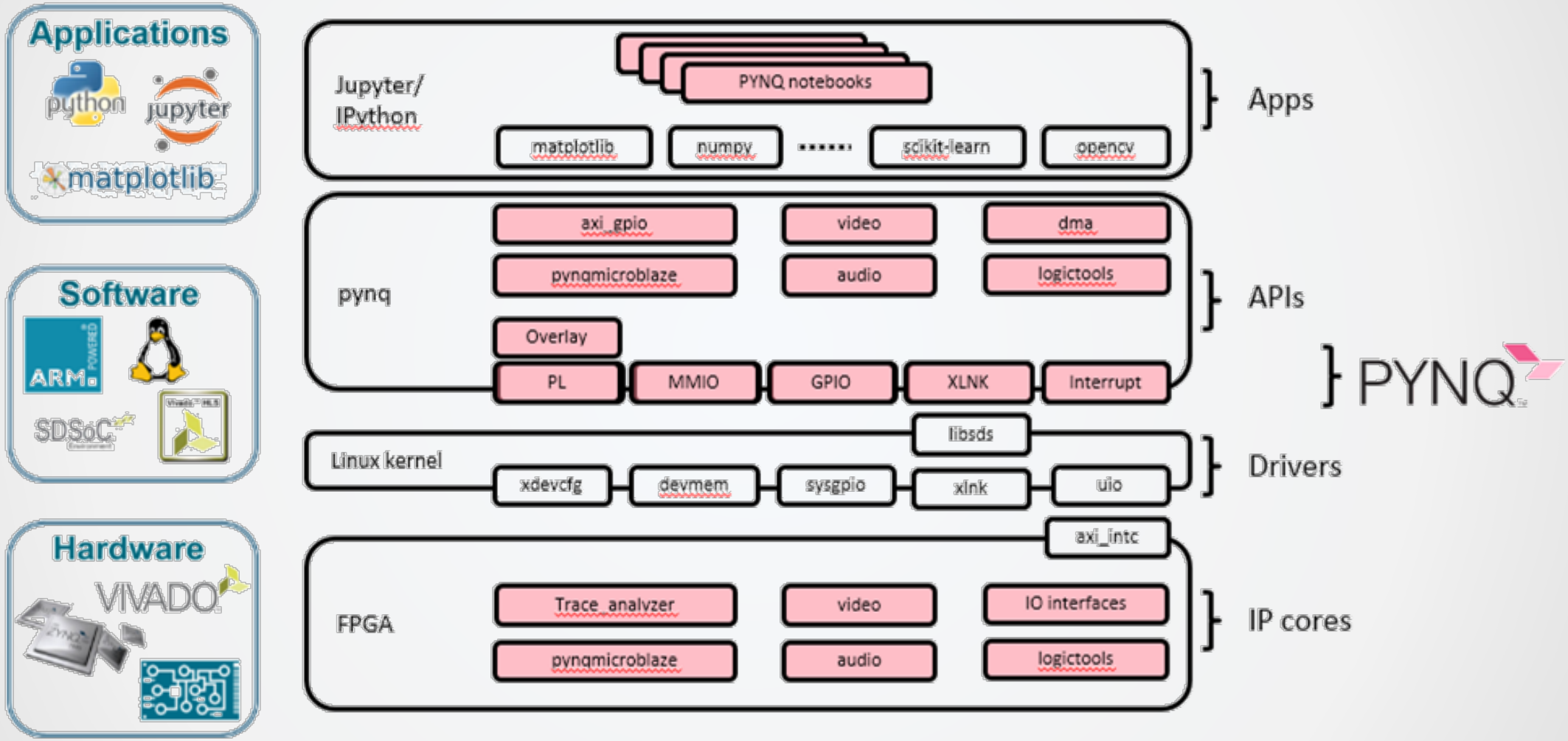
PYNQ-Z2开发板

- 官网：www.pynq.io



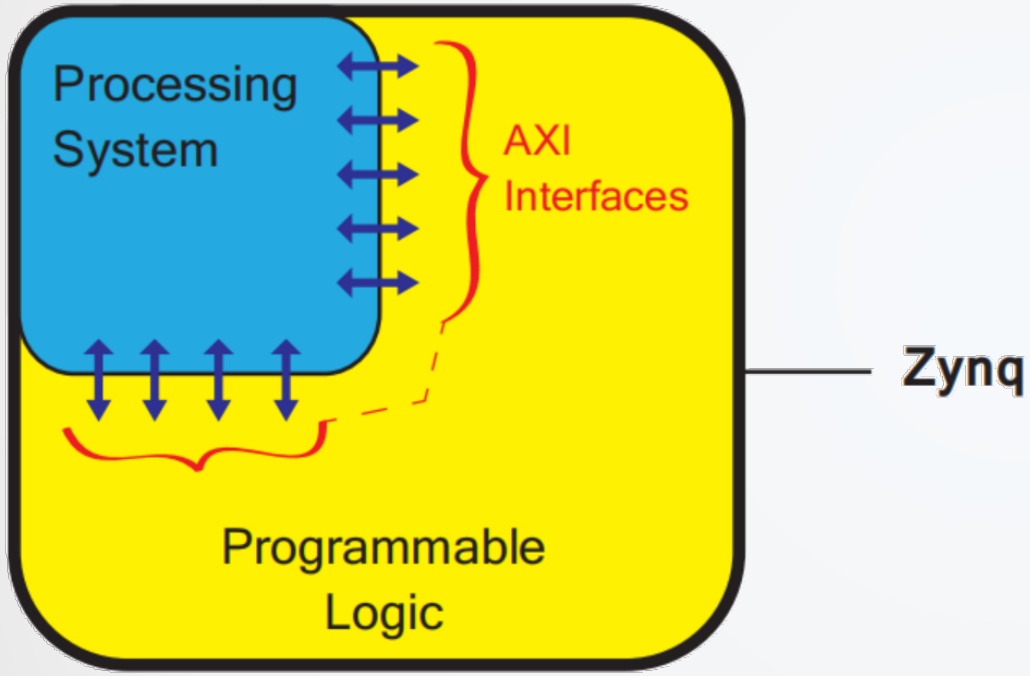
- ZYNQ 7020 SoC，xc7z020clg400-1
- 双核Cortex-A9 @ 667MHz
- 1.3M可编程逻辑门
- 256KB OCM，512MB DDR3内存
- HDMI，USB，Ethernet，Pmod，UART
- 兼容树莓派接口
- ……

PYNQ系统层次结构


Applications
python
jupyter
matplotlib
Software
ARM
SDSoC
Hardware
VIVADO
Jupyter/
IPython
PYNQ notebooks
matplotlib
numpy
scikit-learn
opencv
Apps
pynq
axi_gpio
video
dma
pynqmicroblaze
audio
logictools
Overlay
PL
MMIO
GPIO
XLNK
Interrupt
APIs
PYNQ
Linux kernel
libsds
xdevcfg
devmem
sysgpio
xlnk
uio
Drivers
FPGA
axi_intc
Trace_analyzer
video
IO interfaces
pynqmicroblaze
audio
logictools
IP cores

PYNQ架构

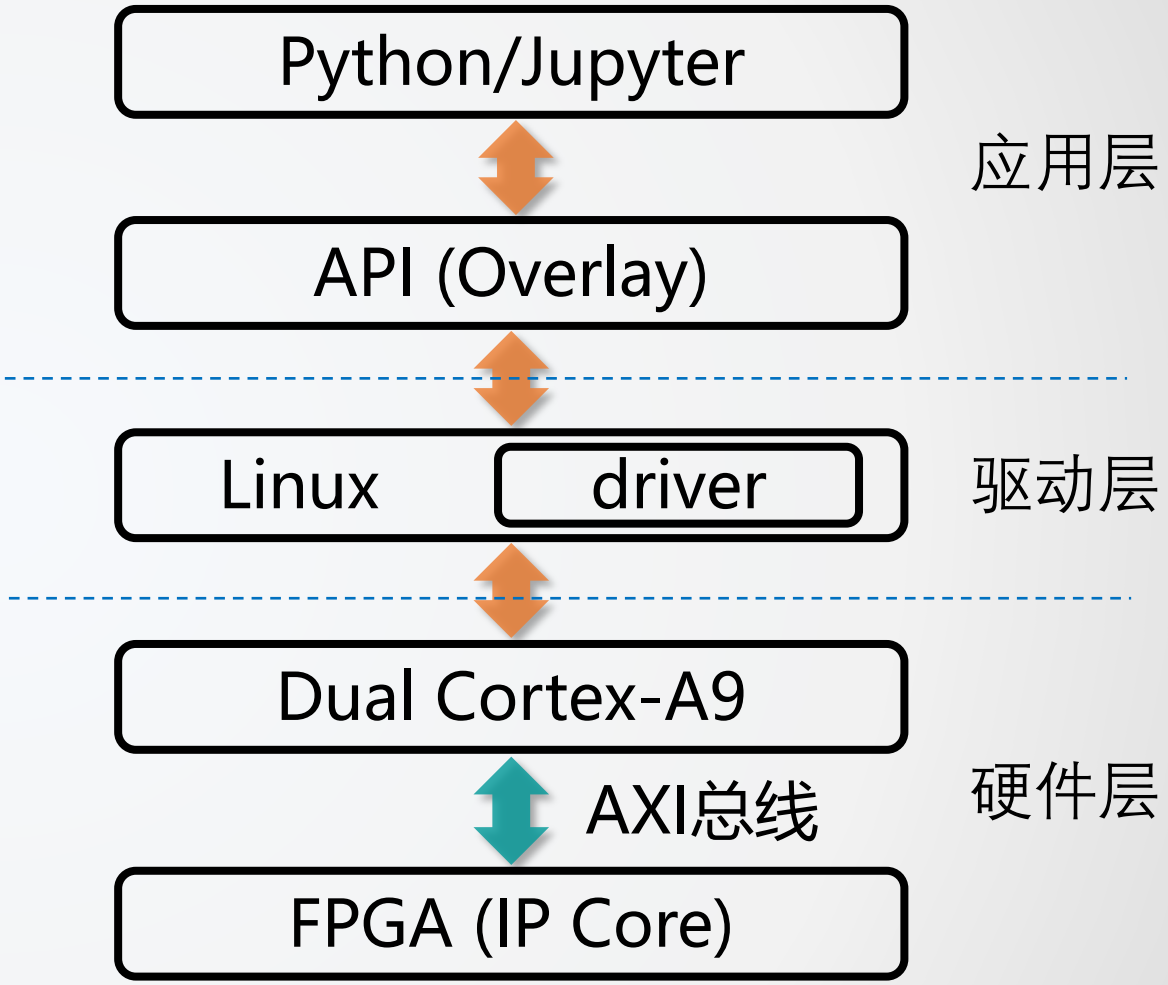
- 硬件层：ZYNQ = PS + PL



- PS（Processing System）
  - CPU、Cache、总线控制器、IO等
- PL（Programmable Logic）
  - FPGA、总线接口、编解码器
- PS与PL通过AXI总线交互

PYNQ系统层次结构

- 硬件层
  - PS和PL协同进行数据处理、运算
- 驱动层
  - 为OS访问底层硬件提供统一标准接口
  - 为应用提供API
- 应用层
  - 通过API完成相应的计算任务



哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

实验一 神经网络加速器设计入门

实验目的

- 了解使用硬件加速神经网络前向推导的基本**原理**，以及硬件加速器设计的基本**方法**
- 掌握利用HLS设计硬件IP核的基本**流程**、**方法**
- 了解ZYNQ异构平台的基本架构，熟悉PYNQ的**系统架构**及**使用方法**

实验内容

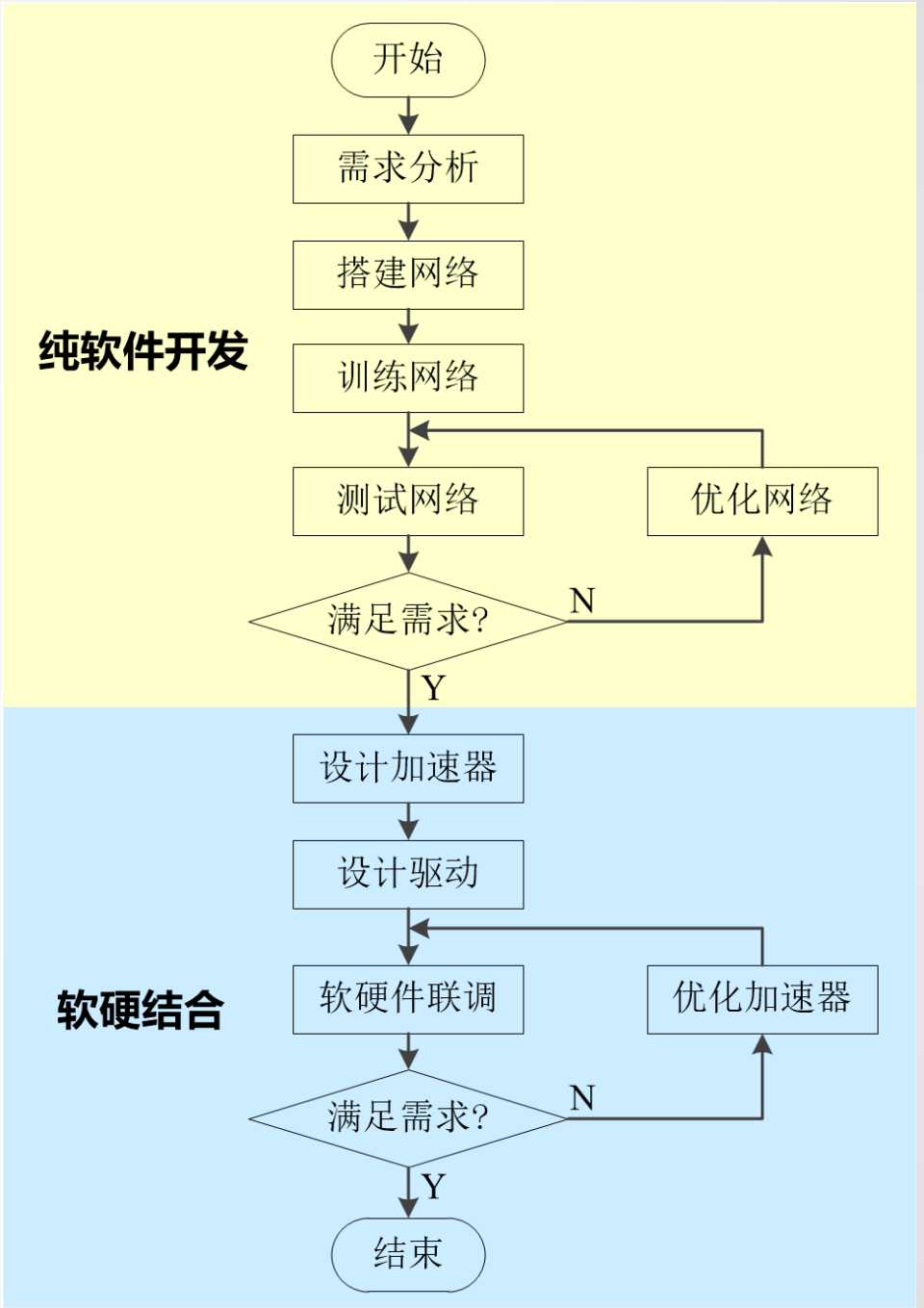
题目：CNN加速器设计

- 使用HLS设计实现卷积IP核
- 构建Block Design，生成比特流，导出Overlay
- 运行测试程序，并阅读代码，分析软件如何调用硬件IP核

实验原理 — 神经网络加速

神经网络（Neural Network，NN）

- 从设计到应用的3个阶段：
  - 搭建、训练、部署（推导）
- NN加速：
  - 通常指的是在**部署阶段**加速NN的前向推导
- 加速器的完整设计流程：
  - 包括**NN设计**和**加速器设计**2个阶段
  - 实验主要针对加速器设计阶段



实验原理 — 加速器设计方法

- 完全硬件化
  - 使用硬件实现整个NN
  - 性能高，但设计复杂、开发难度大、硬件资源消耗大、通用性差
- 部分硬件化
  - 使用硬件实现NN主要的操作
  - 开发难度低、硬件资源消耗少、灵活性较高，具有一定通用性

2种方法各有优劣，需根据实际情况灵活选择。

实验原理 — HLS

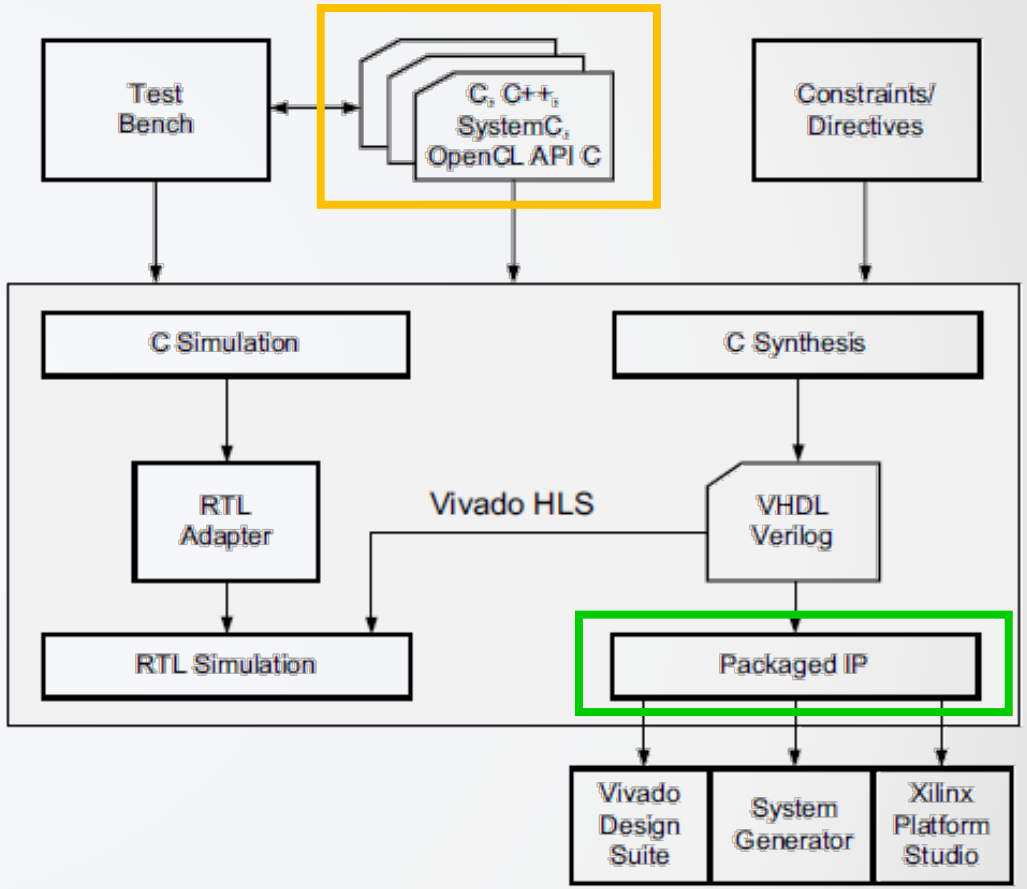
HLS（High-Level Synthesis，高层次综合）

- 传统FPGA开发使用HDL，开发周期长、难度较大
- 基于HLS开发：
  - 高级语言代码 -> 硬件IP核，快速生成算法的硬件加速器
    - 支持C/C++、System C、OpenCL等
  - 对软件开发人员友好，降低FPGA开发门槛，缩短开发周期
  - 支持软件定义硬件接口（常用AXI），具有良好的灵活性

实验原理 — HLS

HLS设计流程

1. 使用高级语言编写并调试算法
2. 将算法综合成RTL实现
   - 支持#pragma优化、综合分析报告
3. 验证RTL，打包生成IP核
4. 构建Block Design电路原理图
5. 生成Overlay，上板测试
   - Overlay = .tcl + .bit



实验原理 — HLS

HLS设计规范

1. 不使用动态内存分配（malloc()、free()、new、delete）
2. 不使用系统调用（abort()、exit()、printf()）
3. 不使用递归语句
4. 减少使用指针对指针的操作
5. 减少使用标准库函数（仅支持部分常用数学函数）
6. 减少使用C++的函数指针和虚函数

时刻铭记：即使在写高级语言，但本质还是在设计硬件！

实验步骤

1. 创建HLS工程
2. 仿真、调试、综合并生成IP核
3. 创建Vivado工程
4. 构建Block Design电路模块图
5. 生成并导出Overlay（.tcl和.bit）
6. 上传Overlay，运行测试程序

验收与提交

- 检查实验现象（**2**分）
- 提交实验报告（**2**分）
  - 报告要求：详见指导书
  - 命名规则：**学号_姓名_DLA实验1.pdf**
  - 提交方法：https://hitsz-cslab.gitee.io/dla/ojguide
  - DDL：下周同一上课时间前

哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

开始实验