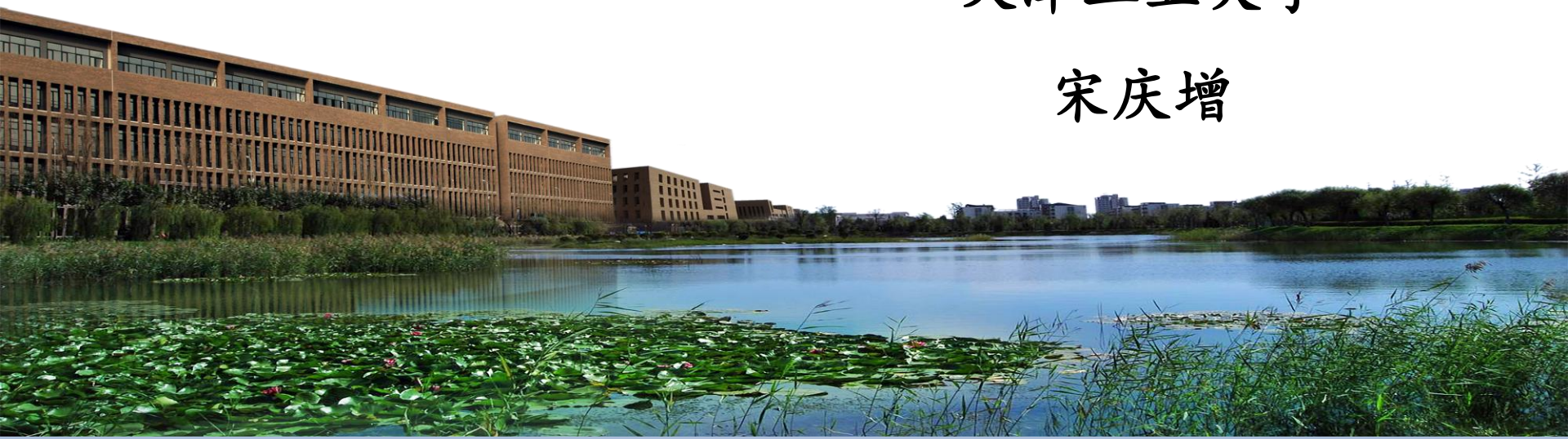
基于FPGA的人脸识别系统的设计与实现

天津工业大学

宋庆增

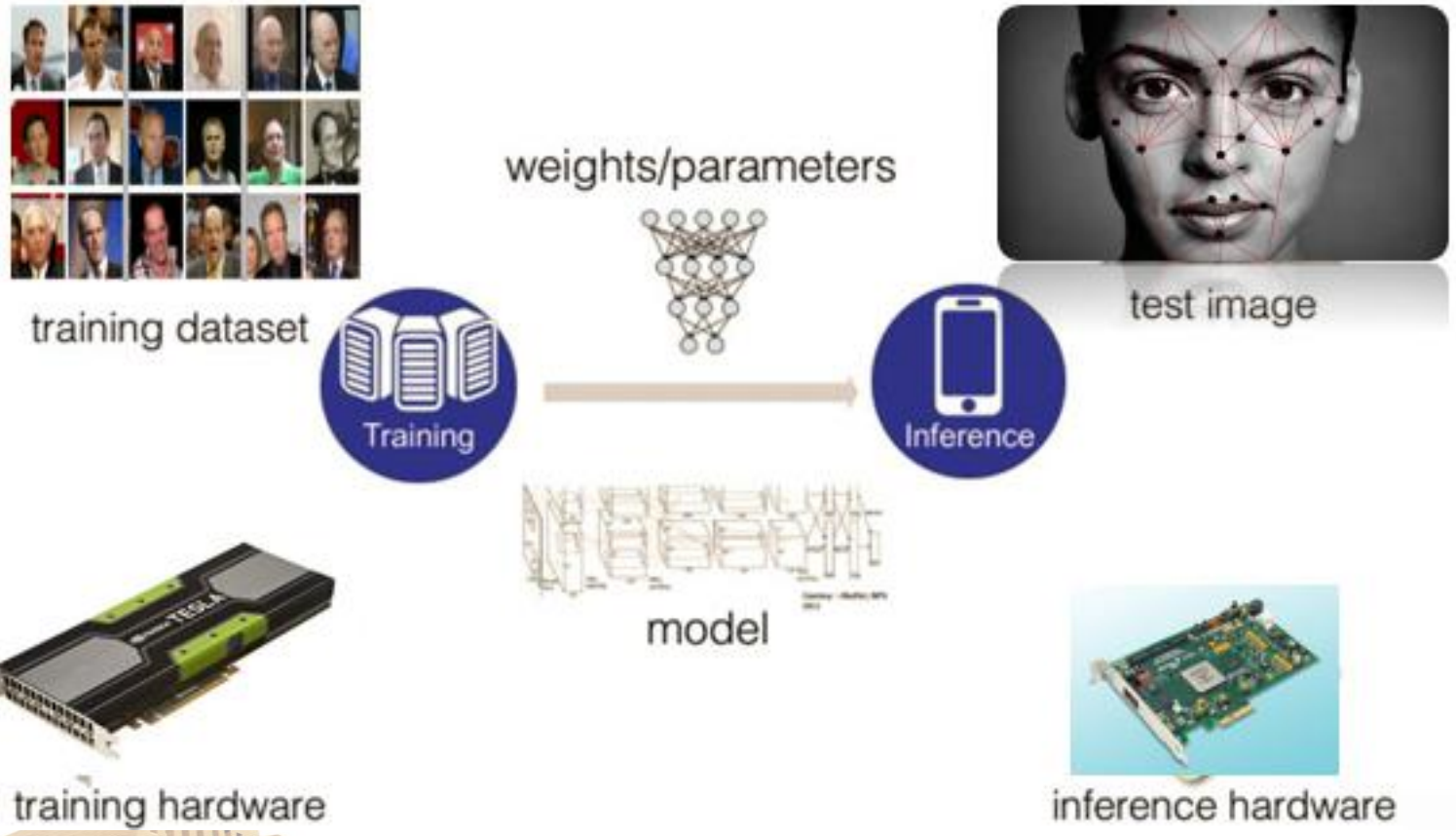


提纲

- 整体构想
- 难点和关键点
- FPGA实现
- 总结与展望

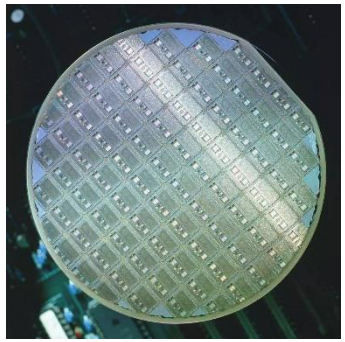
整体构想

人脸识别，是基于人的脸部特征信息进行身份识别的一种生物识别技术。用摄像机或摄像头采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行脸部识别。 它集成了人工智能、深度神经网络、视频图像处理等多样专业技术。


weights/parameters
training dataset
test image
Training
Inference
model
training hardware
inference hardware

可供的选择硬件

GPU



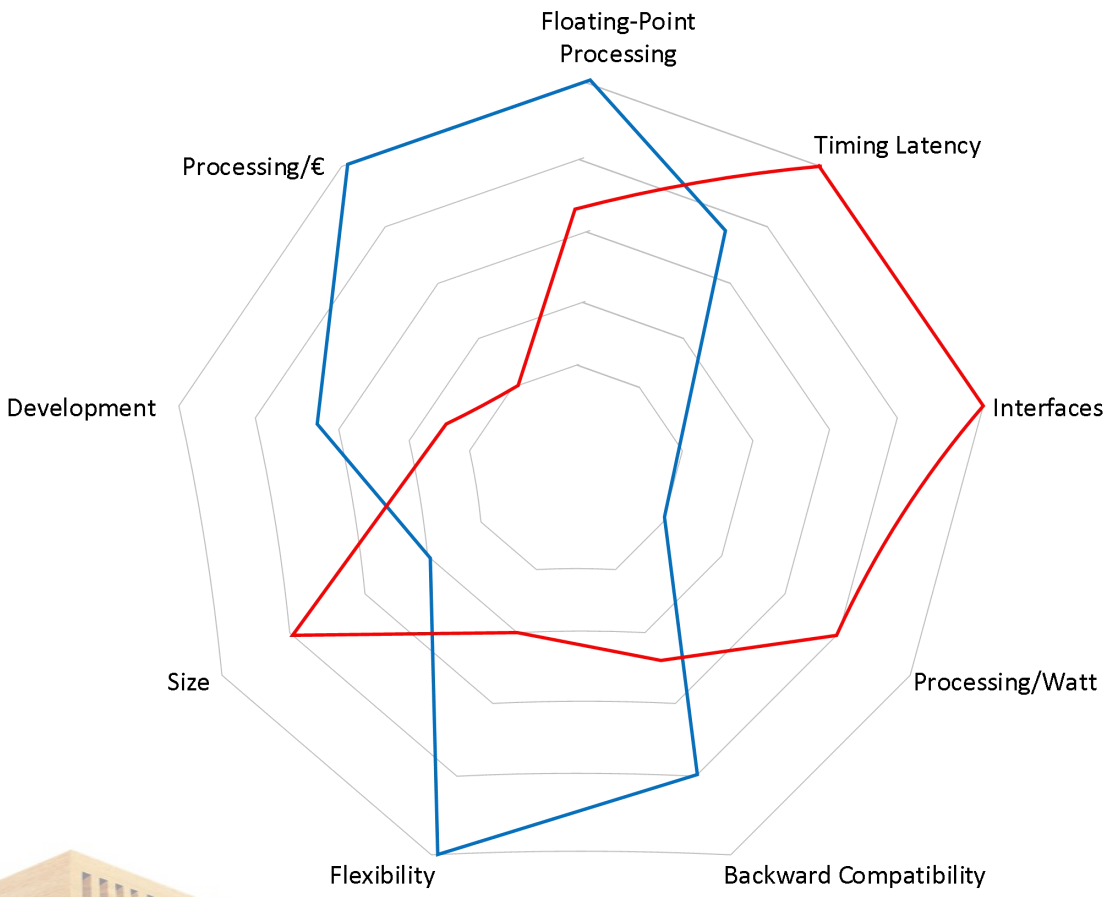
ASIC

FPGA

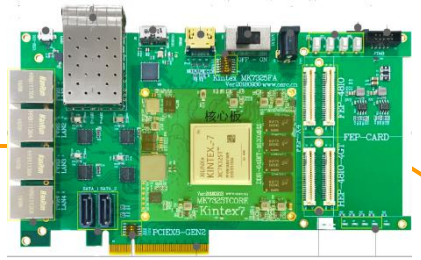
TX2

NPU

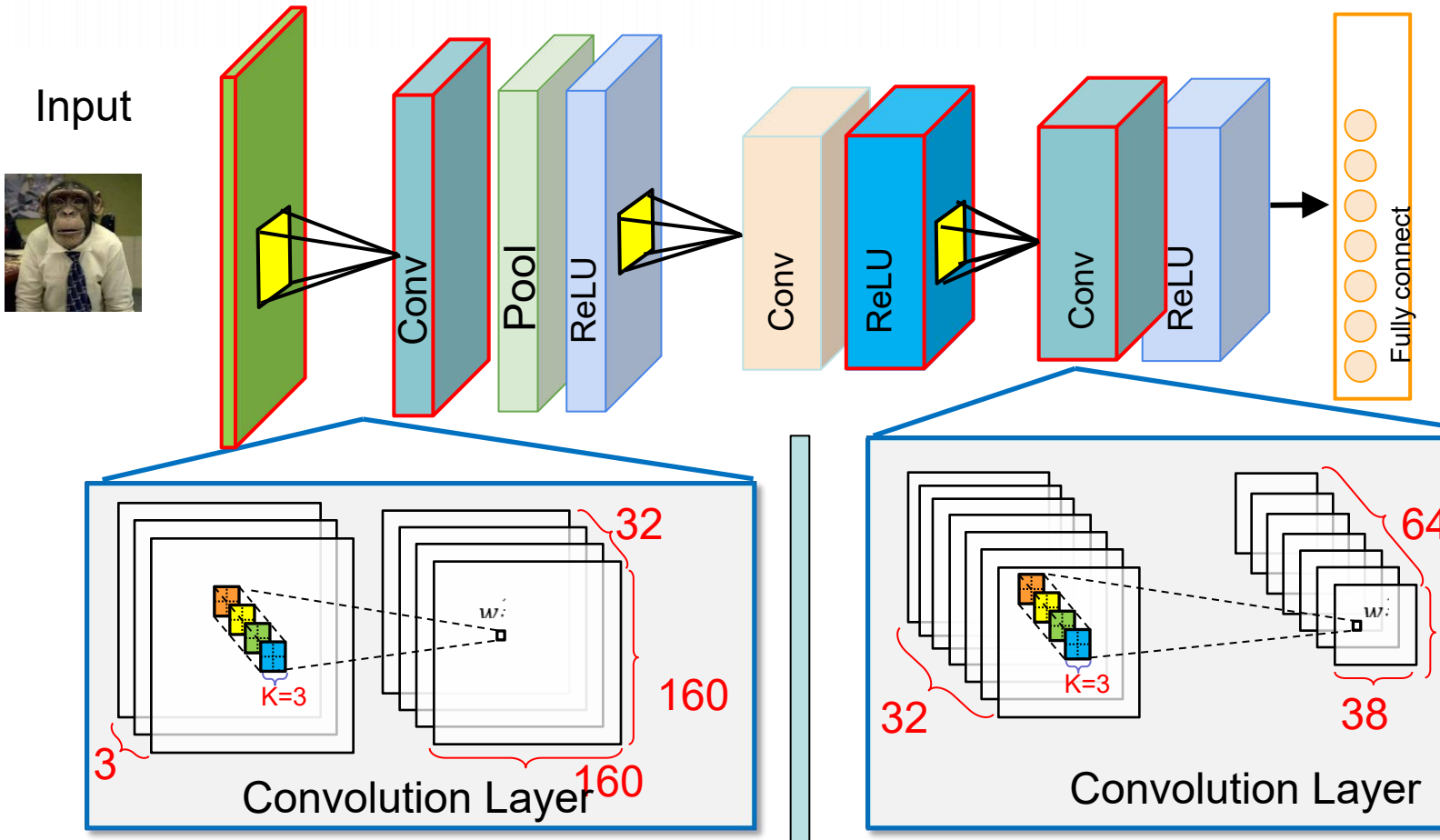
优缺点

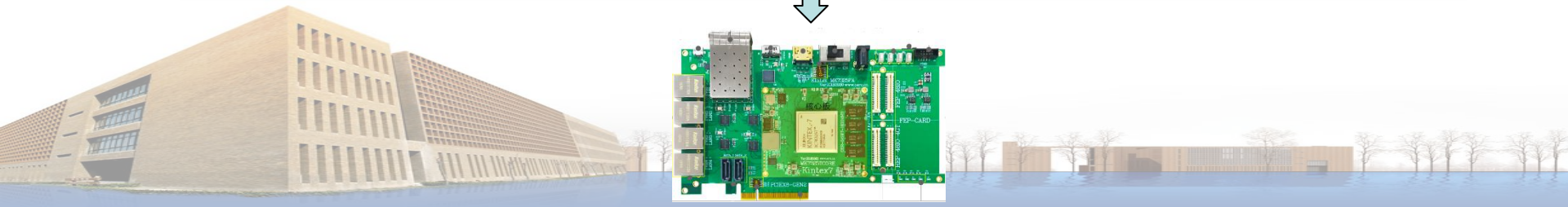

Floating-Point Processing
Timing Latency
Interfaces
Processing/Watt
Backward Compatibility
Flexibility
Size
Development
Processing/€
GPU
FPGA

整体框架



映射到FPGA


Input
Conv
Pool
ReLU
Conv
ReLU
Conv
ReLU
Fully connect
32
160
K=3
3
160
Convolution Layer
64
38
K=3
32
38
Convolution Layer



可供的选择算法

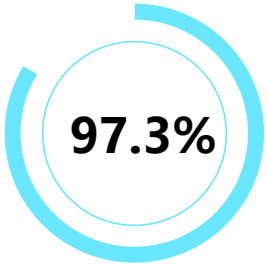
SeetaFace

中科院 山世光

C++实现

人脸检测速度慢

准确率低



DeepFace

需3D对齐处理。

生成的特征向量拼接成高维向量并采用PCA再次进行降维。

Face++

商业化最好的库

FaceNet

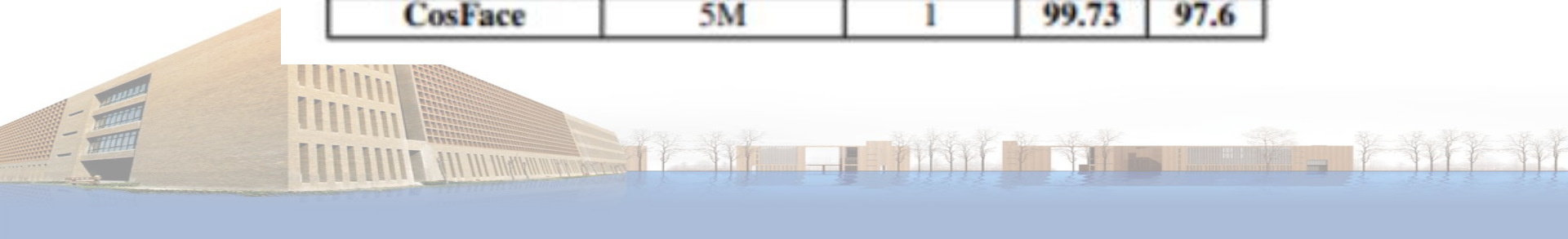
端到端的学习

128维特征向量

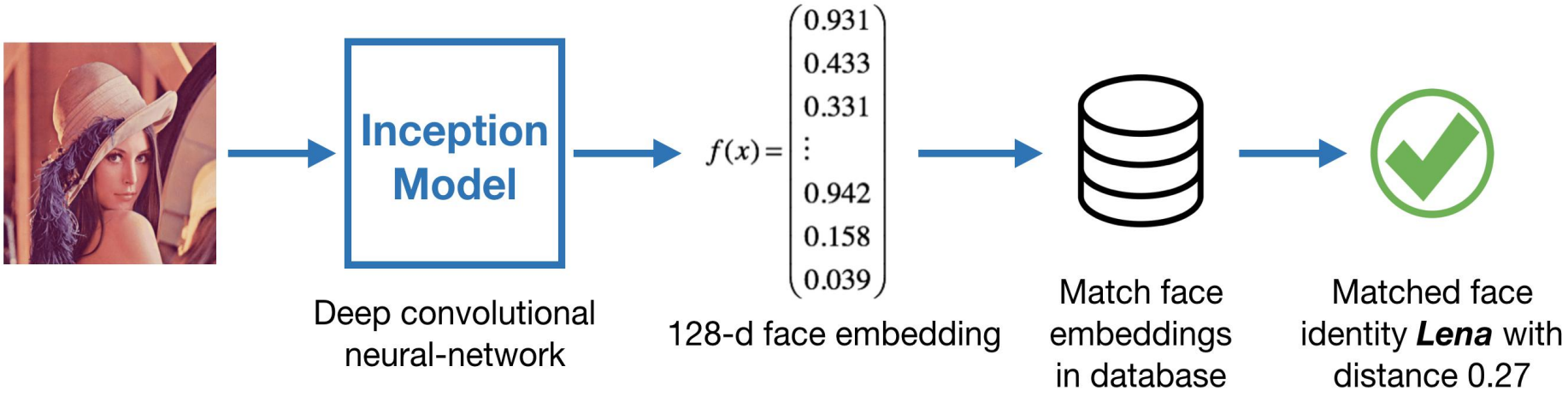
精度对比

No.	Method	Images	Networks	Acc.
1	Fisher Vector Faces [21]	-	-	93.10
2	DeepFace [29]	4M	3	97.35
3	Fusion [30]	500M	5	98.37
4	DeepID-2,3		200	99.47
5	FaceNet [17]	200M	1	98.87
6	FaceNet [17] + Alignment	200M	1	99.63

Method	Training Data	#Models	LFW	YTF
Deep Face[35]	4M	3	97.35	91.4
FaceNet[29]	200M	1	99.65	95.1
DeepFR [27]	2.6M	1	98.95	97.3
DeepID2+[33]	300K	25	99.47	93.2
Center Face[42]	0.7M	1	99.28	94.9
Baidu[21]	1.3M	1	99.13	-
SphereFace[23]	0.49M	1	99.42	95.0
CosFace	5M	1	**99.73**	**97.6**



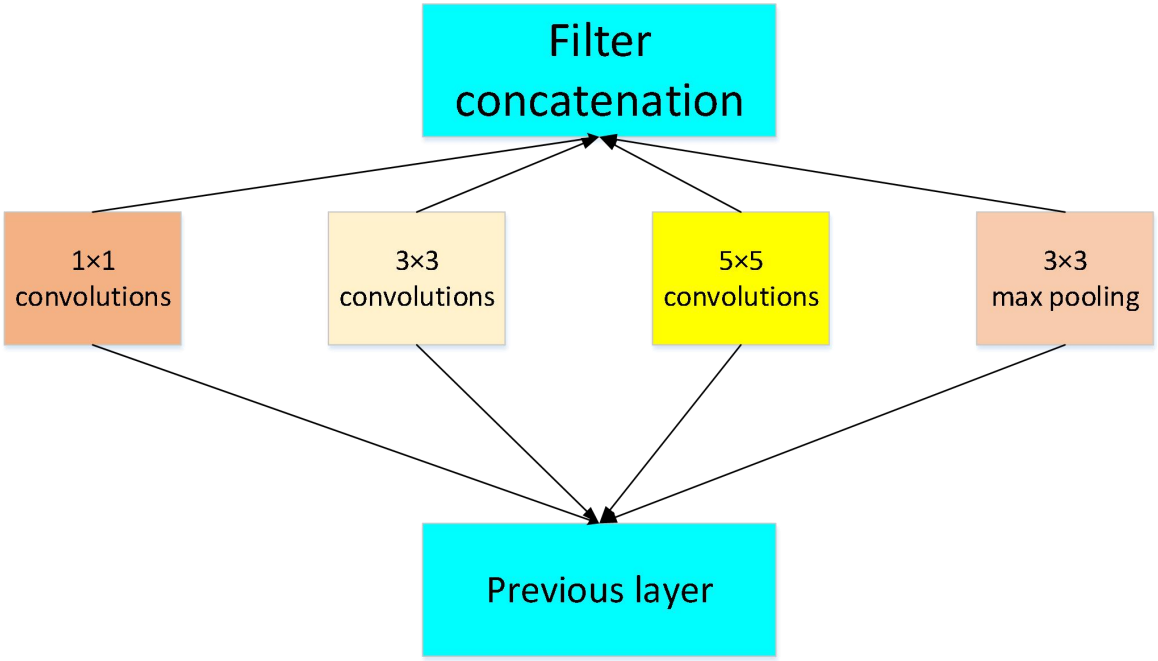
FaceNet模型结构



Facenet是一个通用的系统，采用CNN神经网络将人脸图像映射到**128维**的欧几里得空间，我们可以根据两幅人像的欧几里得距离去判断两个人像的相似程度。两个人像之间的欧几里得距离越近，说明它们越相似。

Inception V

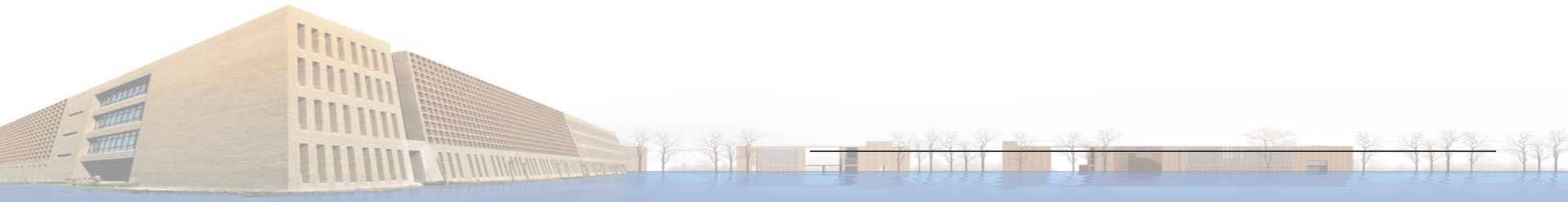
Inception v1
Inception v2
Inception v3



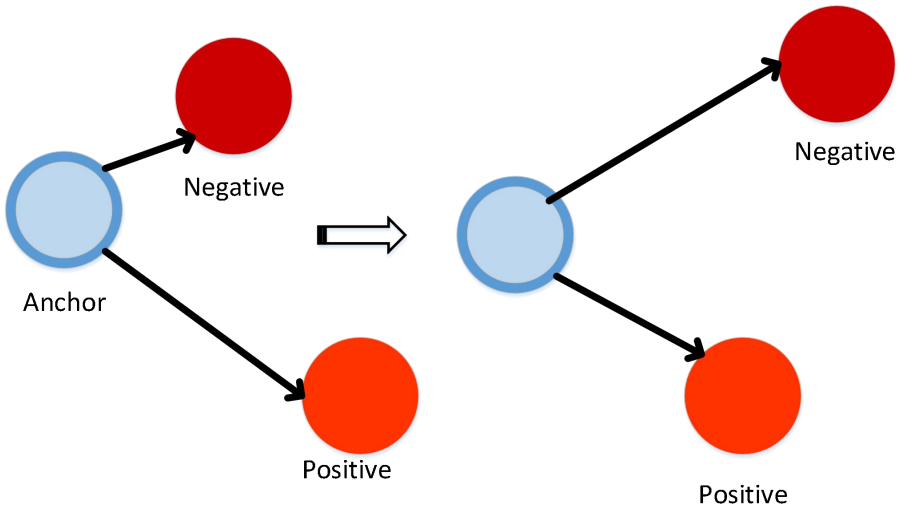
Inception v4（Inception-ResNetv1&Inception-ResNetv2）

L2归一化和三元组损失函数

L2 归一化层，是将特征进行归一化处理，通过欧几里德距离判断人脸的相似度。经过归一化后，图像的特征都会被映射到一个超球面上，接着再去优化这些特征。这里采用了三元组损失函数（tripletloss）训练网络。使用三元组数据训练，使得相同身份之间的特征距离要尽可能的小，而不同身份之间的特征距离要尽可能的大。最后进行人脸验证时候，只需要直接计算128维特征向量。



三元组优化



从训练数据集中随机选一个样本，该样本称为Anchor，然后再随机选取一个和Anchor(记为 x_a)属于同一类的样本和不同类的样本，这两个样本对应的称为Positive (记为x_p)和Negative (记为x_n)，由此构成一个（Anchor，Positive，Negative）三元组。

在这里，要确保一个特定人员的人脸图像 x_i^a（Anchor）更接近同一个人的所有其他图 x_i^p （Positive），而不是任何其他人的任何图 x_i^n （Negative）。

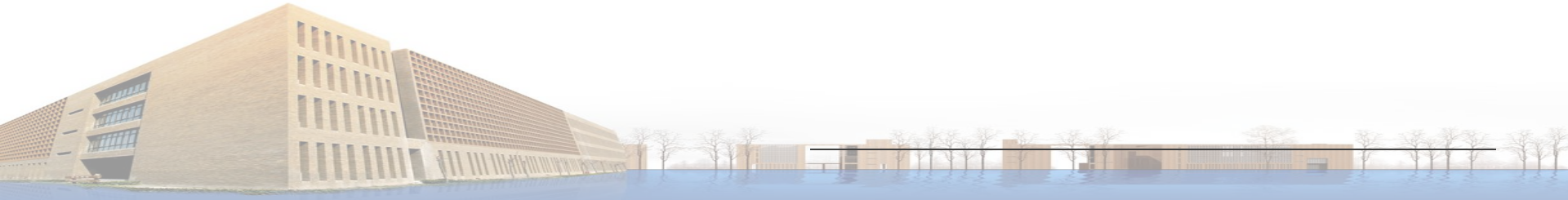
Triplet_loss目标函数

$$\sum_{i}^{N}\left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha\right]_+$$

f(x)表示嵌入层，它将图像嵌入到 d 维欧几里得空间中。

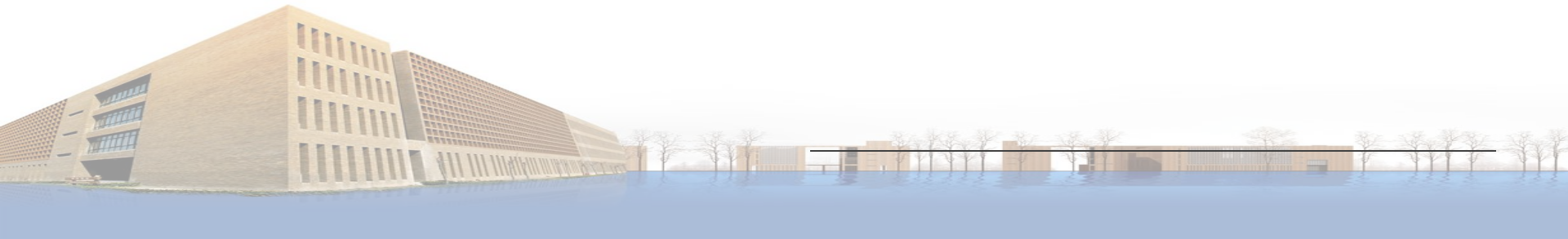
$\| f(x_i^a) - f(x_i^p) \|_2^2$ 表示的是Positive元和Anchor之间的欧式距离度量。

$\| f(x_i^a) - f(x_i^n) \|_2^2$ 表示的是 Negative 和 Anchor 之间的欧式距离度量。



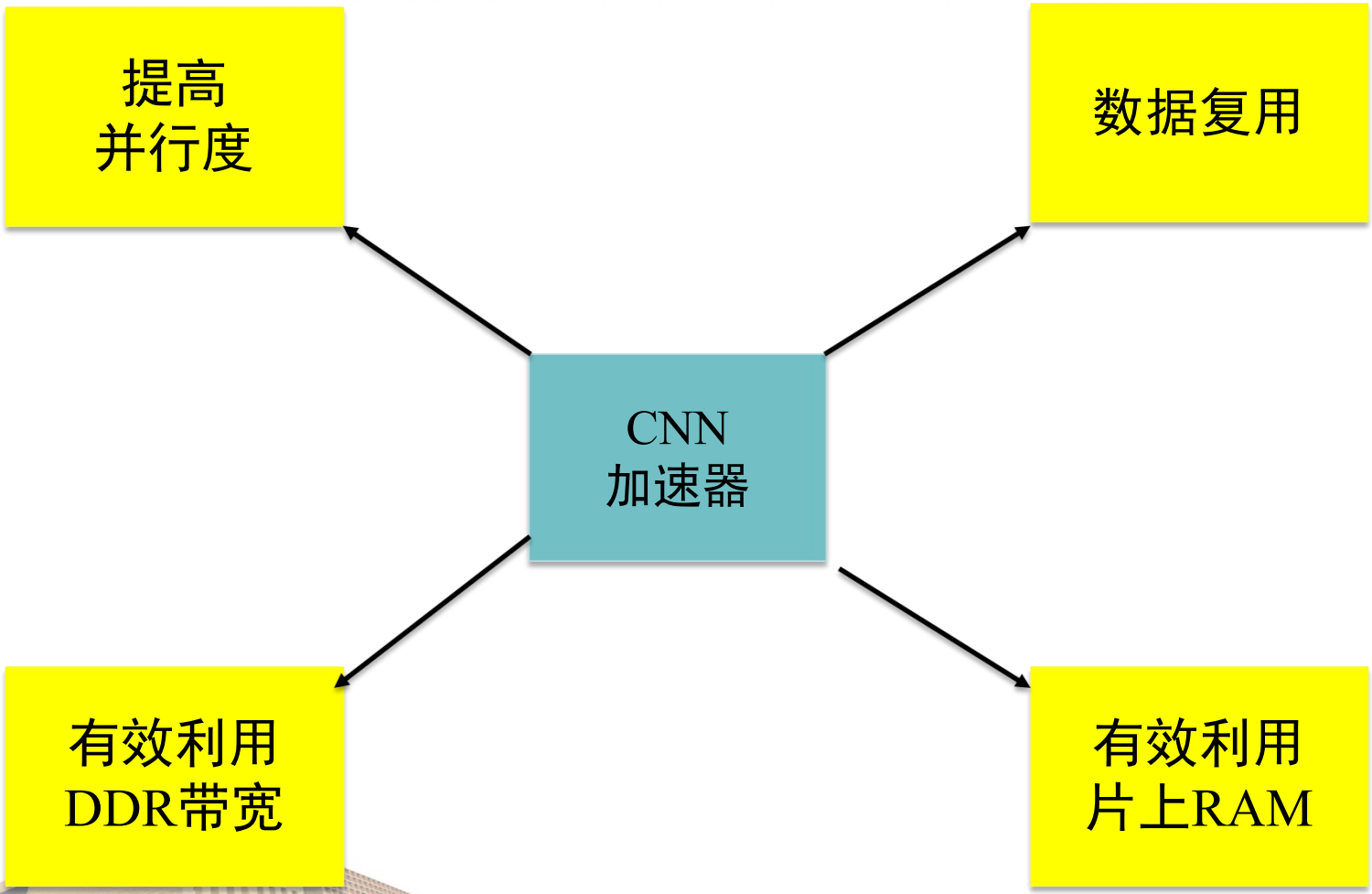
与其他的深度学习方法的不同

FaceNet并没有用传统的softmax的方式去进行分类学习，然后抽取其中某一层作为特征，而是直接进行**端对端学习**一个从图像到欧式空间的编码方法，然后基于这个编码再做人脸识别、人脸验证和人脸聚类等。

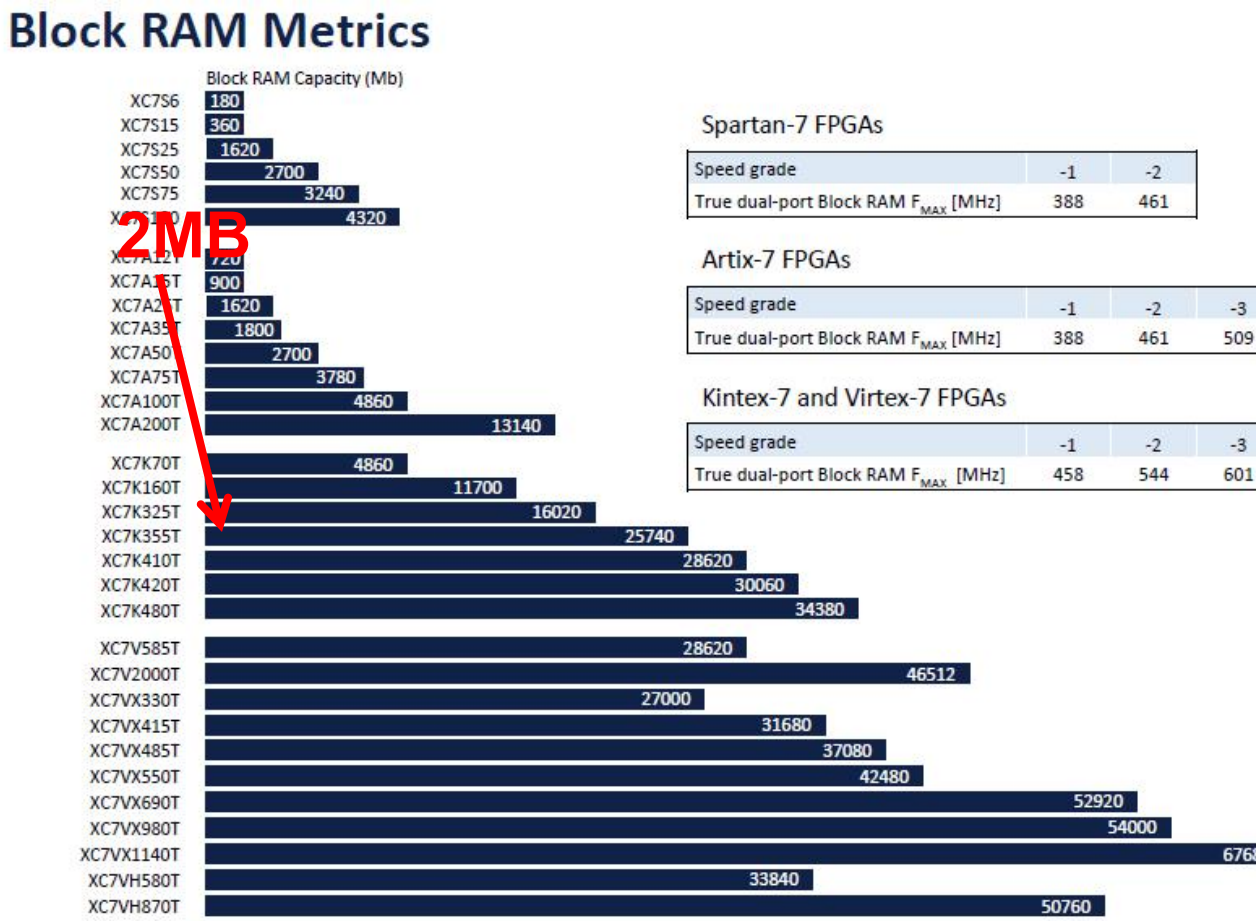


难点和关键点

难点



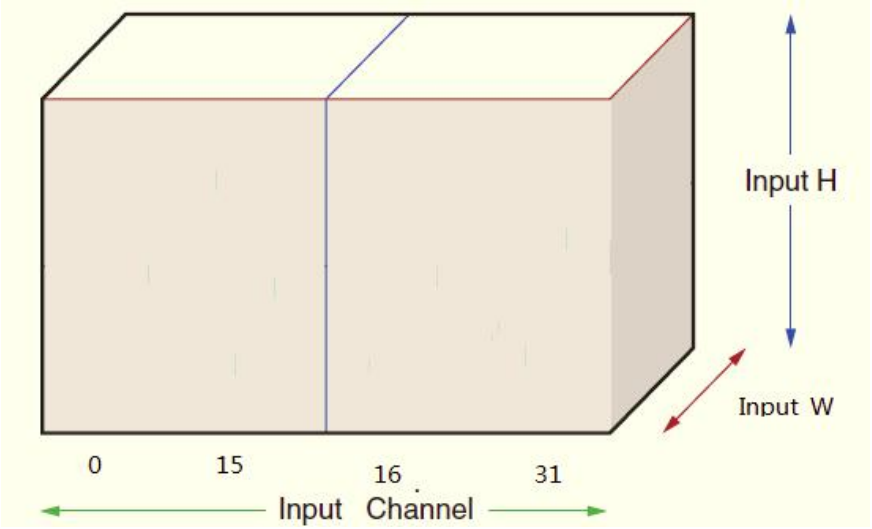
特征图和权重量大，
无法放入FPGA中，
需要通过DDR。
参数>100MB
中间结果数量级为
GB
即使连一层的中间结果也无法完全保存



分解

输入特征分段
权重分段
输出结果分段

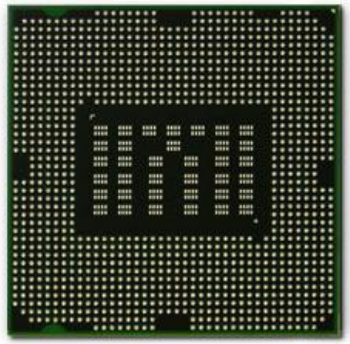
输入 32 （2）
输出 64 （4）



Weight 分段 （3,3,32,64）？

主机配置

麒麟+飞腾1500A



	飞腾1500
处理器	FT-1500A
主频	**1.8GHz**
内存	华芯DDR3-4G
硬盘	TOSHBA 1.0T
显卡	独显
操作系统	银河麒麟

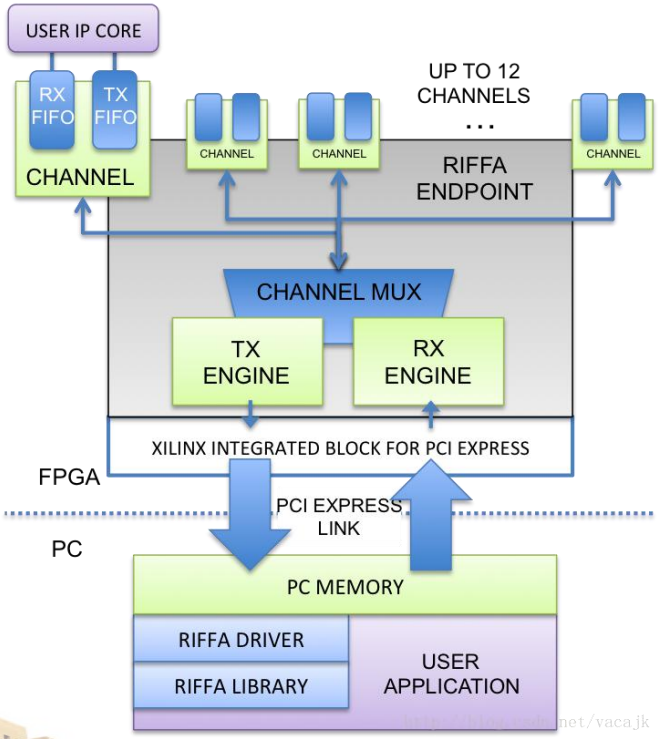
自主可控

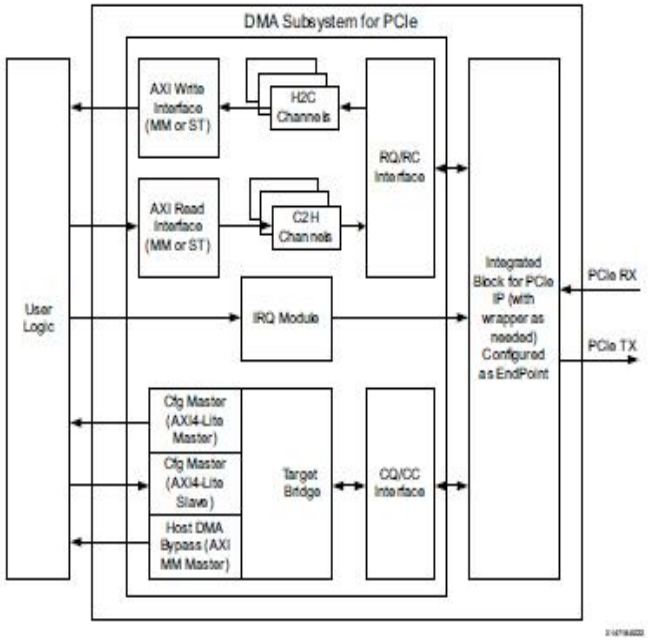
PCI-E的选择

RIFFA

XPDMA

XDMA





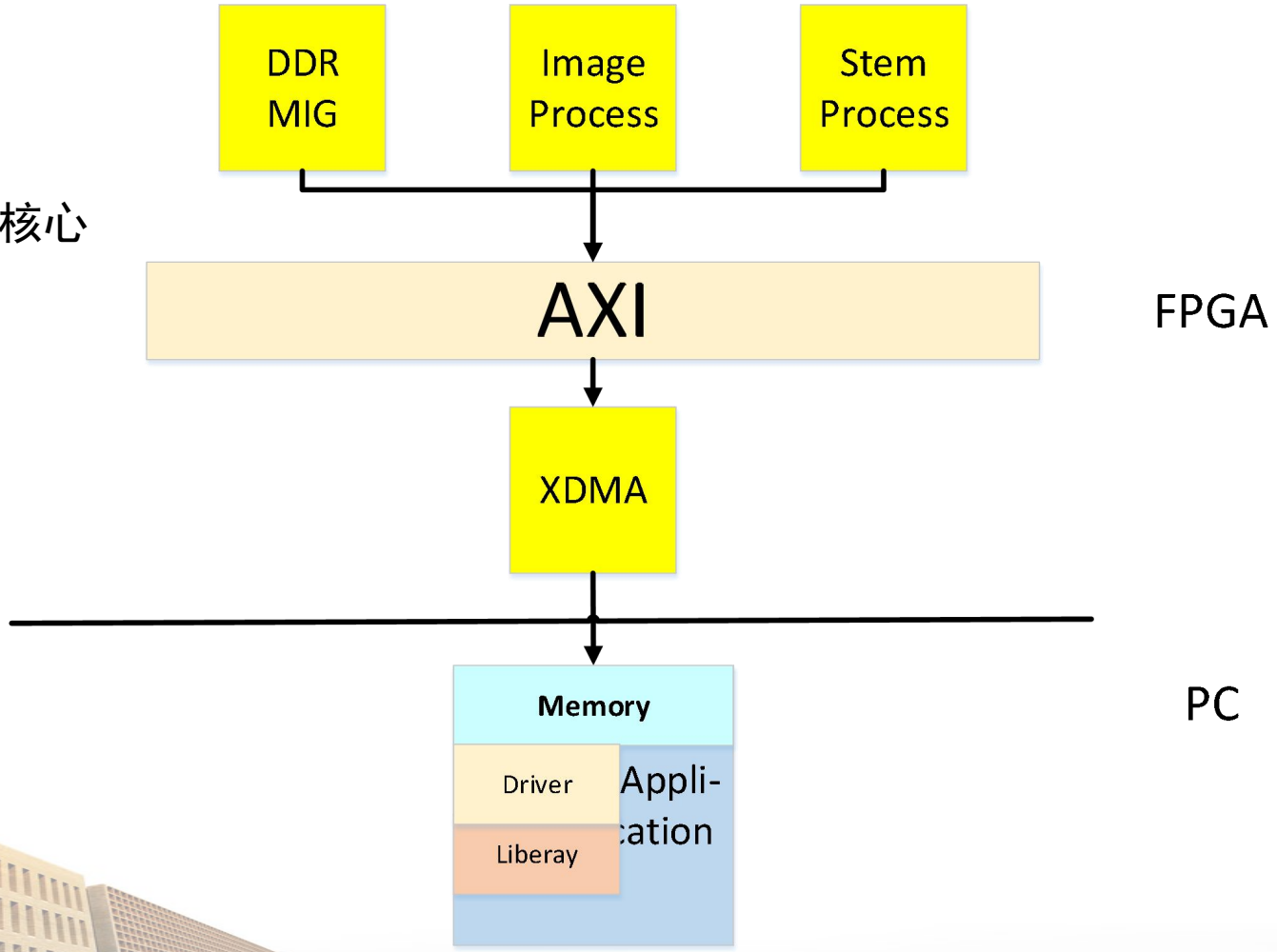
解决方案

- 选择合适的PCI控制IP
- 移植PCI驱动
- 移植上位机

FPGA设计

框架

以AXI总线为核心



控制流程

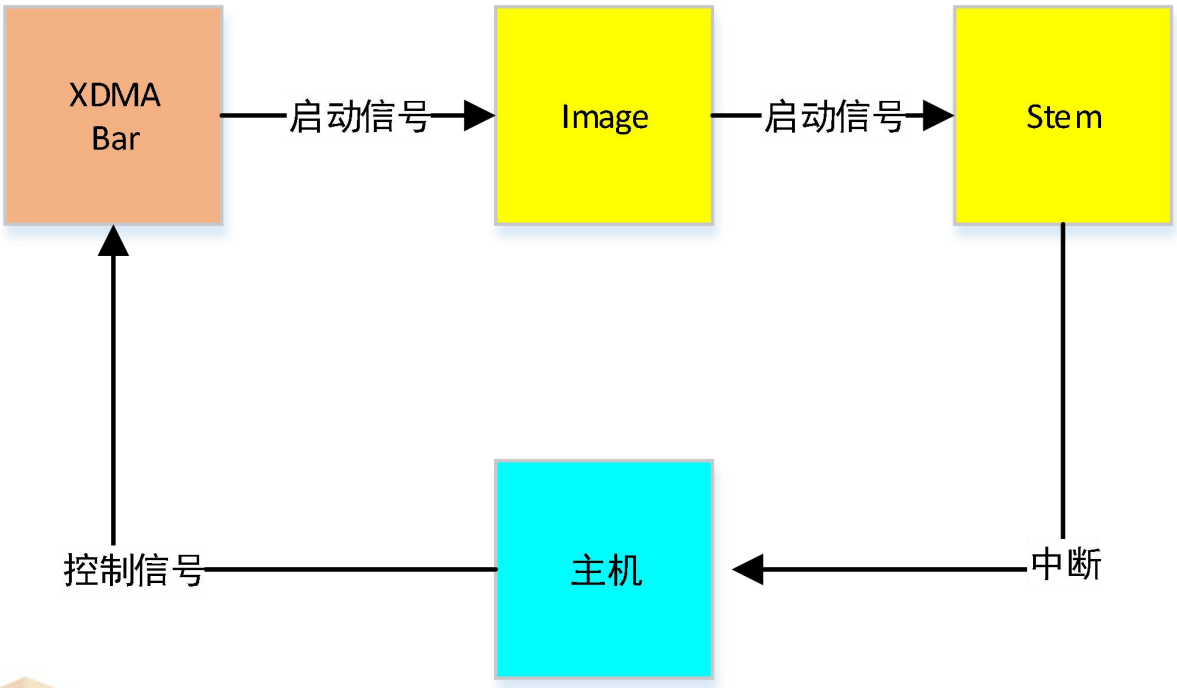
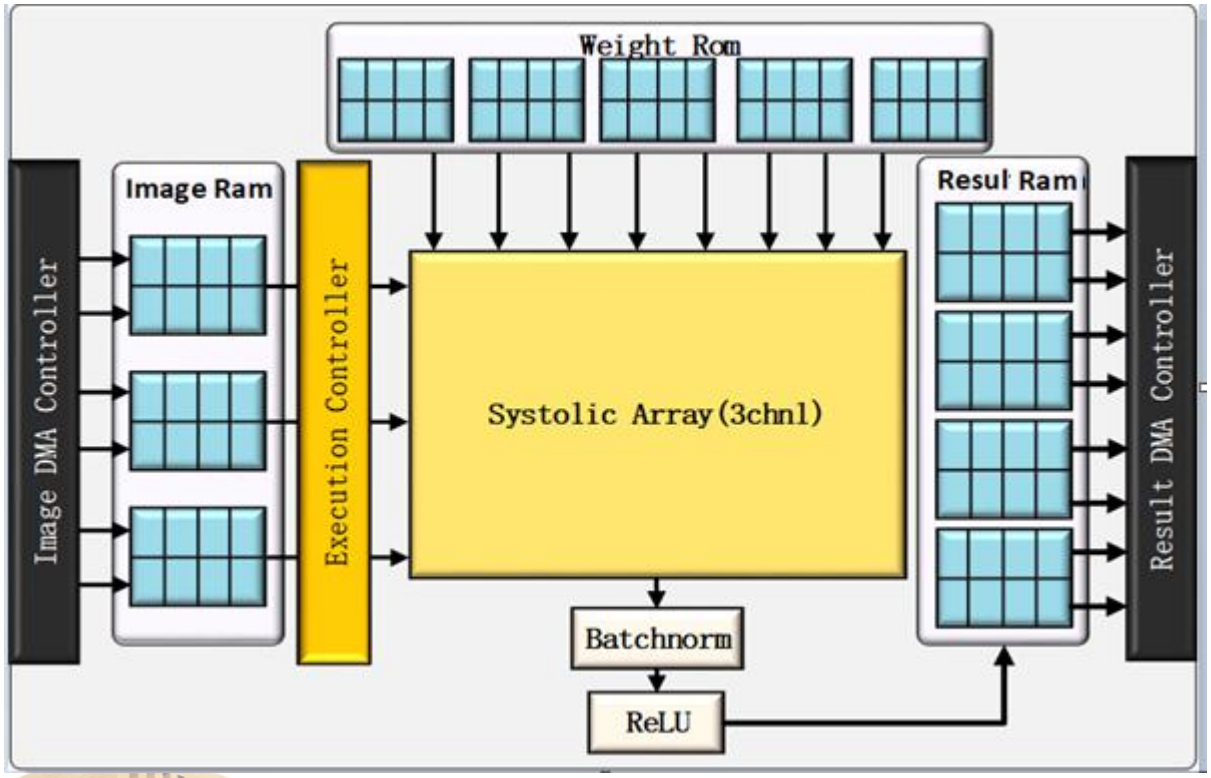
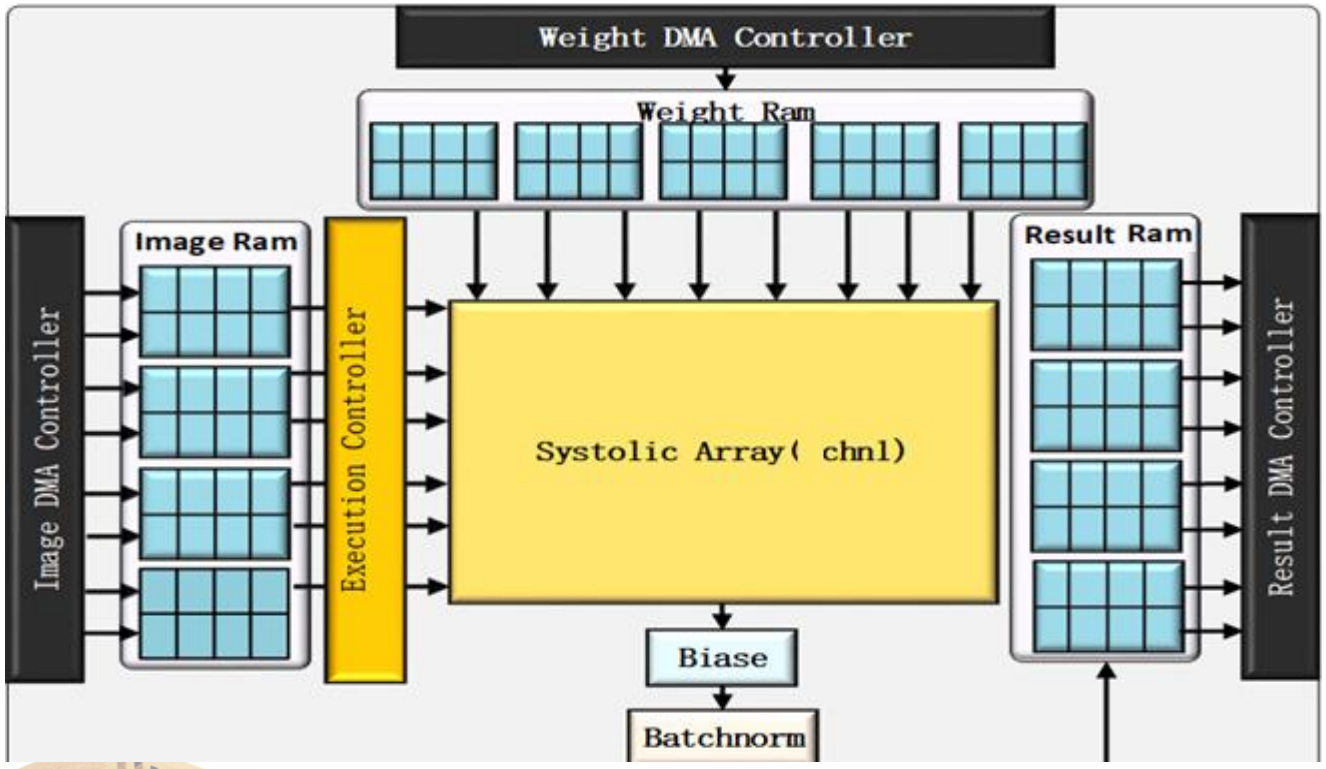


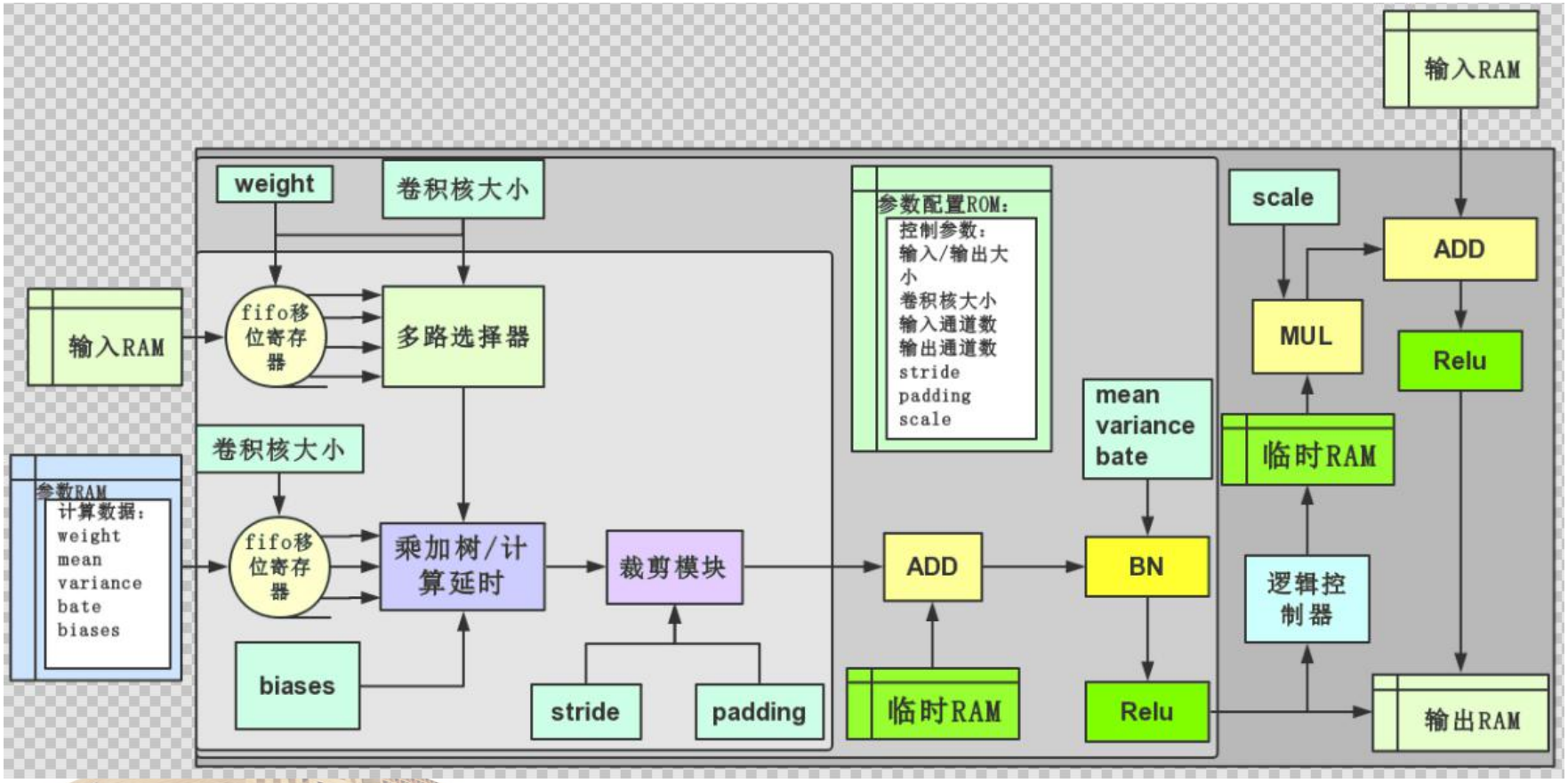
Image Process



Stem Process



关键模块


输入RAM
weight
卷积核大小
参数配置ROM:
控制参数:
输入/输出大小
卷积核大小
输入通道数
输出通道数
stride
padding
scale
scale
ADD
MUL
Relu
fifo移位寄存器
多路选择器
mean
variance
bate
临时RAM
卷积核大小
参数RAM
计算数据:
weight
mean
variance
bate
biases
fifo移位寄存器
乘加树/计算延时
裁剪模块
ADD
BN
逻辑控制器
biases
stride
padding
临时RAM
Relu
输出RAM

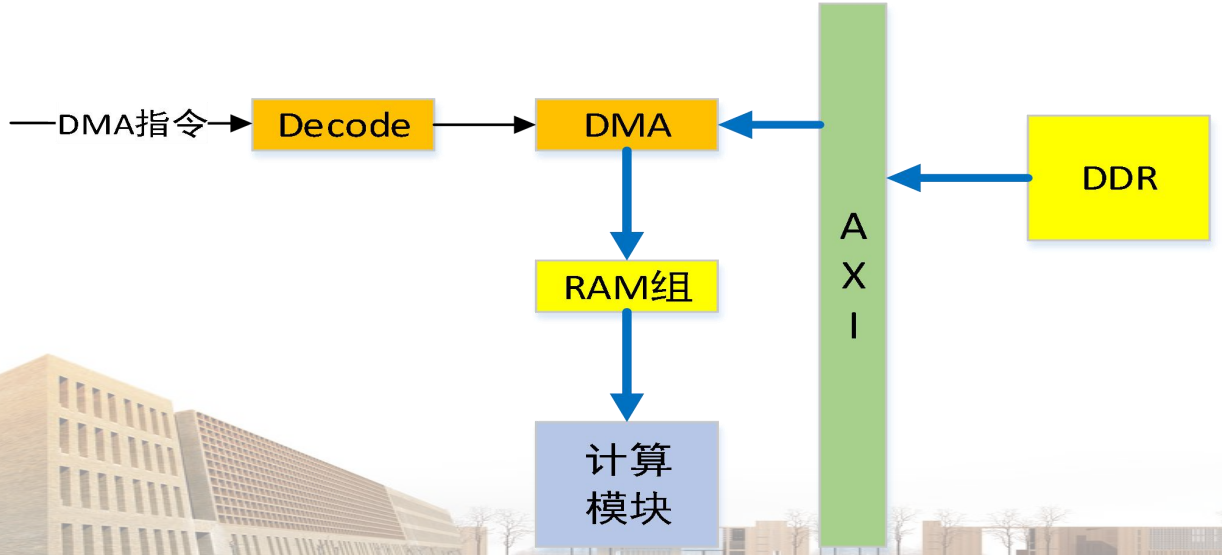
状态机

指令编码	名称	代表的意义
0000_0000	Rst_state	复位（不用）
0000_0001	Read_Instruction_state	读指令
0000_0010	DMA_Rd_DDR_state	读DDR数据
0000_0100	Con2d_state	启动一次16通道的卷积计算
0000_1000	DMA_Weight_state	启动一次写Weight FIFO
0001_0000	DMA_Wr_DDR_state	计算结果写回DDR
0010_0000	End_state	结束（最后一条指令）

DMA指令格式

00100000　00000800　00000080　000000001

DDR地址　数据量　RAM编号



控制与参数

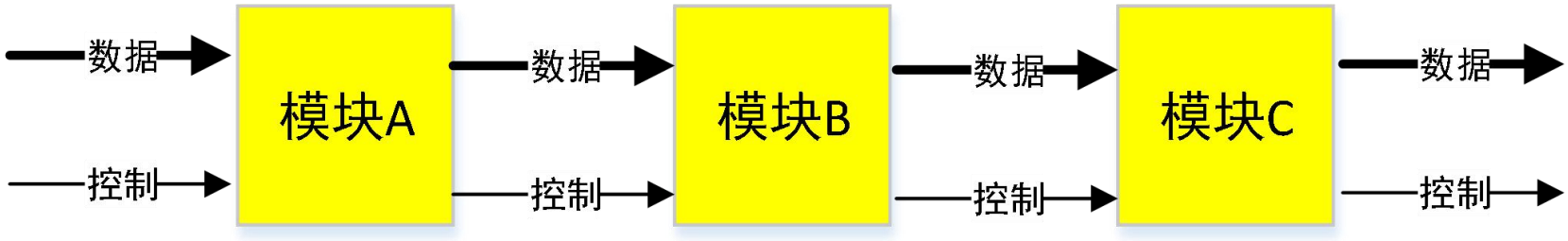
控制流指令

63：20	19:16	15:7	6	5	4	3	2	1	0
保留	Compute_ID	保留	BN_en	ReLu_en	Pool_en	Padding	Stride	Add_zero	Wr_back_en
						Same =1	有stride =1	有 =1	
						Valid = 0	没有 stride =0	没有=0	

参数指令

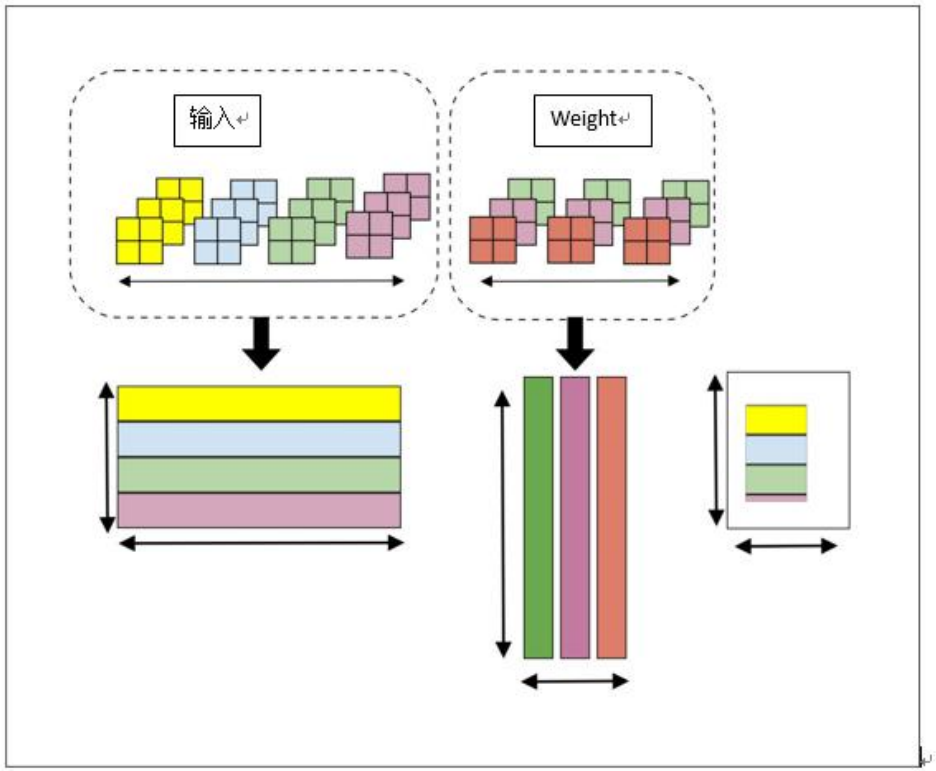
63:40	39:16	15:7	7:0
Num_feature_one_out	Num_feature_one	Num_row_column_out	Num_row_column

数据流控制格式

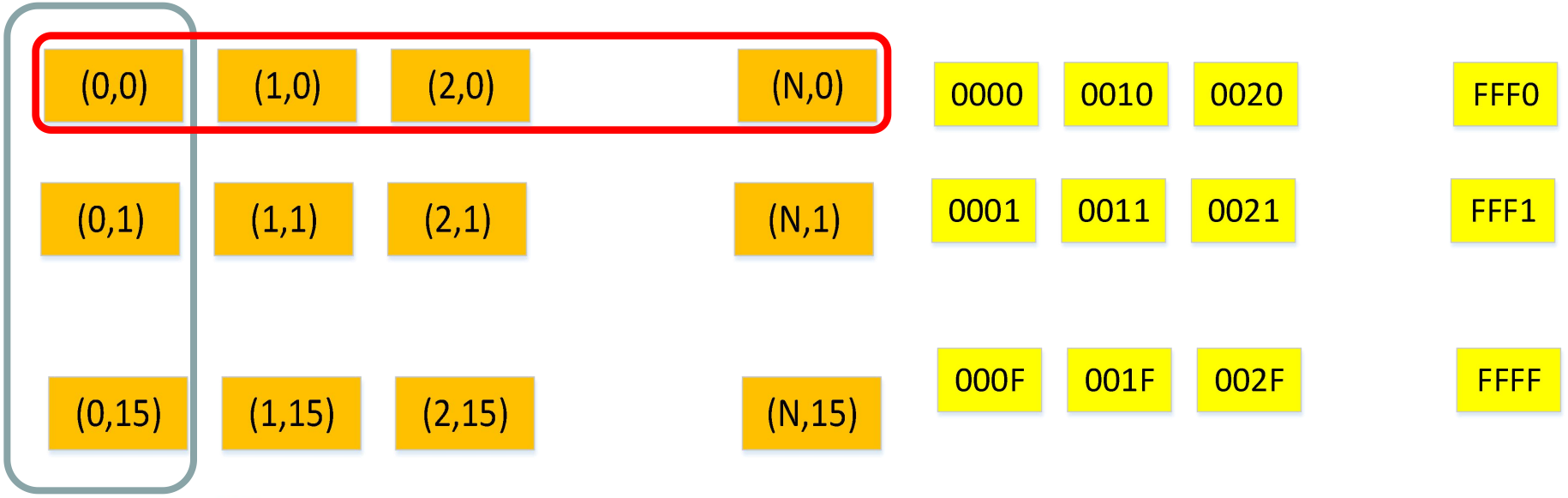


row_last, 一行的结尾
column_last, 一个通道的结尾
data_valid, 数据有效
complete 16个通道的结尾

并行度



结果重排/写回


(0,0)
(1,0)
(2,0)
(N,0)
0000
0010
0020
FFF0
(0,1)
(1,1)
(2,1)
(N,1)
0001
0011
0021
FFF1
(0,15)
(1,15)
(2,15)
(N,15)
000F
001F
002F
FFFF
Wr_back_en

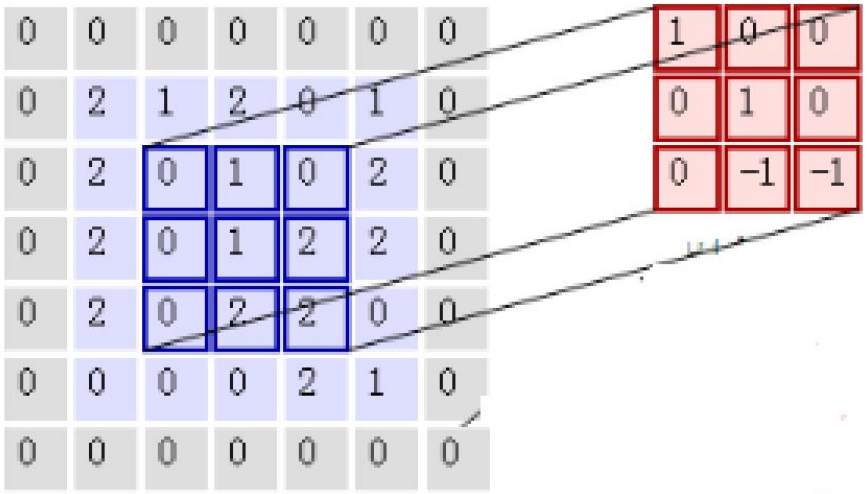
卷积计算的选择

空间卷积　　（主流）
通用矩阵乘法
FFT算法　　（快速算法）
Winograd 算法（快速算法）

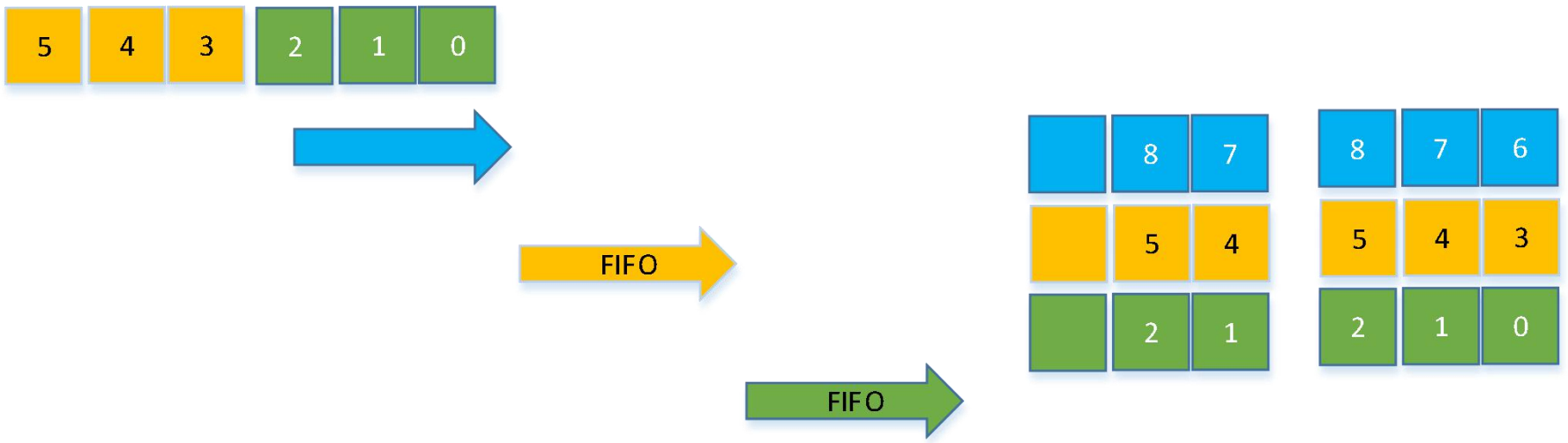
卷积计算

二维滑窗

```
// 0 1 2
// 3 4 5
// 6 7 8
//上述3x3的滑窗对应的次序是 data_out的次序是 MSB[0 1 2 3 4 5 6 7 8]LSB,
```



数据重排



Padding

填充

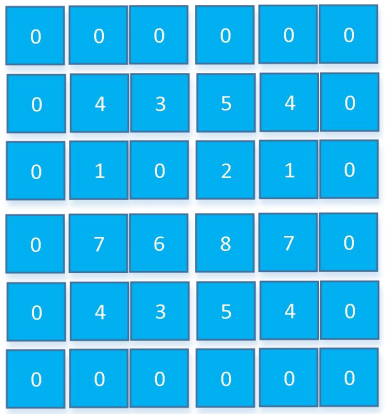
0	0	0	0	0	0
0	4	3	5	4	0
0	1	0	2	1	0
0	7	6	8	7	0
0	4	3	5	4	0
0	0	0	0	0	0

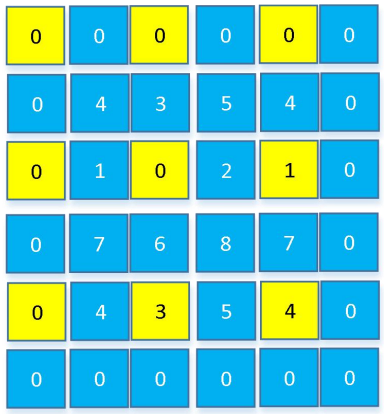
裁剪

4	3	5	4
1	0	2	1
7	6	8	7
4	3	5	4

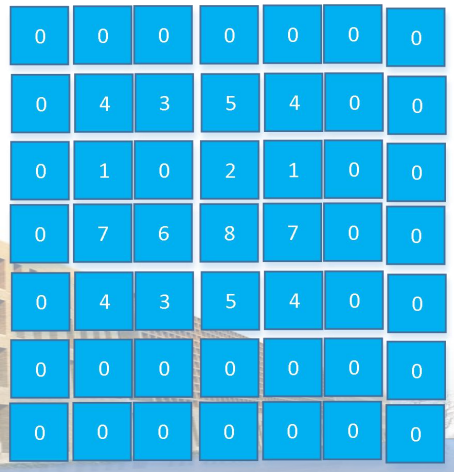
Padding_EN

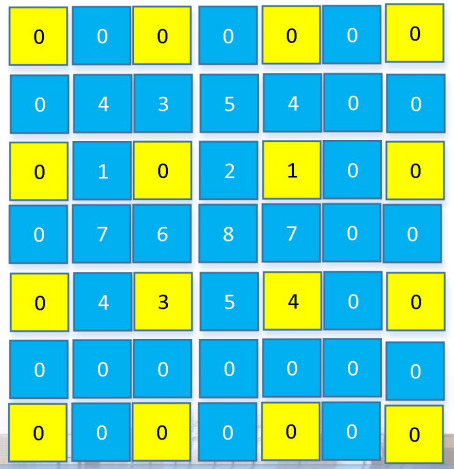
Stride



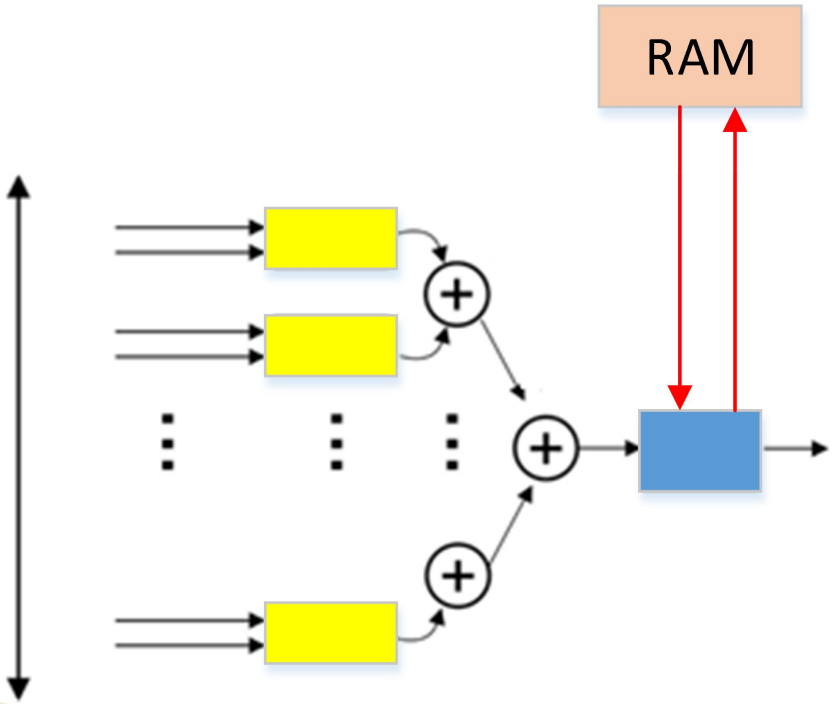


Stride_EN





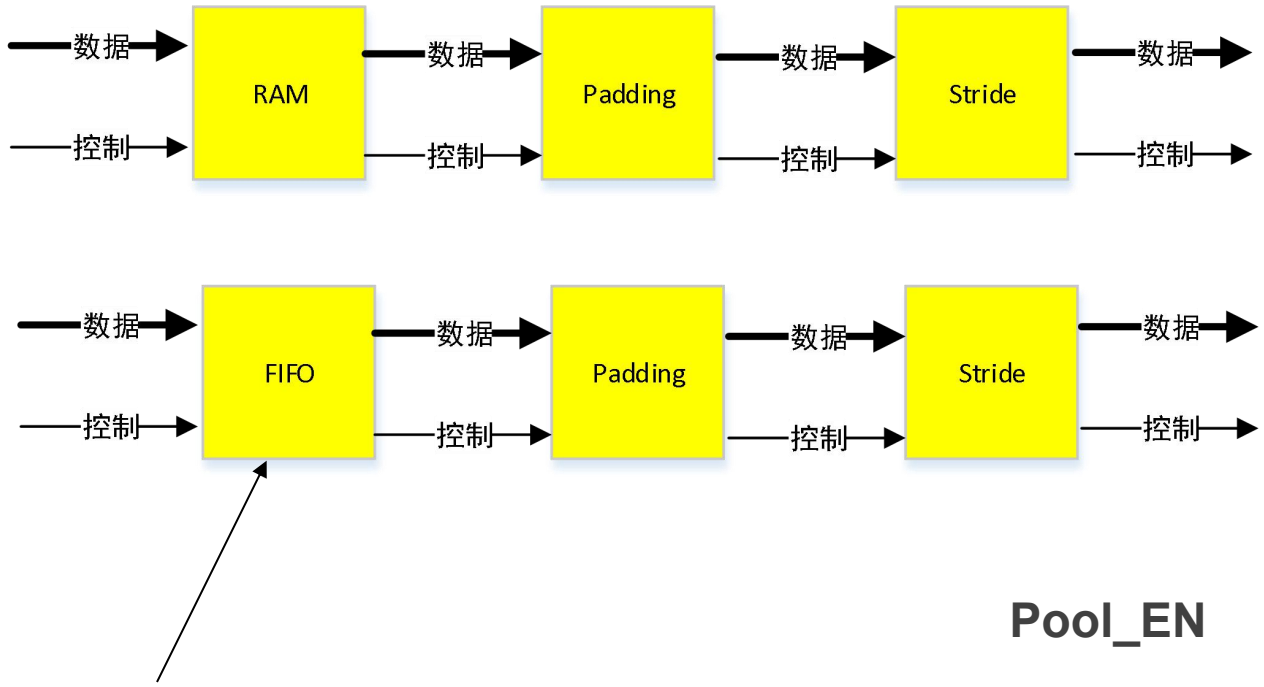
规约操作



Add_zero_EN

Pool层

两种方案


数据
控制
RAM
数据
控制
Padding
数据
控制
Stride
数据
控制
数据
控制
FIFO
数据
控制
Padding
数据
控制
Stride
数据
控制
Pool_EN
排空

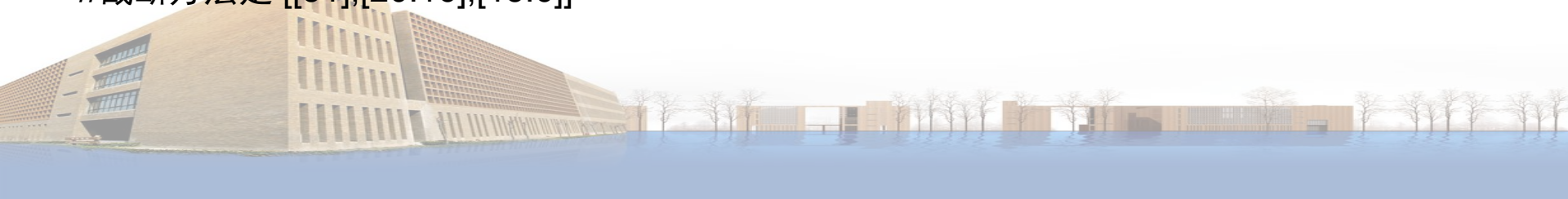
小数点定标

输入、输出，BatchNorm等:　　1:5:10

Weight参数:　　1:1:14

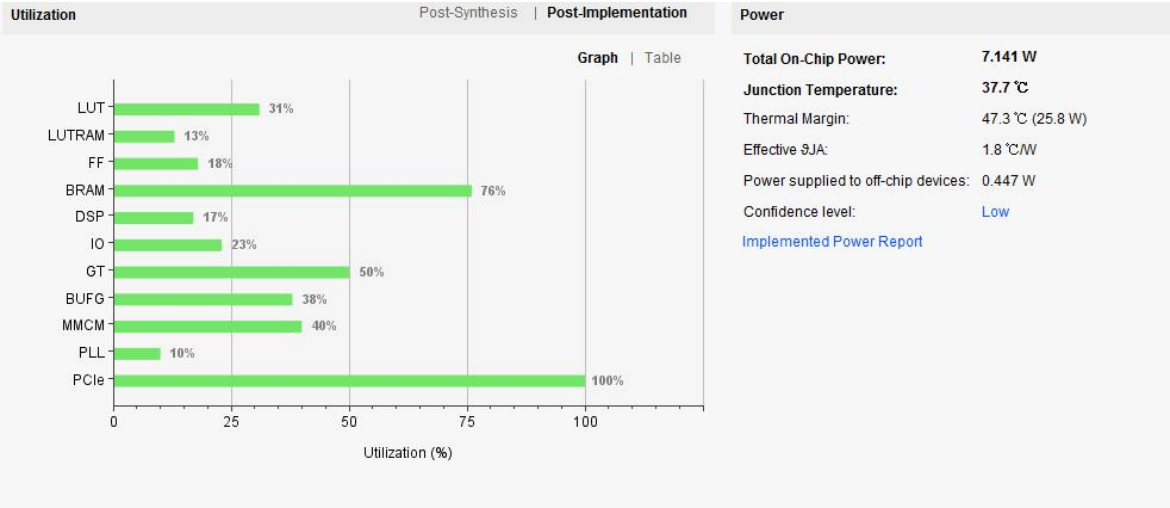
中间结果：　　32位 动态调整

//32为定点变16位定点模块
//16位数据格式为 1 5 10（符号位，整数位，小数位），[15],[14:10],[9:0]],
//32位数据格式为 1 11 20 ,[[31],[30:20],[19:0]]
//加法操作挪动了小数点4次，因此数据变为 1 15 16 ，[[31],[30:16],[15:0]]
//截断方法是 [[31],[20:16],[15:6]]



资源利用

资源	LUT	BRAM	FF	DSP
利用	62757	325	74729	145

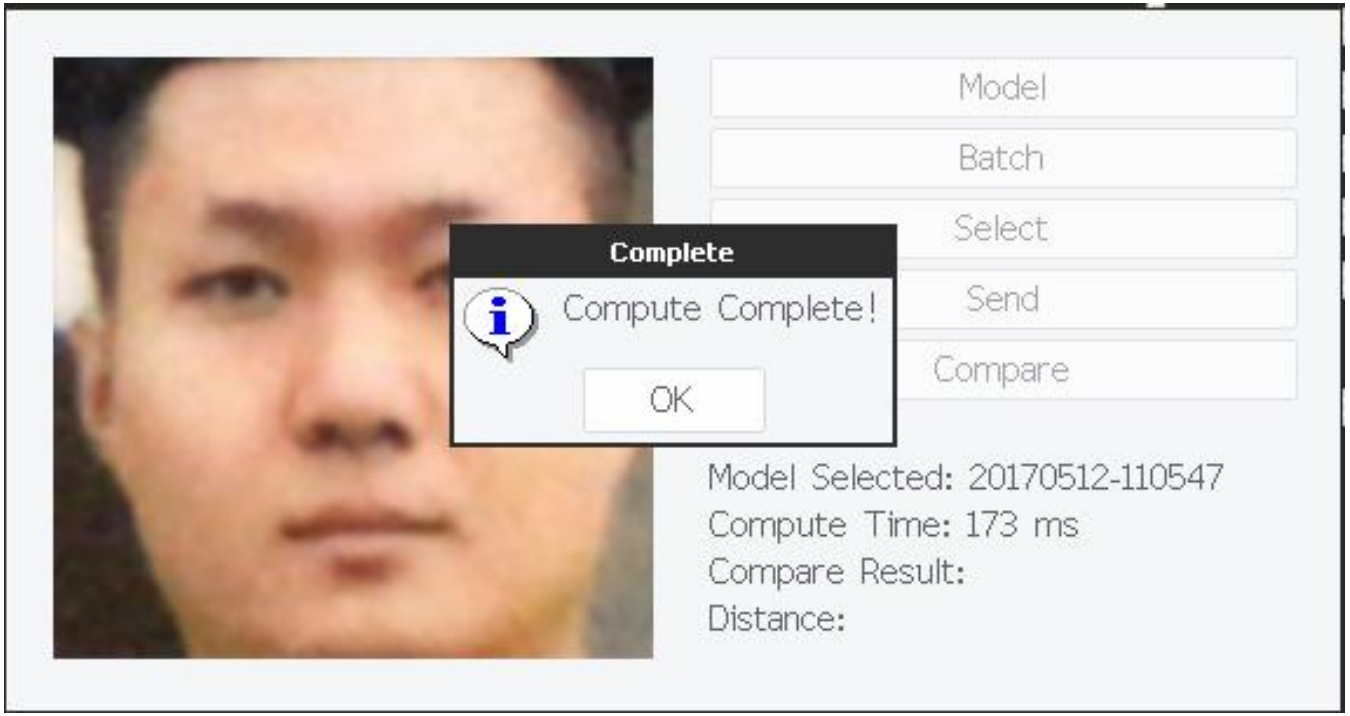


时钟:200MHz

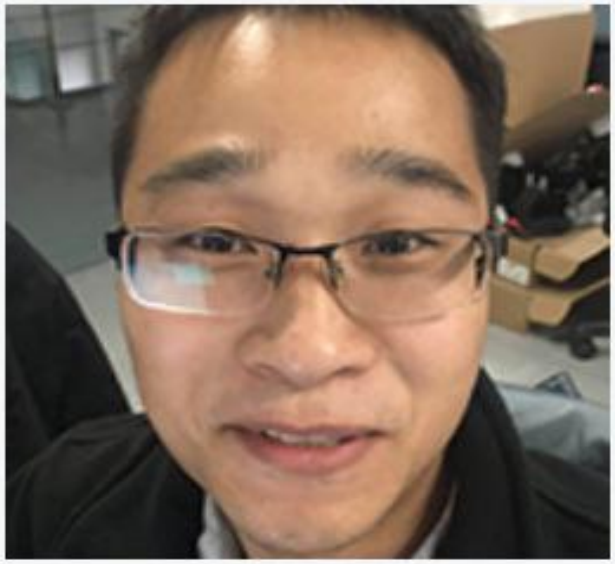
上位机


Model
Batch
Select
Send
Compare
请选择图片
Model Selected:
Compute Time:
Compare Result:
Distance:

上位机


Model
Batch
Select
Send
Compare
Complete
Compute Complete!
OK
Model Selected: 20170512-110547
Compute Time: 173 ms
Compare Result:
Distance:

上位机



Model

Batch

Select

Send

Compare

Model Selected: 20170512-110547
Compute Time: 175 ms
Compare Result: BaiLifei
Distance:1.74195e-06

总结与展望

耦合性

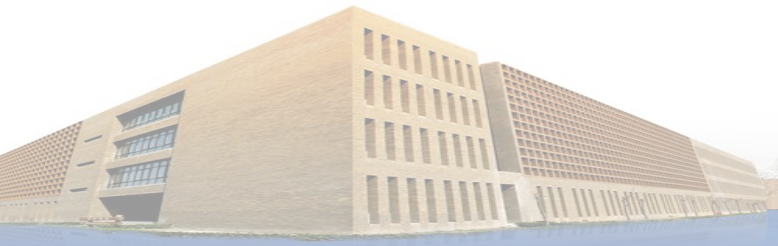
耦合

结构

已完成：固定网络结构

缺　点：网络结构单一

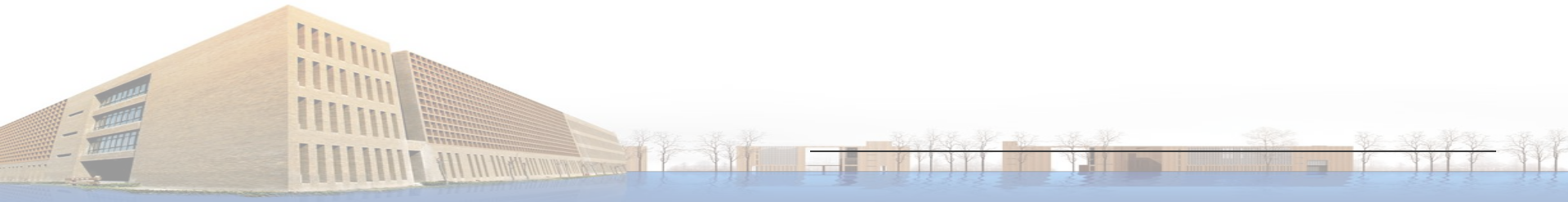
进一步改进：可以通过在线下载可配置文件更改网络



进一步改进

自动代码生成

构建完整的系统，脱离FPGA的限制，通过使用自动工具生成FPGA可配置文件，供FPGA直接使用。



谢　　谢！

欢迎各位老师同学指导！