

B0911007Y-01/02 2021-2022学年春季学期

计算机组成原理（研讨课）

实验项目0 问题解答

2022年3月4日



中国科学院大学
University of Chinese Academy of Sciences



中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY, CAS

问题1：设置了GitLab免密操作，但是不起作用

❑ 原因：PPT提供的远程仓库地址有错误

❑ 解决方法：依次执行下面两条命令

```
cd ~/COD-Lab && git remote set-url origin ssh://git@gitlab.agileserve.org.cn:8082/ucas-cod-2022/GitLab用户名.git
```

```
cd ~/COD-Lab && git remote set-url upstream ssh://git@gitlab.agileserve.org.cn:8082/ucas-cod-2021-dev/cod-lab.git
```

注意：请务必替换其中第一条命令的**GitLab**用户名

注意：第二条命令中的**2021**没有错误

问题2：虚拟机中没有安装vim



□ 编辑文件时，出现如下内容

```
ucas@ucas-cod-2022:~/COD-Lab$ cd ~/COD-Lab && vim fpga/design/ucas-cod/hardware/
sources/example/adder.v

Command 'vim' not found, but can be installed with:

sudo apt install vim          # version 2:8.1.2269-1ubuntu5.7, or
sudo apt install vim-tiny     # version 2:8.1.2269-1ubuntu5.7
sudo apt install vim-athena   # version 2:8.1.2269-1ubuntu5.7
sudo apt install vim-gtk3     # version 2:8.1.2269-1ubuntu5.7
sudo apt install vim-nox      # version 2:8.1.2269-1ubuntu5.7
sudo apt install neovim       # version 0.4.3-3

ucas@ucas-cod-2022:~/COD-Lab$
```

□ 解决办法：请执行 `sudo apt install -f vim` 后，再编辑文件

问题3: Access Token设置错误导致无法查看波形

□ 打开波形文件时，出现如下错误

```
ucas@ucas-cod-2022:~/COD-Lab$ cd ~/COD-Lab && make FPGA_PRJ=ucas-cod FPGA_BD=nf SIM_TARGET=example wav_chk
/usr/bin/curl
/usr/bin/jq
project_3689_bot
jq: error (at <stdin>:0): Cannot index object with number
Error: Pipeline not found, please check CI/CD pipeline on GitLab.
make: *** [fpga/design/ucas-cod/scripts/hardware.mk:38: wav_chk] Error 1
ucas@ucas-cod-2022:~/COD-Lab$
```

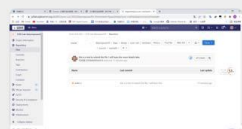
□ 解决办法：务必设置用户的Token，切勿设置Project的Token

问题4：不太“认识” FPGA



请问大家的仿真需要多久？

已经十分钟了还没有仿真完成



还是说没有仿真成功

你可以点进去看到哪步了



代永宣 1班：还是说没有仿真成功

到 bit-gen

对实验流程没有理解，认为都是仿真过程

实验一：4位加法器行为建模
实验一附加实验一：1位全加器行为建模与结构建模
实验一附加实验二：generate语句实现4bit全加器

实验二：4bit比较器串行实现16bit比较器
实验二附加实验一：4-16译码器
实验二附加实验二：4bit先行进位加法器

实验三：状态机实现序列检测
实验三附加实验：电梯设计（三楼层，电梯内外按键）

实验四：8写4读FIFO（全读全写）
实验四附加实验：8写4读FIFO（随读随写）

评分规则：
主体实验在课上及课后验收，附加实验只能在课堂上验收
实验一：90%主体+5%附加1+5%附加2
实验二：75%主体+5%附加1+20%附加2
实验三：90%主体+10%附加
实验四：80%主体+20%附加

用什么工具呢？只仿真？



vivado 只仿真 部分实验
老师会给 template 然后
比对结果 有些实验则需要
自行编写 testbench

前序课程实验重点关注逻辑行为仿真

B0911007Y-01/02 2021-2022学年春季学期

计算机组成原理（研讨课）

E01 FPGA基础知识

2022年3月4日



中国科学院大学
University of Chinese Academy of Sciences

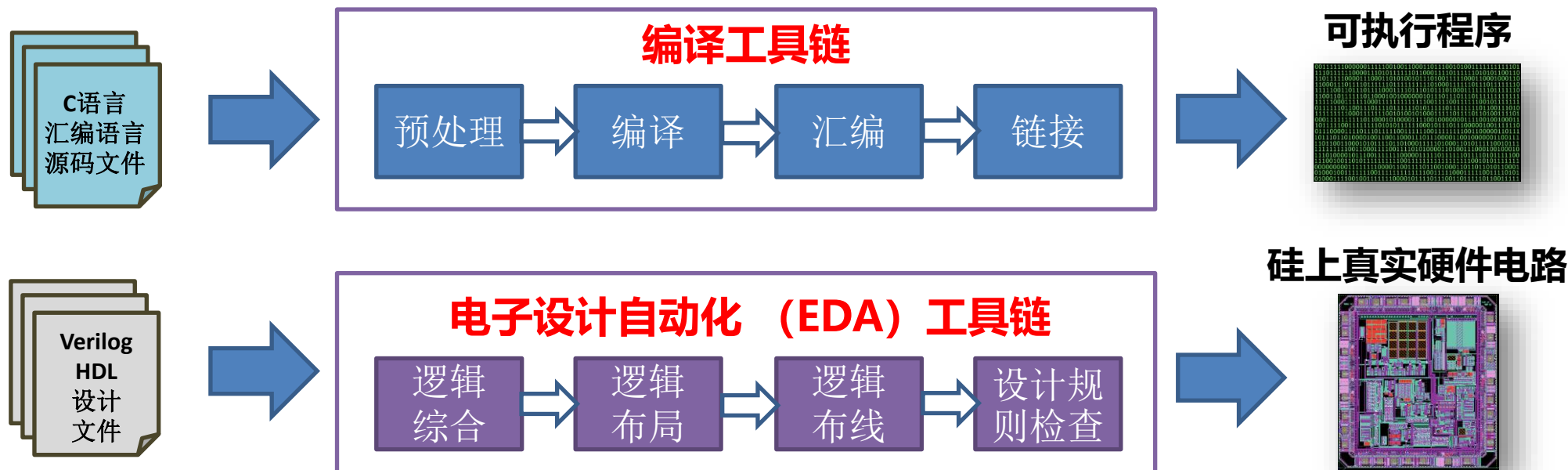


中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY, CAS

Verilog HDL硬件描述语言



❑ Verilog HDL不是编程语言，是一种硬件逻辑电路的描述语言



- ❑ 构建电路比编译可执行程序要慢很多，出错成本也大很多
- ❑ 写Verilog HDL：依照设计的电路结构，用对应的代码结构描述
- ❑ 不能用写软件程序的思维方式来描述电路
 - 软件程序（模块串行）：函数执行顺序有先后
 - 硬件电路（模块并行）：所有电路结构在上电后同时工作

□ 组合逻辑电路 Combinational logic（各类逻辑门）

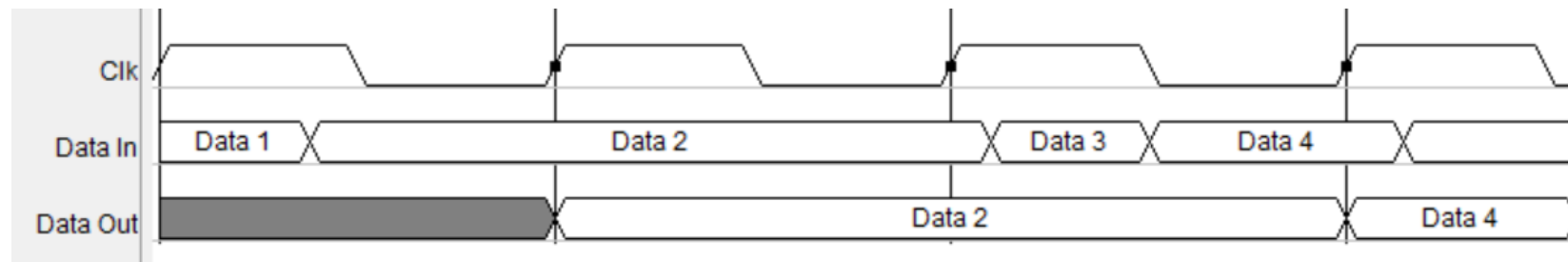
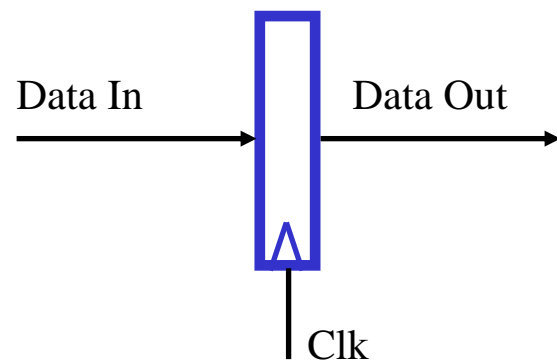
- 输出信号的变化仅仅与输入信号的电平有关，不涉及对信号跳变沿的处理
- Once the input arrives, the combinational logic starts processing it to make outputs

□ 时序逻辑电路 Sequential logic（D触发器）

- 不管输入如何变化，仅当时钟的边沿（一般为上升沿）到达时，才有可能使输出发生变化
- Once the input arrives, you have to wait until the next positive or negative clock edge, before the input can be read in and processed

□ 1-bit的D触发器

- Data Out的值取决于时钟上升沿到来时Data In的输入

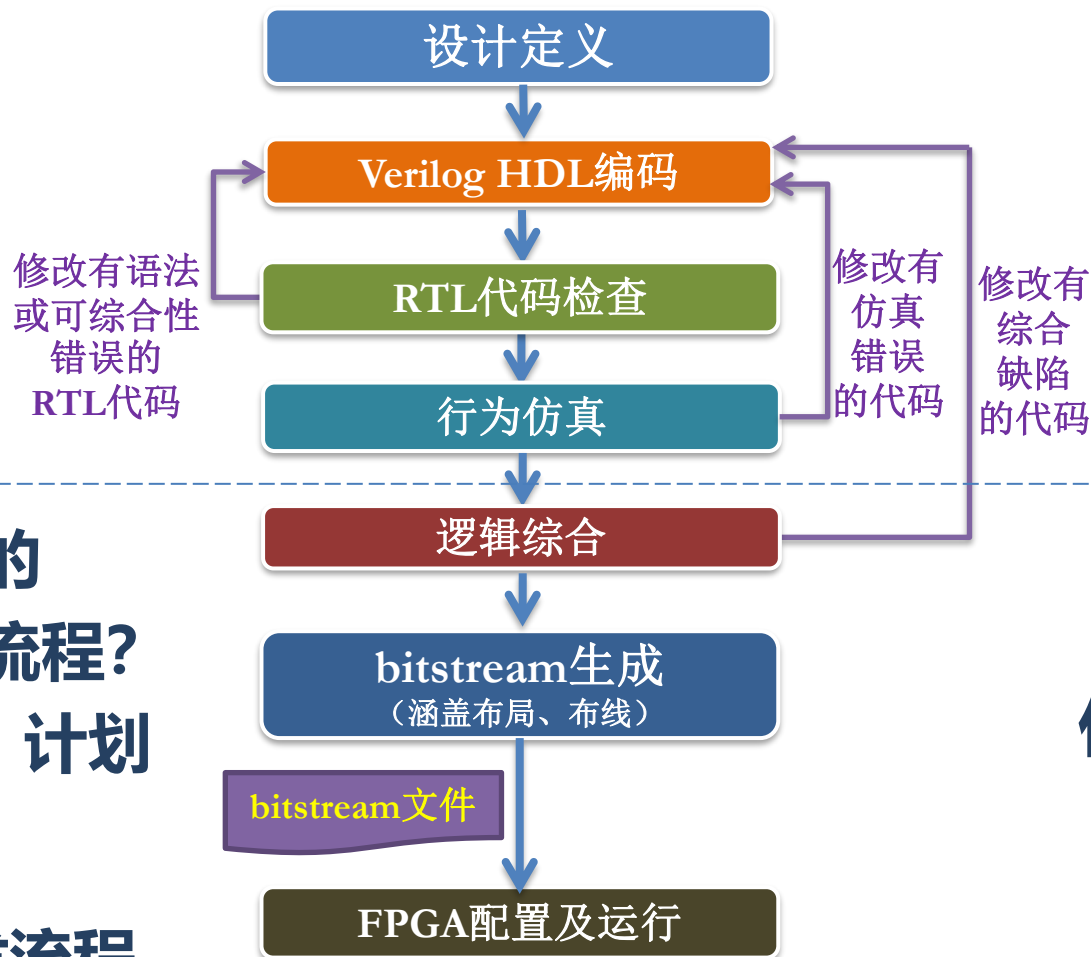


数字逻辑电路的基本开发流程



前序课程
以**逻辑行为仿真**为主
重点关注Verilog HDL
的语法和逻辑实现

面向芯片设计的
数字逻辑电路开发流程?
—— “一生一芯” 计划
面向FPGA的
数字逻辑电路开发流程



代码文本
↓
硅上真实电路

什么是FPGA

什么是FPGA?



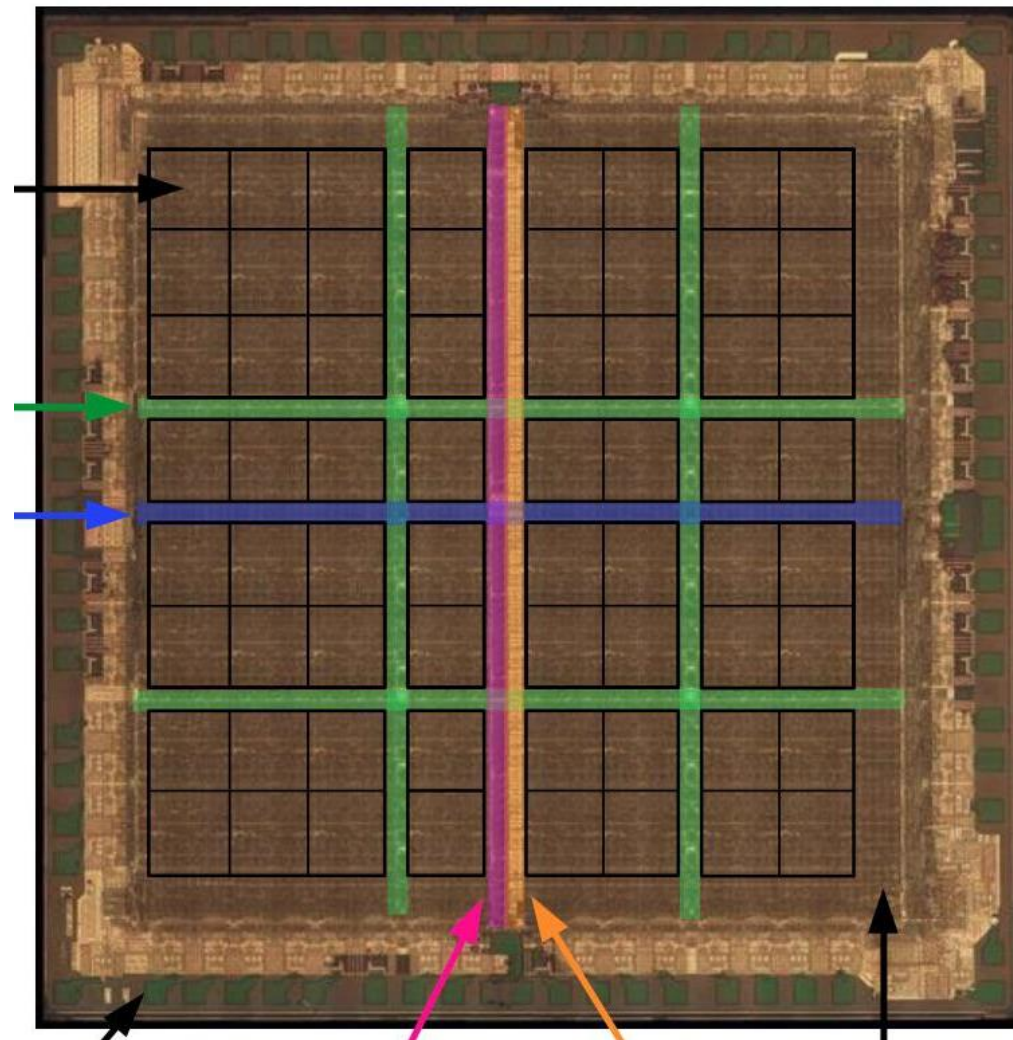
大规模的可重配置的硬件逻辑电路

The World's First FPGA Chip – Xilinx XC2064, 1985

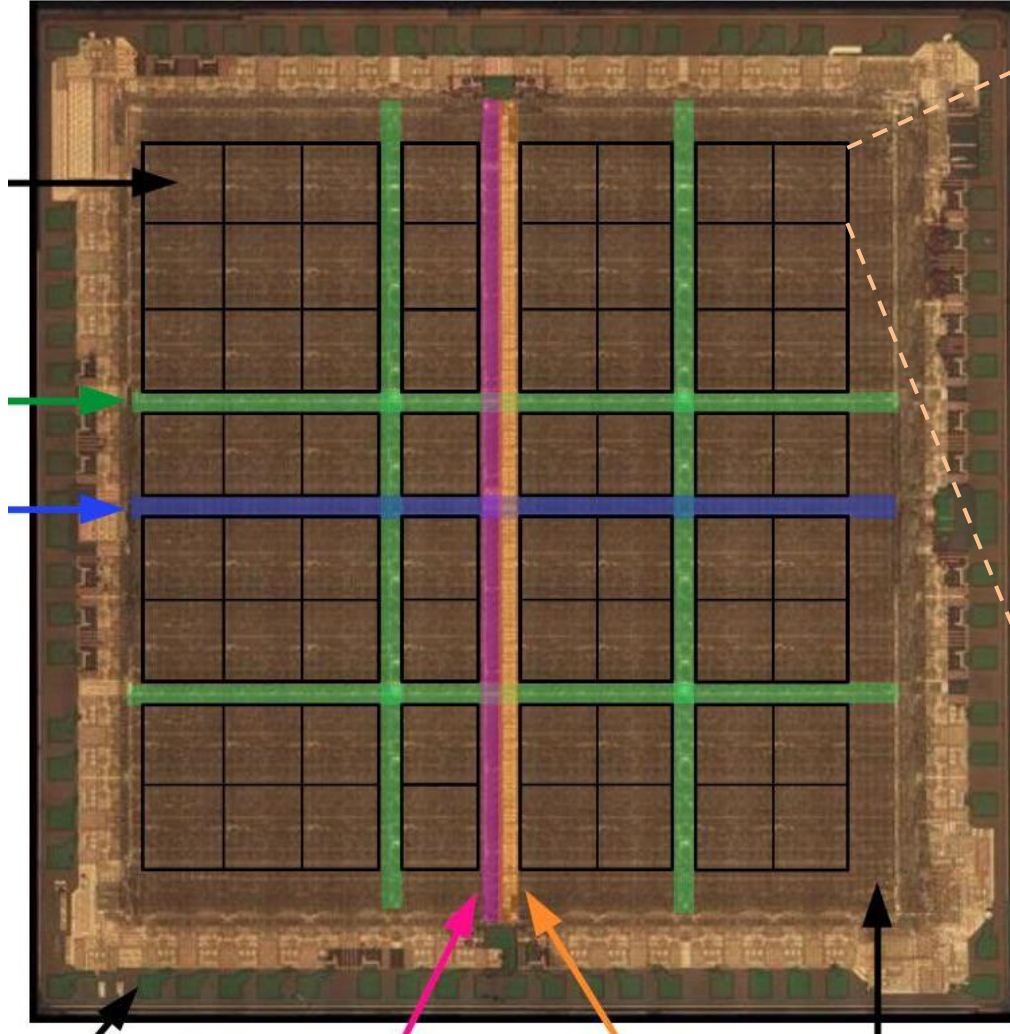


图片来源: <http://www.righto.com/2020/09/reverse-engineering-first-fpga-chip.html>

Xilinx（赛灵思公司）：全球领先的可编程逻辑完整解决方案的供应商，研发、制造并销售范围广泛的高级集成电路、软件设计工具以及作为预定义系统级功能的IP（Intellectual Property）核。**已于2022年被AMD公司收购**



The World's First FPGA Chip – Xilinx XC2064, 1985



□ 可配置逻辑块

(Configurable Logic Block, CLB)

- 分别基于查找表 (Look Up Table, LUT) 和D触发器 (D Flip-Flop, 简称FF) 实现组合逻辑和时序逻辑
- 提供芯片内的内存 (Memory) 资源

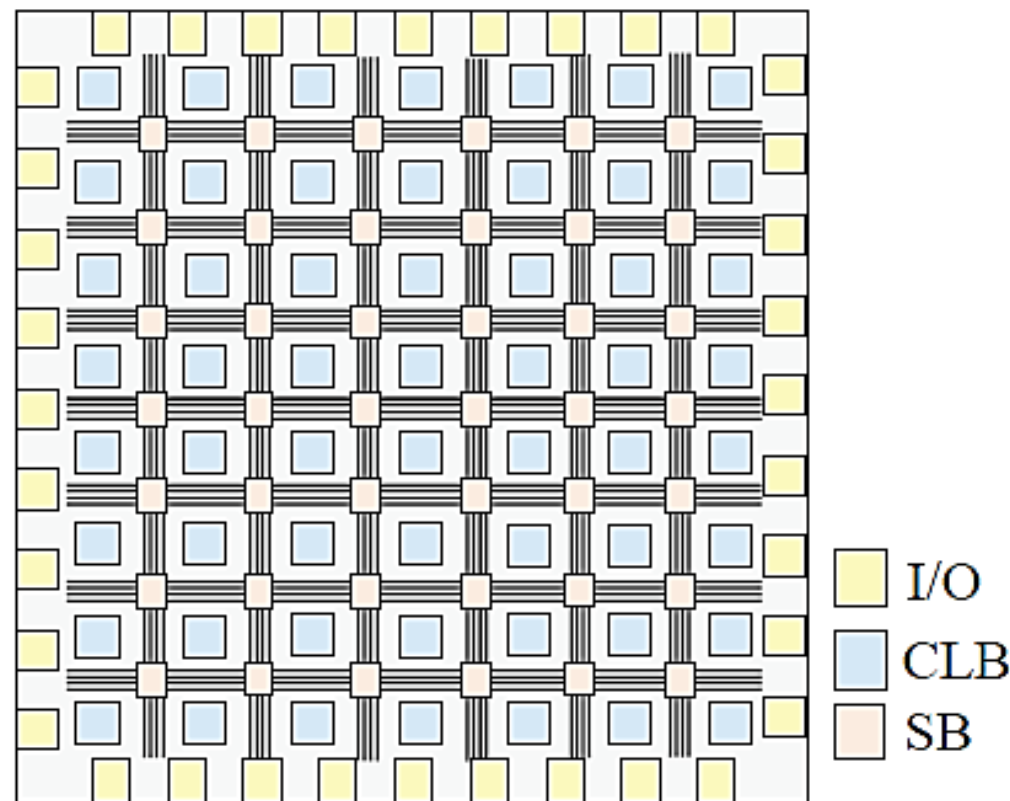
□ 可配置输入输出缓冲

(I/O Buffer)

- 设置某一引脚的方向 (in/out/三态)、延迟、输出电压等属性

□ 可配置互连开关 (Switch Box, SB)

- 将众多可编程逻辑块与I/O引脚连接起来
- 为时序逻辑提供时钟

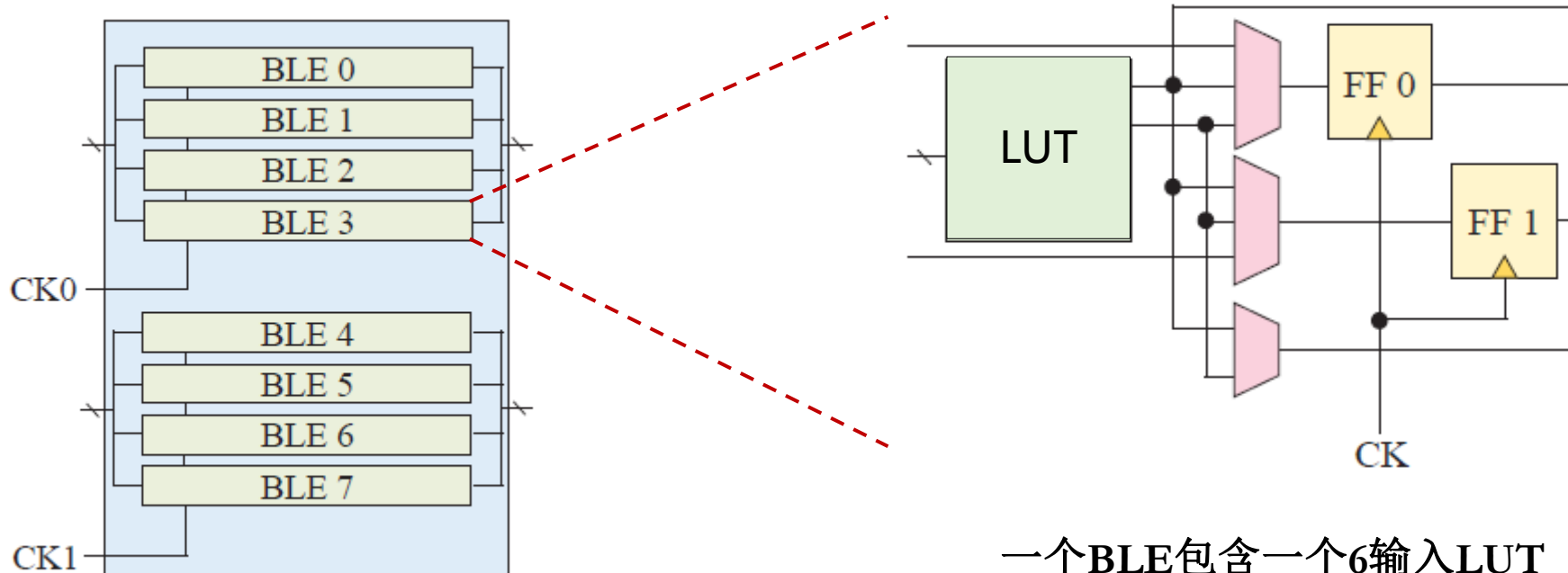


图片来源: S. Chen and Y. Chang. FPGA Placement and Routing. ICCAD'17

可编程逻辑块CLB



以实验课使用的赛灵思（Xilinx）UltraScale+系列FPGA为例



一个CLB包含8个
基本逻辑单元BLE

一个BLE包含一个6输入LUT
和两个触发器FF
(6输入LUT也可被配置成两个5输入LUT)

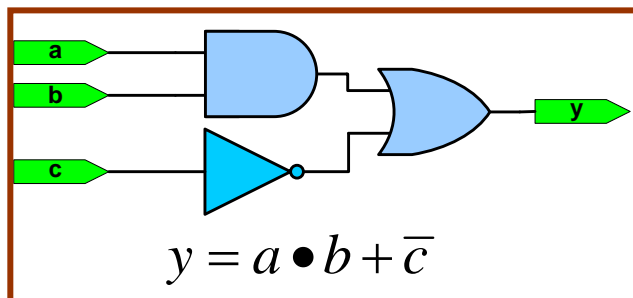
图片来源：S. Chen and Y. Chang. FPGA Placement and Routing. ICCAD'17

□ FPGA中实现各种组合逻辑的核心单元

- LUT的输入信号用来生成查找地址，读出LUT内部真值表的值
- LUT的真值表在FPGA配置时填充好

举例：一个3输入LUT

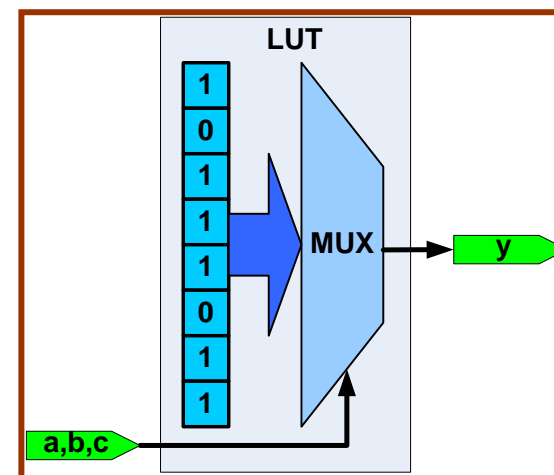
逻辑表达式



真值表

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

配置好的查找表

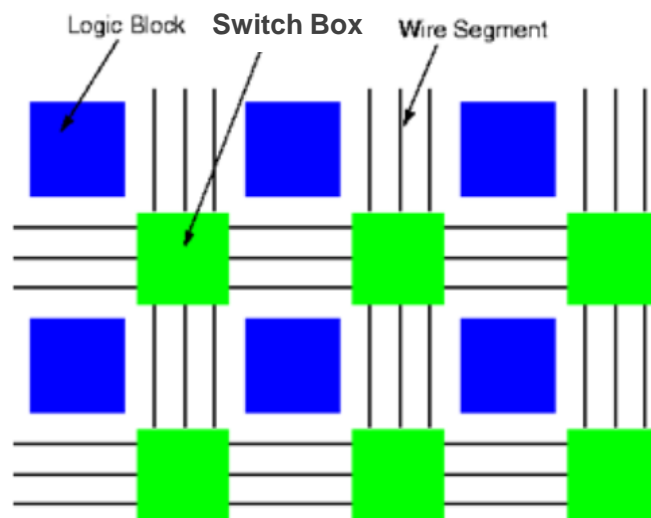


□ 绕线（routing wires）资源

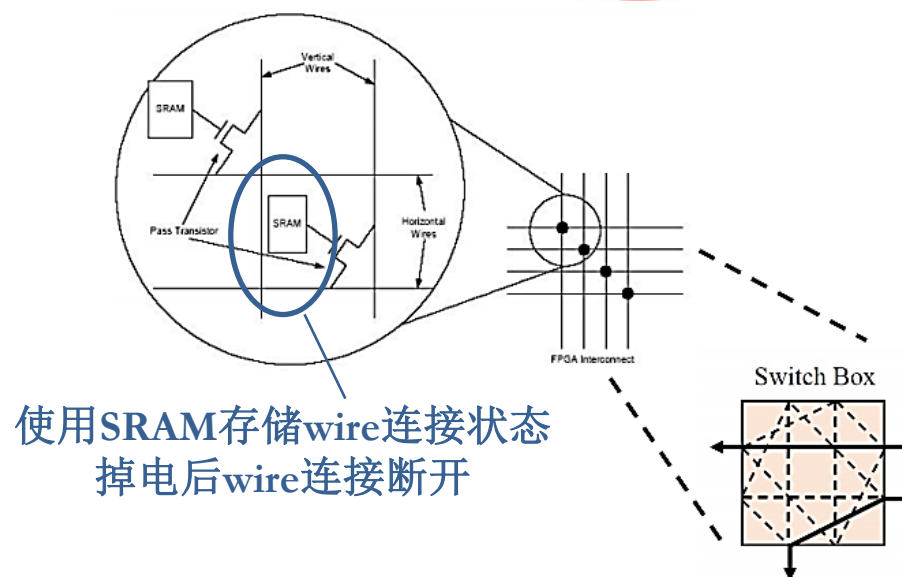
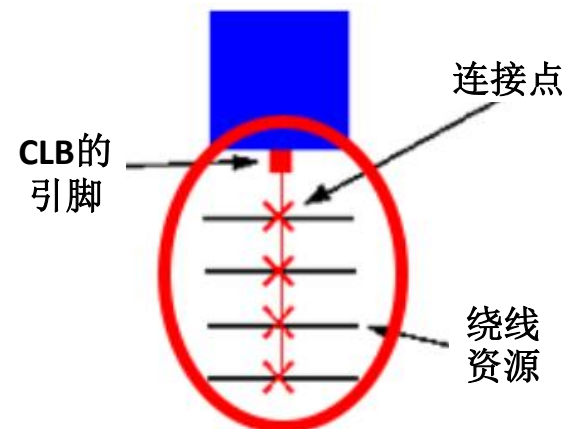
- 一个CLB引脚可同时连接到任意数量的绕线资源上

□ 互连开关（Switch Box）

- 通过编程配置，决定CLB之间绕线的连接关系



图片来源：英国帝国理工大学Digital Electronics
2 (E 2.1) by Prof. Peter Cheung



图片来源：Jakub Szefer. Cloud FPGA. EENG428, Yale Univ. Fall 2019

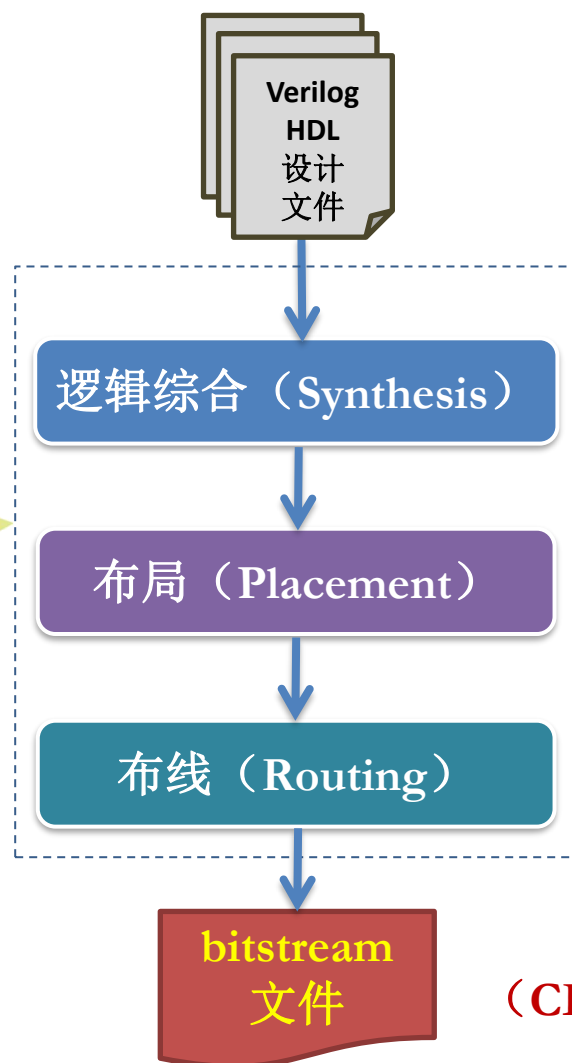
□一个FPGA芯片中有多少LUT?

- FPGA厂商为满足不同类型应用场景需要，会在同一系列FPGA中生产多种容量不同的芯片型号
- 以实验课使用的Xilinx Zynq UltraScale+ MPSoC系列FPGA器件为例
 - 总体LUT的规模为518K
 - 为同学们划分的每个逻辑区中，LUT规模为29.7K

如何使用这些资源？

同学们写的Verilog HDL和这些资源有什么关系？

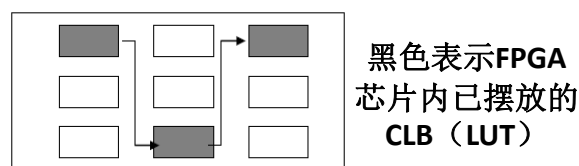
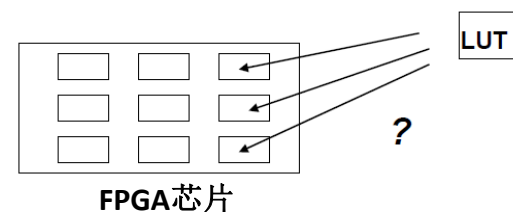
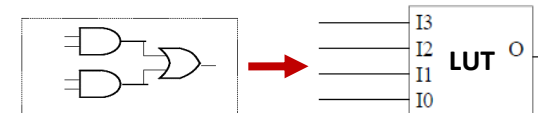
FPGA开发流程



先将Verilog HDL转换为
门级网表 (Netlist)
再将逻辑门网表
转换为CLB级网表

根据芯片型号将网表
中CLB摆放在FPGA
中的某个对应资源上

利用可编程开关
完成FPGA芯片内
已摆放CLB输入
输出引脚的连线

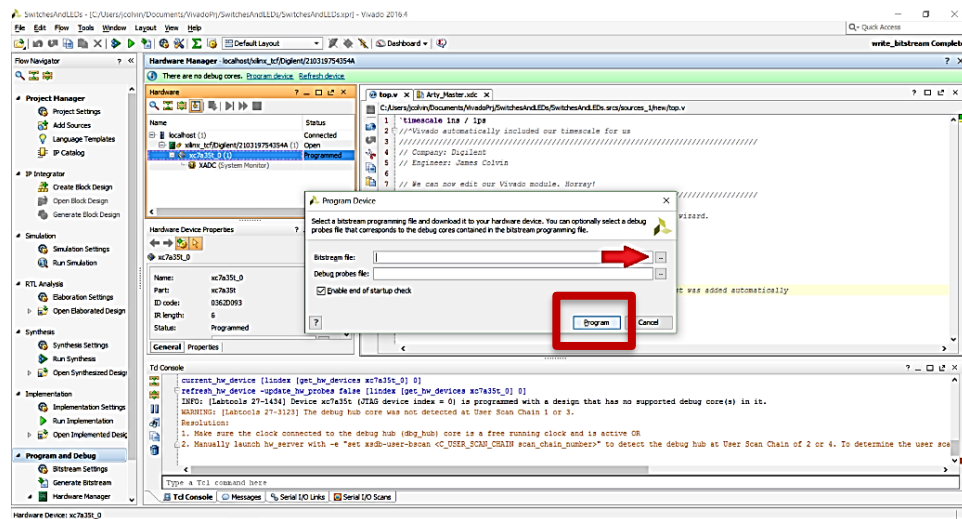


FPGA配置文件
(CLB配置、LUT真值表、互连开关状态等)

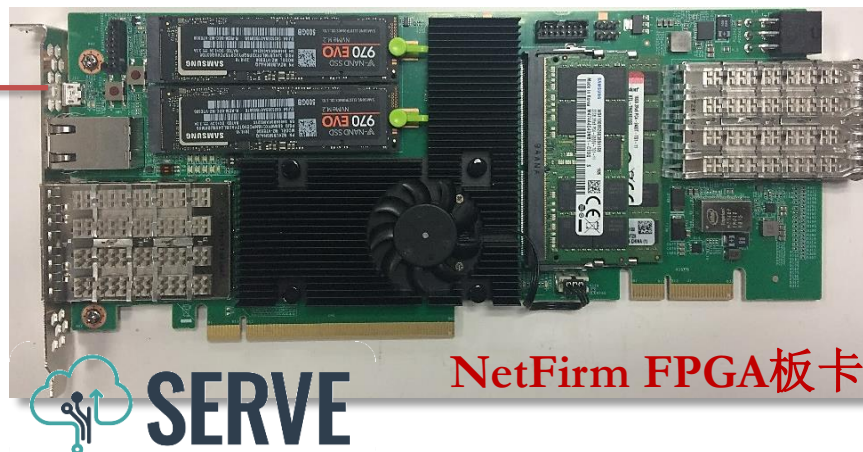
将bitstream配置到FPGA中



- 用笔记本电脑连接FPGA板卡
- 在笔记本电脑上打开Vivado图形界面，扫描目标板卡，选择生成bitstream文件，点击<Program>按钮（红框），开始FPGA配置



USB-JTAG线



配置完成
设计的电路就
开始正常工作

FPGA背景知识及发展现状阅读材料



FPGA发展趋势 (1)

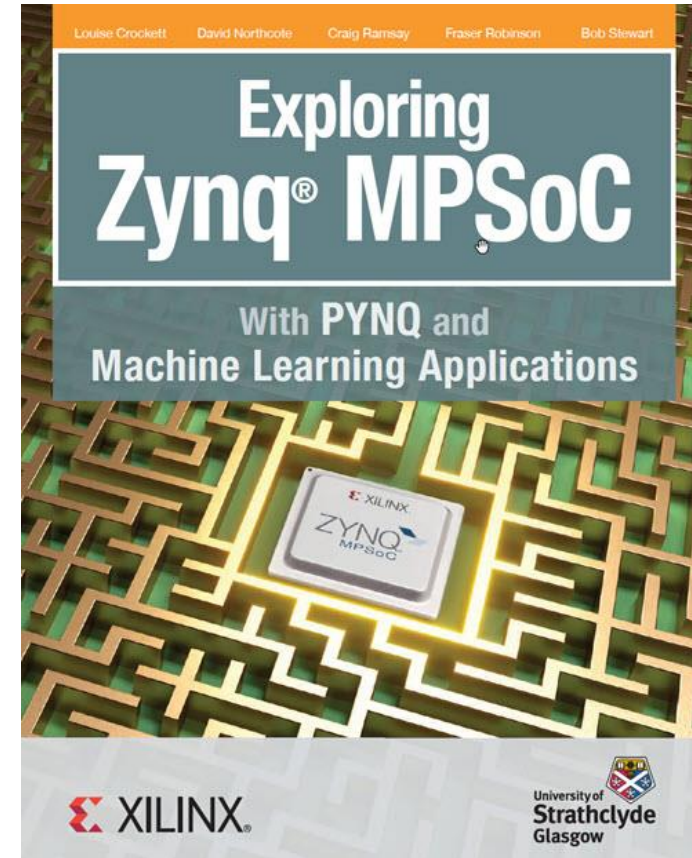
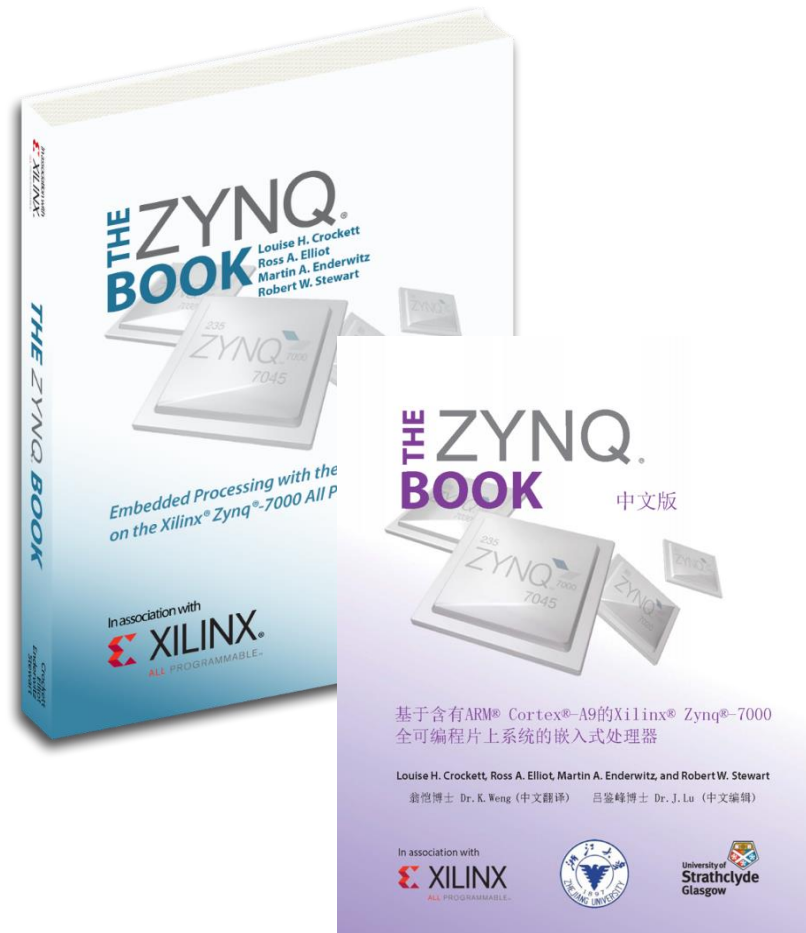


- 2011年Xilinx推出**Zynq**系列器件，首次把FPGA逻辑资源与**基于ARM处理器的硬核（Hard Core）片上系统（SoC）**封装在一个芯片内
- 2015年Xilinx发布**Zynq MPSoC**系列器件（本学期实验课使用的器件），除升级ARM处理器外，进一步集成GPU、实时处理器RPU、视频处理加速器等计算资源

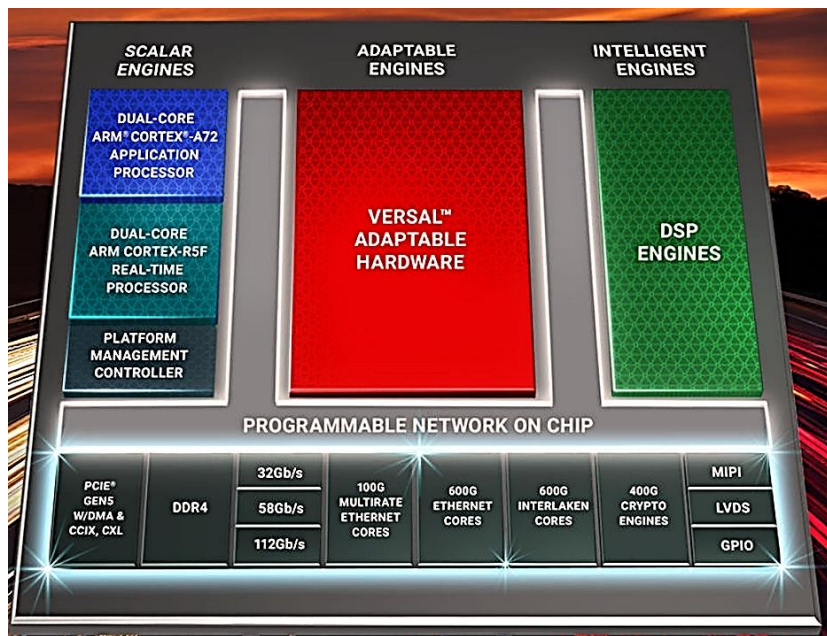
FPGA器件开始具备软-硬件协同可编程能力

The Zynq Book & Exploring Zynq MPSoC Book

<http://www.zynqbook.com/> & <https://www.zynq-mpsoc-book.com/>



Xilinx Versal™ Adaptive Compute Acceleration Platform (ACAP)



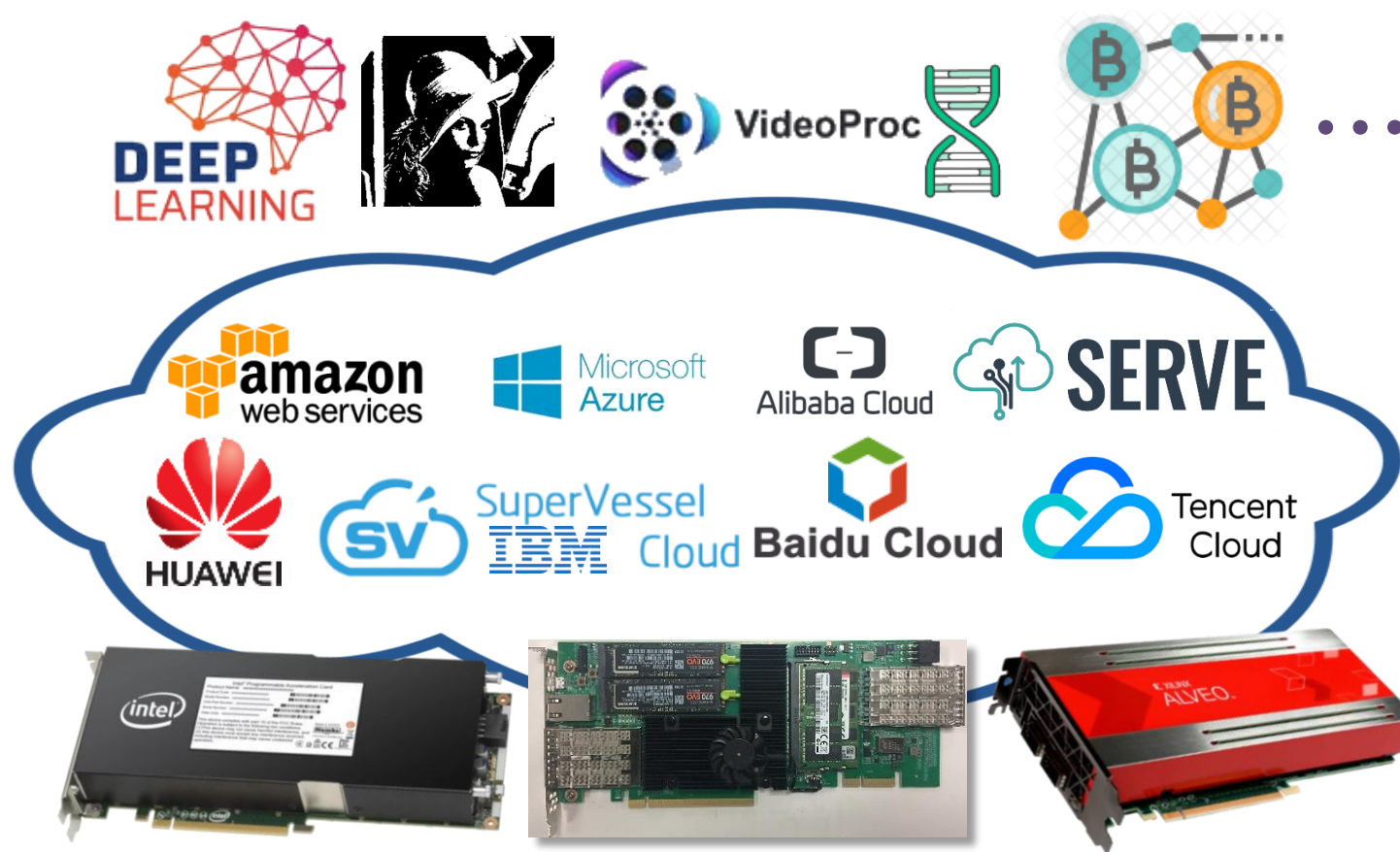
- Xilinx公司2018年推出^[1]
- 通过可编程片上网络（Network-on-Chip），集成各类通用处理单元（Scalar engines）、智能处理单元（Intelligent engines）、PCIe、新型内存HBM及各类高速I/O接口
- 不局限于传统FPGA细粒度LUT及寄存器的配置能力，**更加强调软件可编程和I/O接口可配置**

粗粒度可重构阵列CGRA (Coarse-Grained Reconfigurable Architecture)^[2]

[1] https://www.xilinx.com/support/documentation/white_papers/wp505-versal-acap.pdf

[2] L. Liu and et.al, A Survey of Coarse-Grained Reconfigurable Architecture and Design: Taxonomy, Challenges, and Applications. ACM Computing Surveys, 2019

FPGA发展趋势 (3)



众多云服务商为用户提供远程FPGA资源

- 用户无需手工连接FPGA物理板卡
- 隐藏FPGA板卡细节
- 为应用提供特定的编程框架
- 提供便捷的FPGA部署和使用方法

云是降低FPGA使用门槛的一种重要方式

如何在课程使用的SERVE云上部署使用自己的逻辑电路？

```
Runtime platform
Downloading artifacts from coordinator... ok
Executing "step_script" stage of the job script
$ make ci run
Completed FPGA configuration
FPGA evaluation succeeded.
Uploading artifacts for successful job
Uploading artifacts...
Runtime platform
run/log/: found 2 matching files and directories
Uploading artifacts as "archive" to coordinator..
Cleaning up file based variables
```

Fpga_eval

✓ fpga_eval



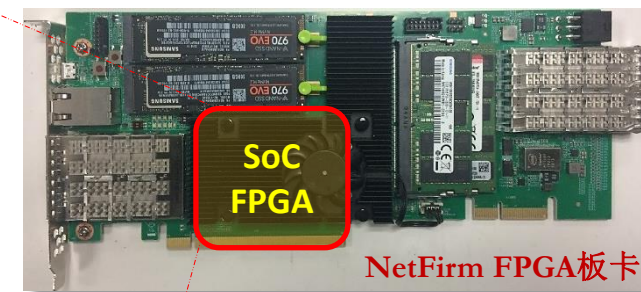
1 学生向GitLab仓库推送代码



6 学生通过GitLab网页查看运行结果

2 将生成的5个bitstream通过网络上传某块FPGA板卡的ARM处理器

5 将步骤4的结果保存到服务器



3 选择一块逻辑区域的bitstream由ARM处理器进行配置

4 ARM上的测试程序访问已配置到逻辑区域中的8-bit加法器(设置操作数, 读取结果)



Q & A ?



中国科学院大学
University of Chinese Academy of Sciences



中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY, CAS