

# 硬件综合实验

ZJUNIX



# 目录

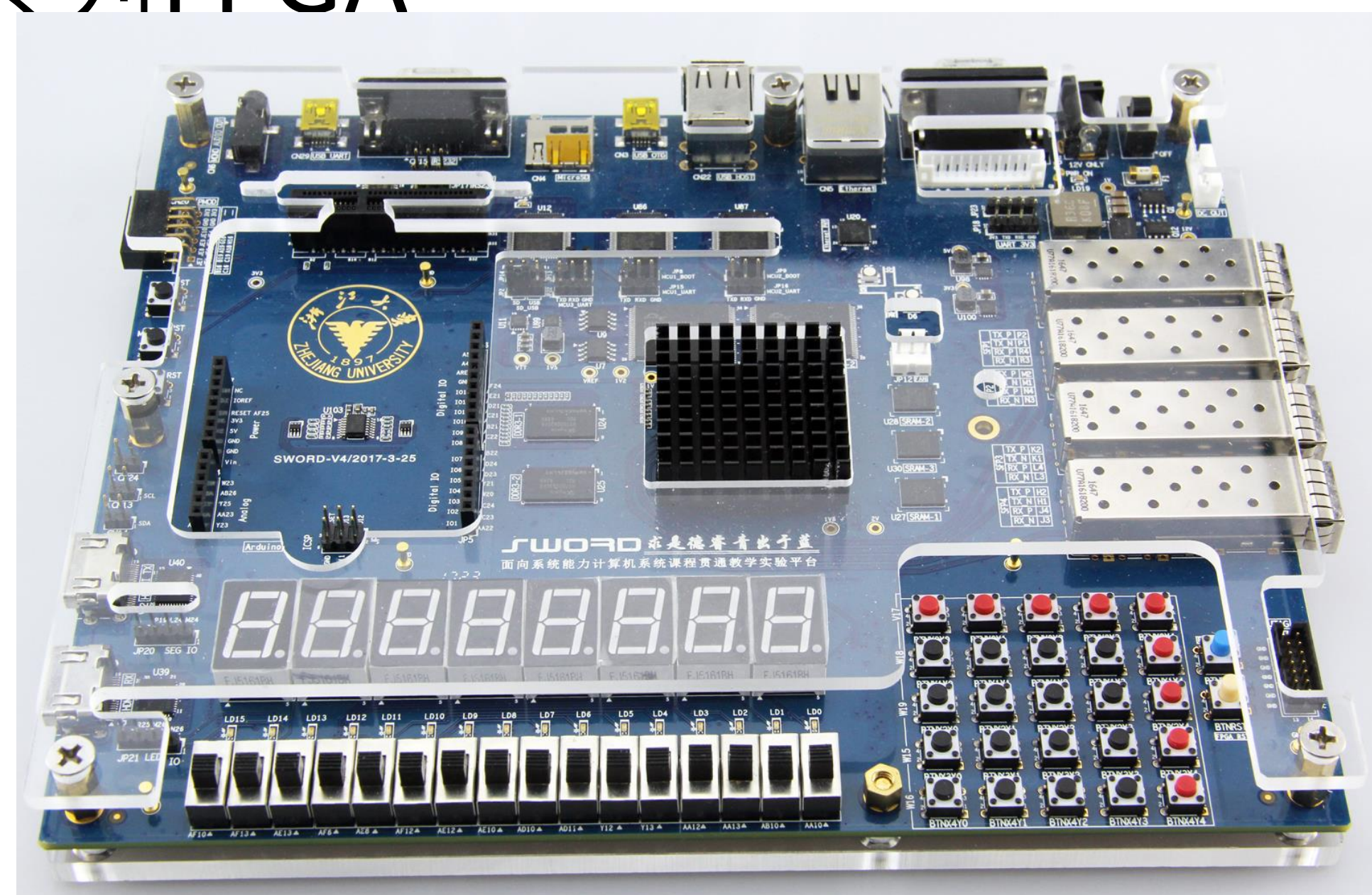
1. 实验平台简介
2. 系统架构简介
3. 实验流程简介

# 实验平台简介

- SWORD4.0 FPGA实验平台

# SWORD4.0实验平台

- SWORD4.0 是一款系统能力培养贯通教学实验平台开发板
- 核心芯片为Xilinx生产的Kintex-7系列FPGA
  - Kintex™-7 XC7K325T-2FFG676
  - 326,080 K个逻辑单元
  - 定制的6M Bytes SRAM（位宽48bit）
  - 层次化存储结构（DDR3、Flash等）
  - 各种低速、高速、超高速接口





# SWORD4.0实验平台

- 主芯片：Xilinx Kintex-7
  - XC7K325T-2FFG676
  - 326,080 K个逻辑单元
  - 16,020Kb容量的片内存储器
  - 840个DSP单元
  - 1个PCIe 2.0硬核控制器
  - 8个12.5Gbps GTX高速串行收发器
- 层次化存储结构
  - 6M Bytes SRAM（位宽48bit）
  - 512M Bytes DDR3 SDRAM（位宽32bit）
  - 32M Bytes BPI FLASH（位宽32bit）
  - 16M Bytes SPI FLASH（用户）
  - 16M Bytes SPI FLASH（固化配置文件）
  - 1路microSD卡插槽

# SWORD4.0实验平台

- 主要接口-1

- 3路串口
- 8个七段数码管
- 16个LED
- 16个拨码开关
- 5x5键盘矩阵
- 2个三色灯
- 1路12位VGA接口

- 主要接口-2

- 1路10M/100M/1000M网口
- 1路HDMI input
- 1路HDMI output
- 1路USB-OTG接口
- 2路USB转PS2接口
- 1组Arduino接口 (3.3V)
- 1组PMOD接口
- 1路JTAG接口 (14pin)
- PCIe Gen2 4 Lane, end point
- 4路10G光纤接口 (SFP+)

# SWORD4.0实验平台优点

- 支持深入学习和修改MIPS core
- 层次化存储结构更贴近实际计算机
- 提供多种IO资源
- 高性能、超高速接口可以支持更高级开发

# 实验平台简介

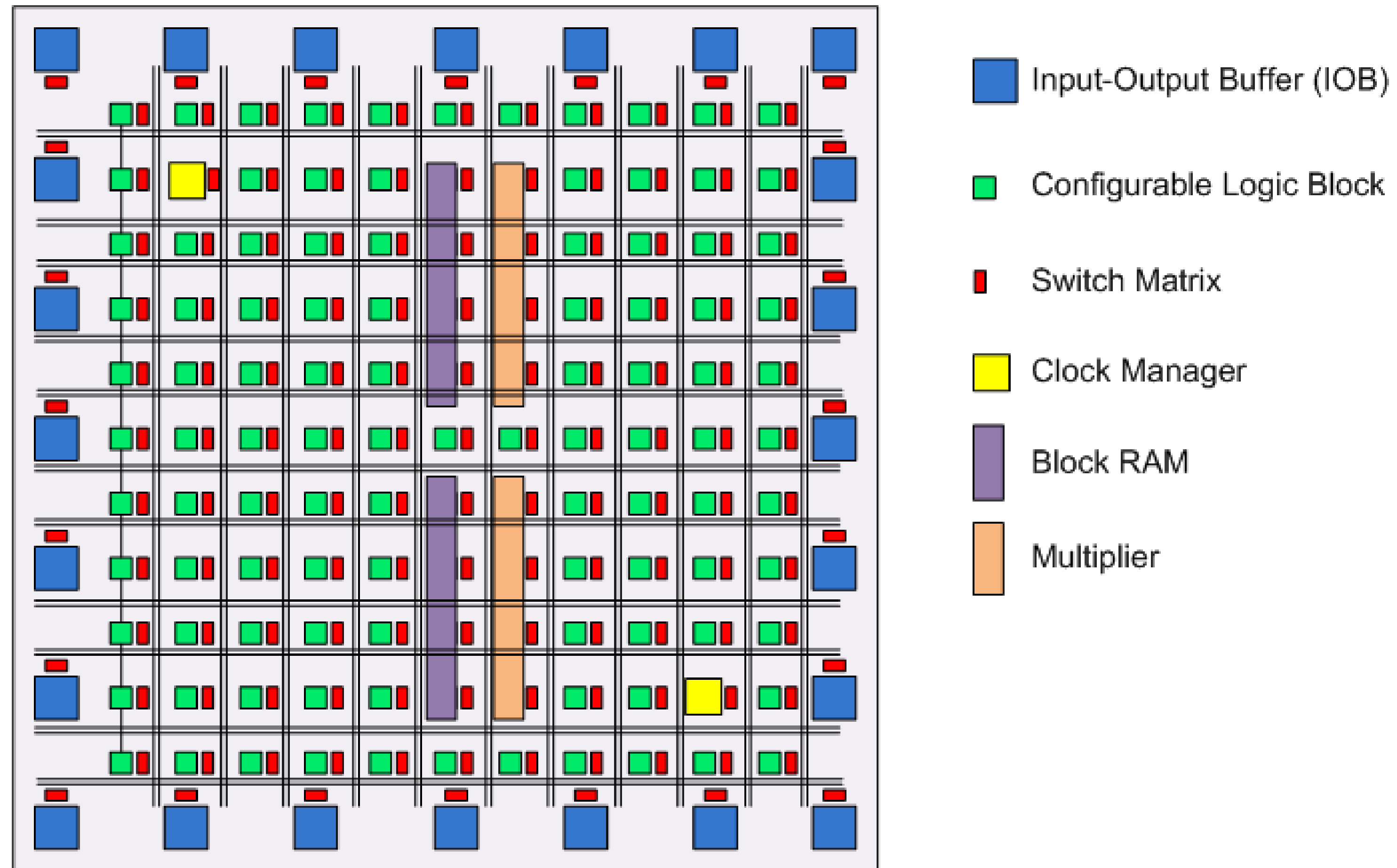
- FPGA基础知识



## 什么是FPGA

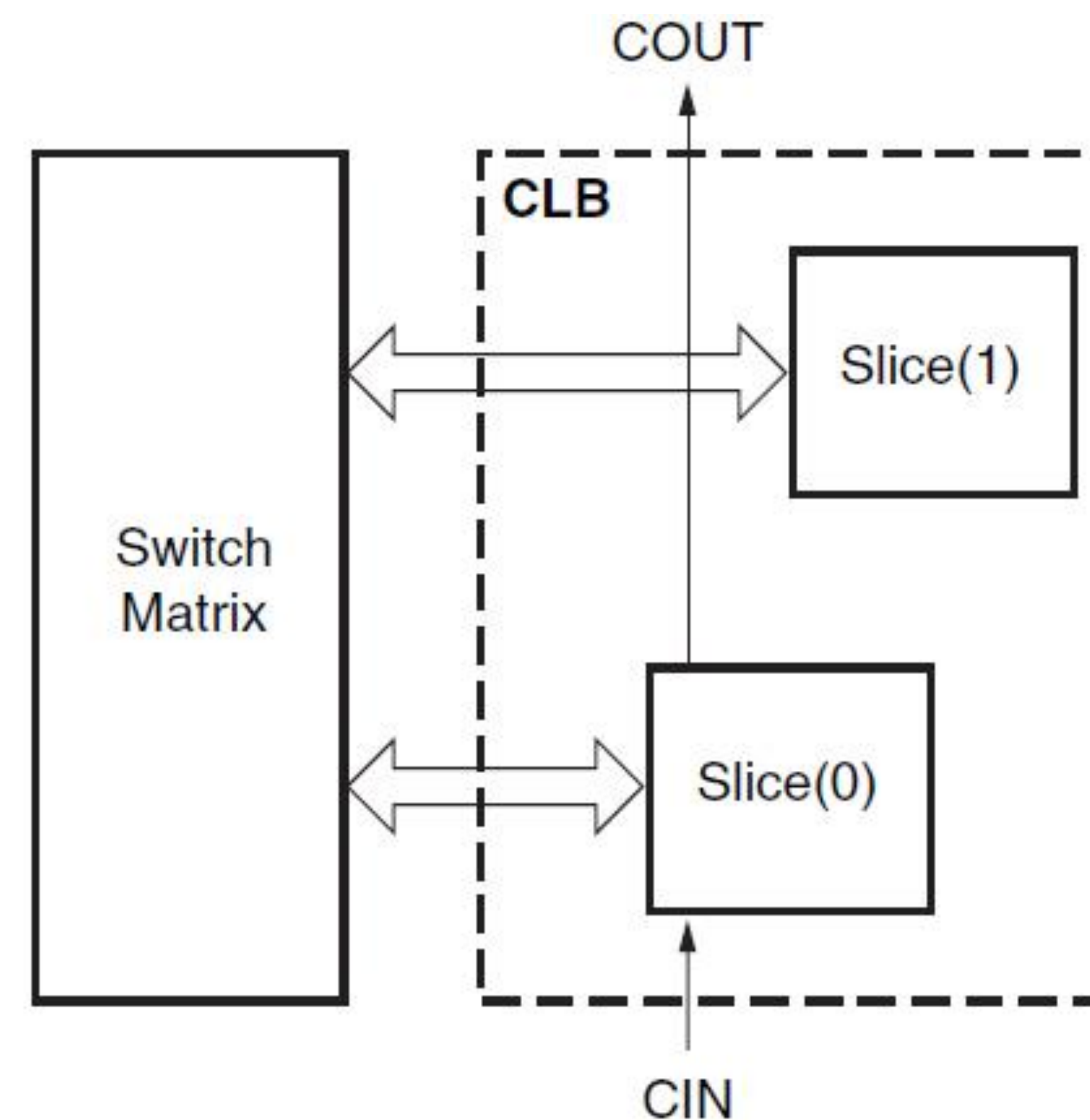
- Filed Programmable Gate Array
- 可用户编程（电路级别的编程）
- 可编程的逻辑电路， 可编程的内部互连
- 高速度
- 众多特别用途功能块（RAM、Clocks、Processors.....）
- 可编程的IO

# FPGA结构



## 可配置逻辑模块CLB

- 基础可编程单元
- 可完成各种逻辑功能
- 包含Slice、开关矩阵、触发器
- 遍布整个FPGA



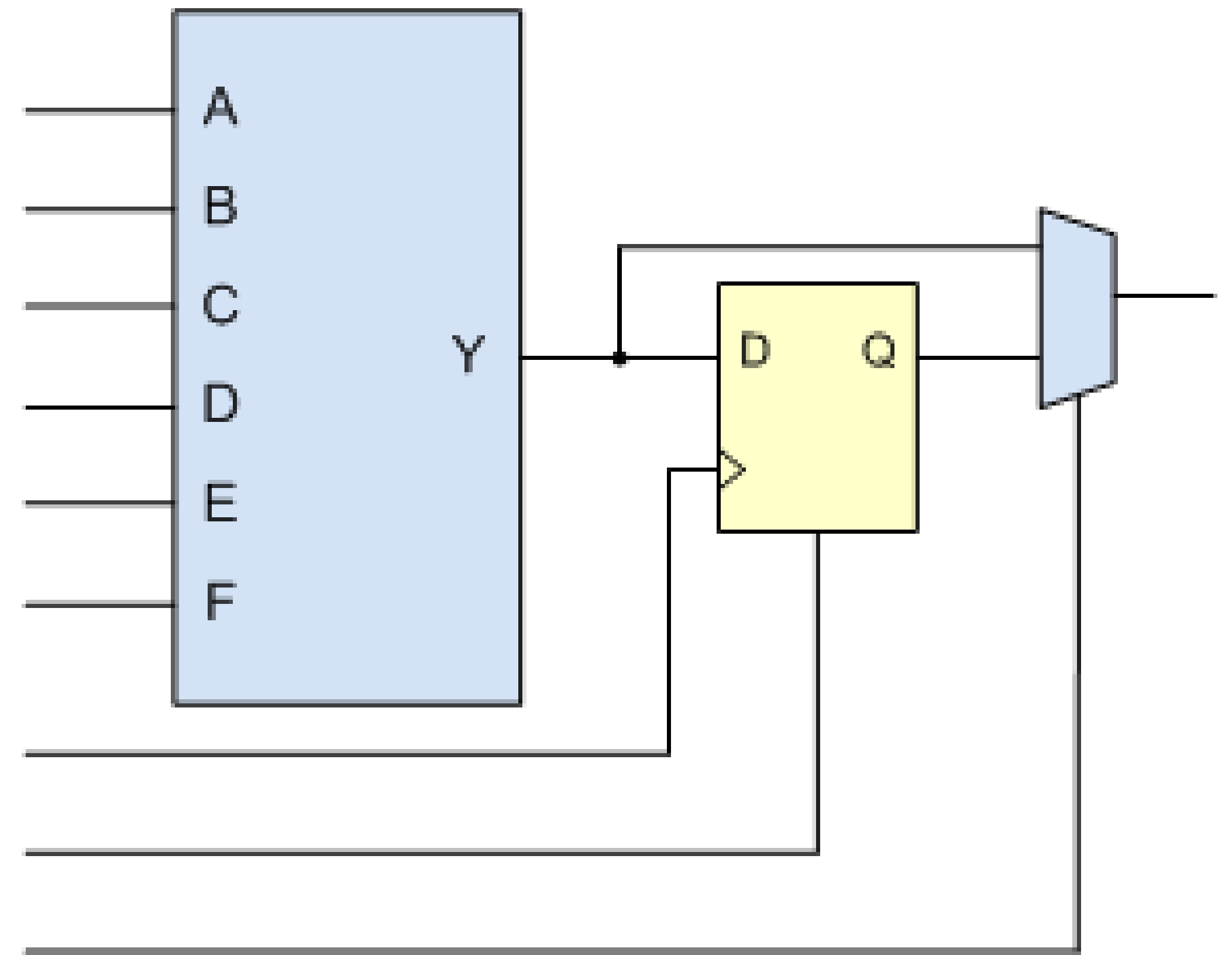
ug384\_01\_042309

Figure 1: Arrangement of Slices within the CLB

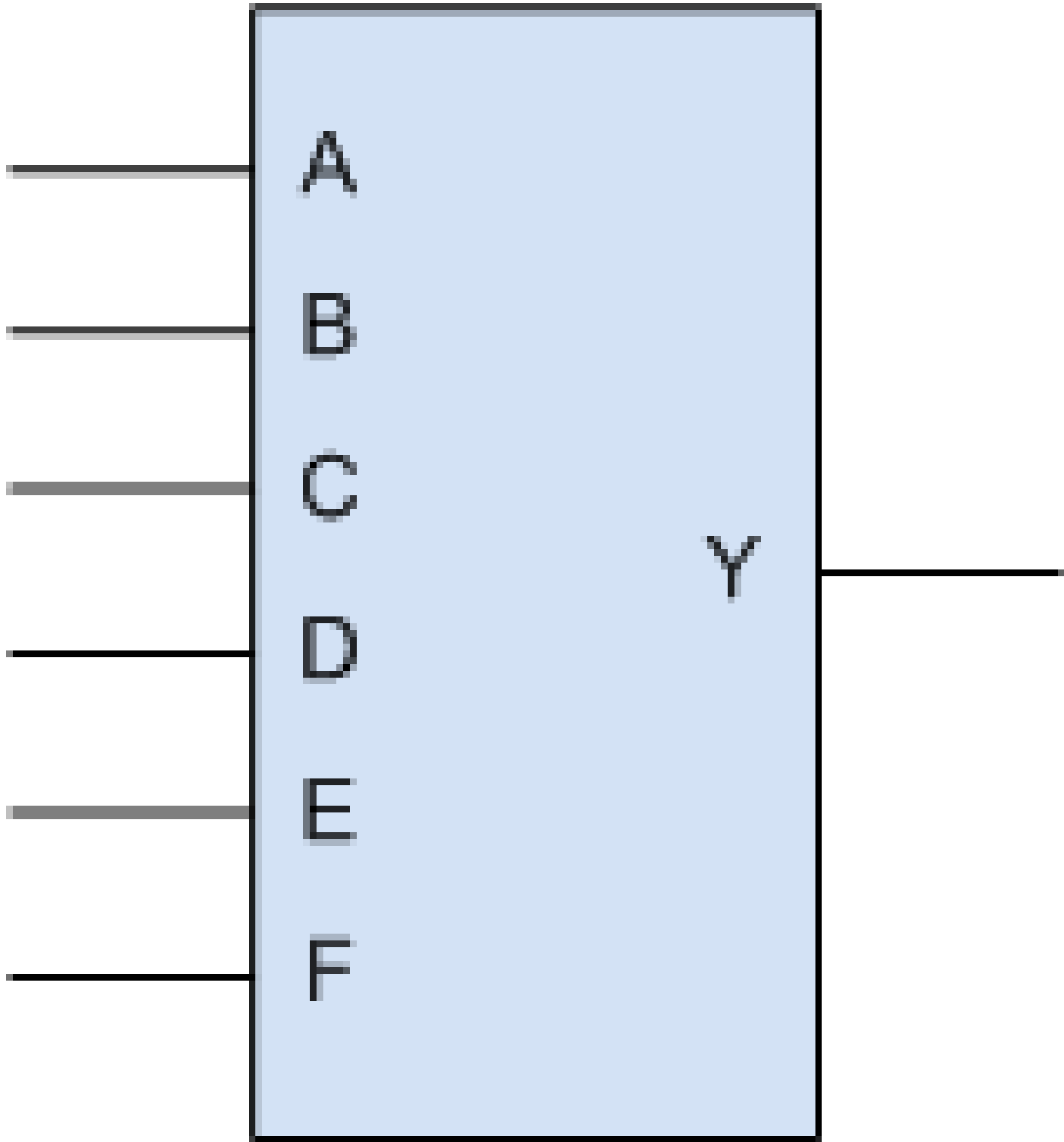


## Slices

- 使用look-up table完成各种函数功能
- LUT可以输出组合逻辑和时序逻辑



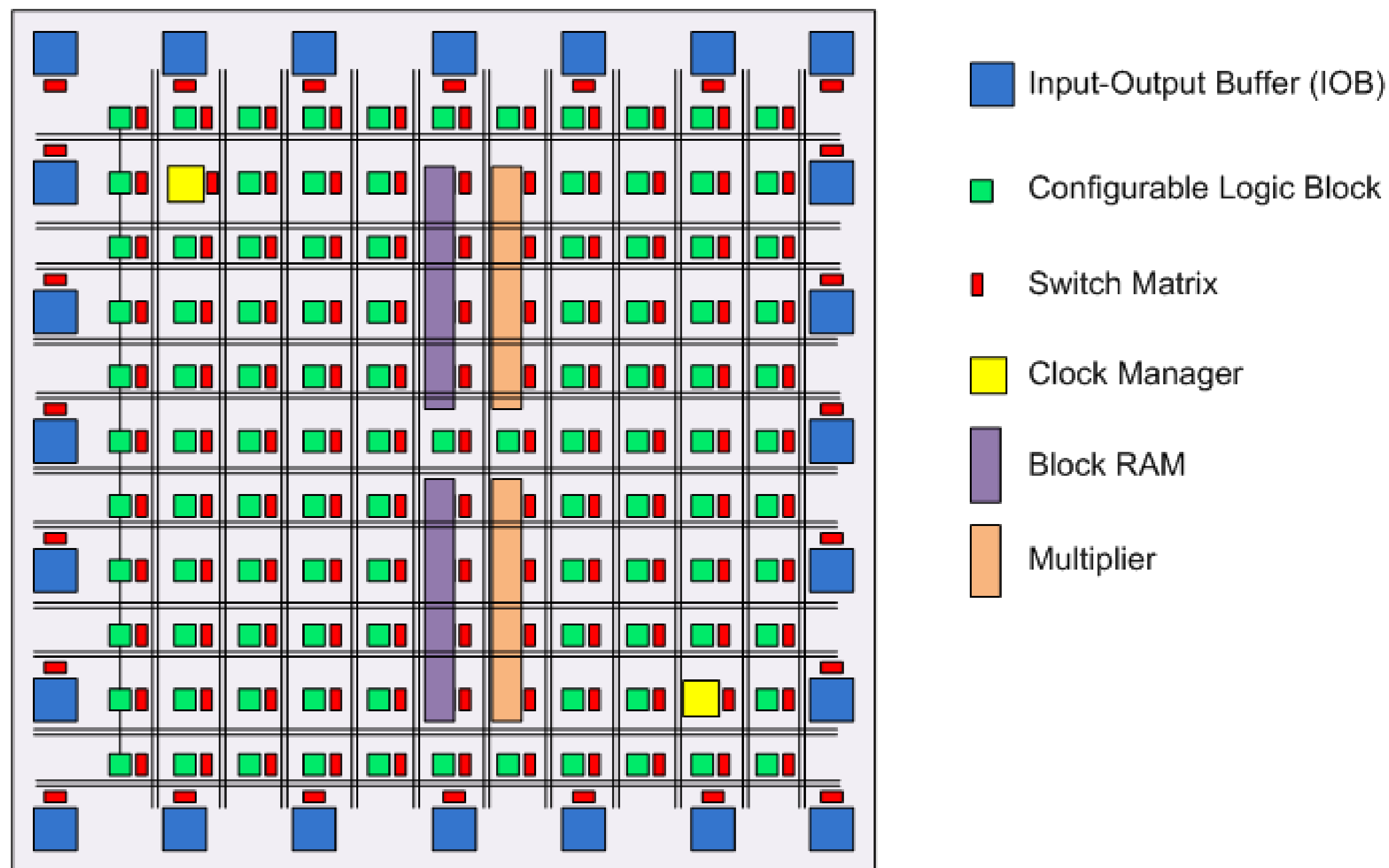
# Slices如何实现函数功能



A	B	C	D	E	F	F <sub>0</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	...	F <sub>6553</sub> 4	...	F <sub>2^6</sub> 4
0	0	0	0	0	0	0	1	0	1	0	1		0		1
0	0	0	0	0	1	0	0	1	1	0	0		1		1
0	0	0	0	1	0	0	0	0	0	1	1		1		1
0	0	0	0	1	1	0	0	0	0	0	0		1		1
0	0	0	1	0	0	0	0	0	0	0	0		1		1
0	0	0	1	0	1	0	0	0	0	0	0		1		1
0	0	0	1	1	0	0	0	0	0	0	0	...	1		1
0	0	1	0	0	0	0	0	0	0	0	0		1		1
0	0	1	0	0	1	0	0	0	0	0	0		1		1
0	0	1	0	1	0	0	0	0	0	0	0		1		1
0	0	1	0	1	1	0	0	0	0	0	0		1		1
0	0	1	1	0	0	0	0	0	0	0	0		1		1
0	0	1	1	0	1	0	0	0	0	0	0		1		1
0	0	1	1	1	0	0	0	0	0	0	0		1		1
0	0	1	1	1	1	0	0	0	0	0	0		1		1
0	0	1	1	1	1	0	0	0	0	0	0		1		1

## 组合各种模块，生成多种功能

- 实现函数
- 生成时钟
- 高速存取
- .....

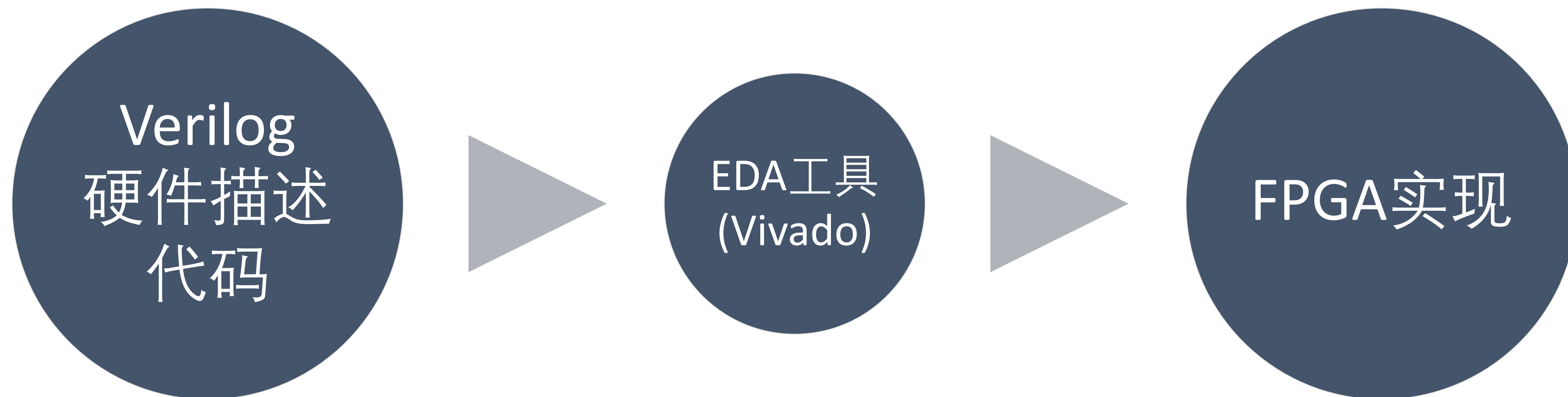




# 实验平台简介

- 使用FPGA实现SoC

## 使用FPGA实现SoC



## Verilog硬件描述代码

```
module unit1(  
    input clk,  
    output clkdiv  
);  
.....  
end module
```

- 通过代码描述硬件，而非FPGA芯片顺序执行代码
- 直接生成硬件电路，通过生成的电路并行处理输入信号



## EDA工具Vivado

- **Synthesis（综合）**
  - 拆分设计到可以实现的基础单元
  - 合理组合各种基础单元完成复杂设计的功能
- **Implementation（实现）**
  - 将综合出的各种单元布置在FPGA内
  - 连接各种单元
  - 调整布置位置和连接方法以达到要求性能
- **Generate Bitstream**
  - 生成下载到FPGA的文件

# 使用FPGA开发板实现计算机系统硬件部分

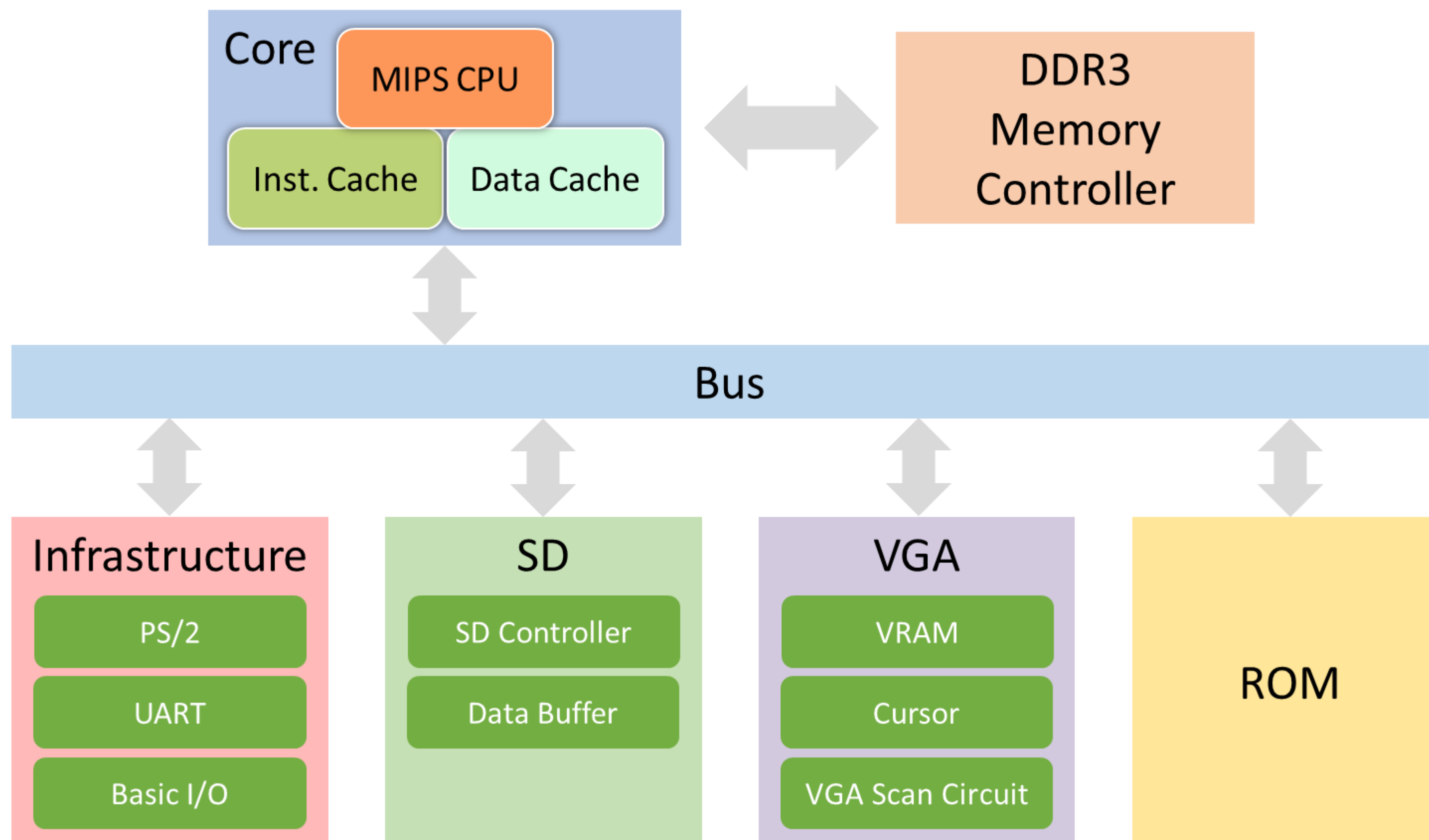
板载资源	实现硬件
FPGA逻辑门	CPU、Cache、wishbone总线等
DDR3 SDRAM	系统主存
SD设备读取接口	外部存储
PS/2键盘接口	输入设备
VGA显示接口	输出设备

# 系统架构简介





## ZJUNIX系统整体架构图

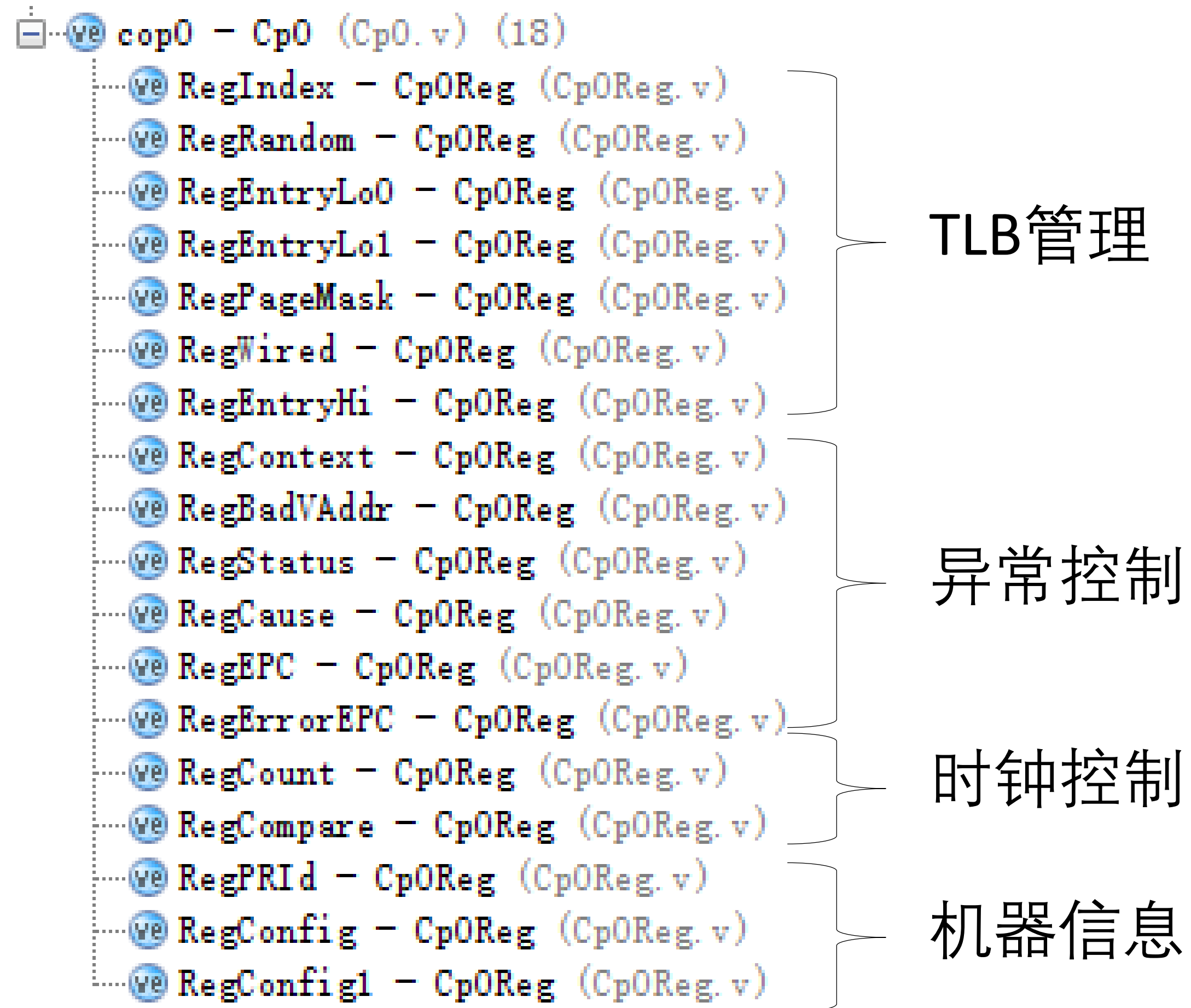


## CPU Core

```
[-] ve cpu - PCPU (PCPU.v) (11)
  [-] ve TP - TranslatePredict (TranslatePredict.v)
  [-] ve BP - BranchPredictor (BranchPredictor.v)
  [+] ve stageID - StageID (StageID.v) (1)
  [-] ve fwdRs - GprFwdUnit (FwdUnit.v)
  [-] ve fwdRt - GprFwdUnit (FwdUnit.v)
  [+] ve stageEX - StageEX (stageEX.v) (2)
  [-] ve stageMem - StageMem (StageMem.v)
  [-] ve regFile - RegFile (Regs.v)
  [+] ve tlb - TLB (TLB.v) (2)
  [-] ve exceptionCtrl - ExcControl (ExcControl.v)
  [+] ve cop0 - Cp0 (Cp0.v) (18)
```

CPU支持MIPS32指令集的子集（总计84条指令），为4级流水线架构，带有动态分支预测、地址翻译预测单元，使用TLB进行虚拟内存管理。TLB按照MIPS32规范设计，为32条全相联、先进先出(FIFO)替换策略。此外，CPU还带有总计128KB的一级缓存（指令、数据各64KB），为二路组相联、64字节块大小，使用LRU替换策略、写回(write back)和写不分配(write not allocate)的写策略。

## CP0协处理器



CP0的主要功能是异常处理、TLB管理和时钟控制等。本硬件平台的CP0完全按照MIPS32规范设计，共实现了18个寄存器。其中与TLB管理相关的寄存器为：Index, Random, EntryLo0, EntryLo1, PageMask, Wired, EntryHi；与异常控制相关的寄存器为：Context, BadVAddr, Status, Cause, EPC, ErrorEPC；与时钟控制相关的寄存器为Count和Compare；另外还有PRId, Config, Config1这3个与机器信息相关的只读寄存器。

## 外设硬件驱动

```
[-] ve infrastructure - Infrastructure_Sword (Infrastruc
  [+] CO - ClockGen (ClockGen.xci)
  [ve] U1 - Keypad (Keypad.v)
  [+] segDevice - Seg7Device (Seg7Device.v) (3)
  [ve] ledDevice - ShiftReg (ShiftReg.v)
  [+] ps2 - PS2Wrapper (PS2Wrapper.v) (3)
  [+] uart - UARTWrapper (UARTWrapper.v) (11)
  [ve] regSeg - GPIOREg (Infrastructure_Sword.v)
  [ve] regLED - GPIOREg (Infrastructure_Sword.v)
  [ve] regVGA0 - GPIOREg (Infrastructure_Sword.v)
  [ve] regVGA1 - GPIOREg (Infrastructure_Sword.v)

[-] ve vga - VGADevice (VGADevice.v) (5)
  [+] RAM0 - GraphicVRAM (GraphicVRAM.xci)
  [ve] RAM1 - CharVRAM (VRAM.v)
  [ve] ROM0 - CharROM (VRAM.v)
  [ve] U0 - VGAScan (VGAScan.v)
  [ve] cursor - vgaCursorGen (VGADevice.v)
```

基本外设包括时钟发生器，5x5的阵列按键，16个拨动开关，8位七段数码管和16个LED。

输入方面，PS/2键盘控制器，UART串口控制器，这两个模块都带有独立缓冲区。

输出方面，使用640x480的VGA显示，具有文本和图形两种模式，颜色深度为RGB各4位。



## BootLoader启动模块

PCPU.v (CPU硬件实现)

```
always @ (posedge clk)
begin
    if(rst)
        IF_PC <= 32'hbfc00000;
    else if(~masterStall)
        begin
            if(excReq)
                IF_PC <= PC_exc;
            else if(~IF_stall)
                IF_PC <= PC_BP;
        end
    end
end
```

BiosMem.v (双端口ROM)

```
initial
    $readmemb("../coe/bootstrap.hex", data);
```

在硬件进行复位的时候，rst信号线会至1，这时CPU的PC指针会指向bfc0\_0000。这时从0xbfc00000地址开始执行ROM中的程序bootloader程序。bootloader将所有内核代码从SD卡里的一个文件中全部读出，并加载到内存中0x80000000开始的地址，此后进入内核初始化入口。内核入口地址定为0x80001000。

## 系统架构升级说明

- 升级目的:
  - 尽可能与MIPS32规范兼容
  - 充分利用现有MIPS工具链和软件资源
- COP0(协处理器0):
  - 17个COP0寄存器, 全部与MIPS32规范兼容
- 虚拟地址空间映射:
  - 与MIPS32规范兼容的TLB MMU设计
  - 与MIPS32规范兼容的虚拟地址划分
- 高速缓存(Cache):
  - 修改了强制写回方式, 提升性能

# 实验流程简介



## 实验步骤

- 按向导生成Vivado工程，完成系统“硬件部分”
- 编译OS内核，完成系统“软件部分”
- 制作可以用于OS启动的microSD卡
- 完成整个OS启动实验



## 实验注意事项

- 请注意自己操作系统是Windows还是Linux，按照实验指导中对应的操作进行实验
- microSD格式化很重要，不合格的格式化会使OS无法启动





THANK YOU



The background image shows a traditional Chinese gate with multiple tiers of dark, tiled roofs and ornate wooden pillars. The gate is silhouetted against a sky filled with soft, white clouds. In the lower portion of the image, the sun is setting, creating a warm, golden glow that illuminates the clouds and the horizon. A modern building is visible in the distance on the right side.

# THANK YOU

实验开始