

远程实验 0_环境搭建和 ALU 设计

简介

计算机组成原理实验的目的在于通过一系列的实验课程使得学生能够逐步掌握并完成一款简单的 CPU 及片上系统的设计。

本次实验，我们将通过一个简单的 ALU（算术运算逻辑单元）实验，使学生熟悉并掌握本学期实验所设计到的实验环境、开发工具和编程语言。

实验目的

完成实验环境搭建

熟悉远程在线实验平台

完成 ALU 开发和调试

实验环境

PC 一台（网络流畅）

远程桌面环境（vlab）

FPGA 远程实验平台（FPGAOL）

实验步骤

Step1: 实验环境搭建（VLAB）

本实验需要用到赛灵思公司的 Vivado 集成开发工具，同学们可以在自己的电脑上下载并安装使用。

在本地搭建环境时需要注意以下两点：第一，该工具非常大，根据版本不同，安装包有 10~20 个 GByte；第二，软件存在兼容性问题，目前支持 Windows 和 Linux 的部分版本，对 Mac 等系统并不支持。

基于以上原因，很多同学在搭建环境时都遇到了困难，为减少同学们在搭建环境时浪费过多的时间精力，我们通过 VLAB 提供一个远程的在线环境，所有同学都可以通过统一身份认证登录我们的网站进行使用，网址为 `vlab.ustc.edu.cn`（支持校外访问）。根据笔者亲身测试，整个环境的搭建只需要 5 分钟左右。下面介绍一下主要流程，更详细的介绍可以参考 VLAB 使用说明。

首先，使用浏览器登录 VLAB 主页，并点击进入“虚拟机管理”页面，然后通过学校的统一身份认证系统登录，并创建一个虚拟机。



创建虚拟机时，务必选择 `vlab01` 镜像，只有该镜像中预装了 `vivado` 开发环境。虚拟机的创建大概需要 1~3 分钟，用户可通过刷新页面查看虚拟机创建情况。

虚拟机名称*

USTC

镜像选择* 我该选择哪个镜像？

`vlab01-ubuntu-desktop-full-18.04.tar.gz`

root 密码* 仅用于 SSH 登录和系统内 `su` 命令

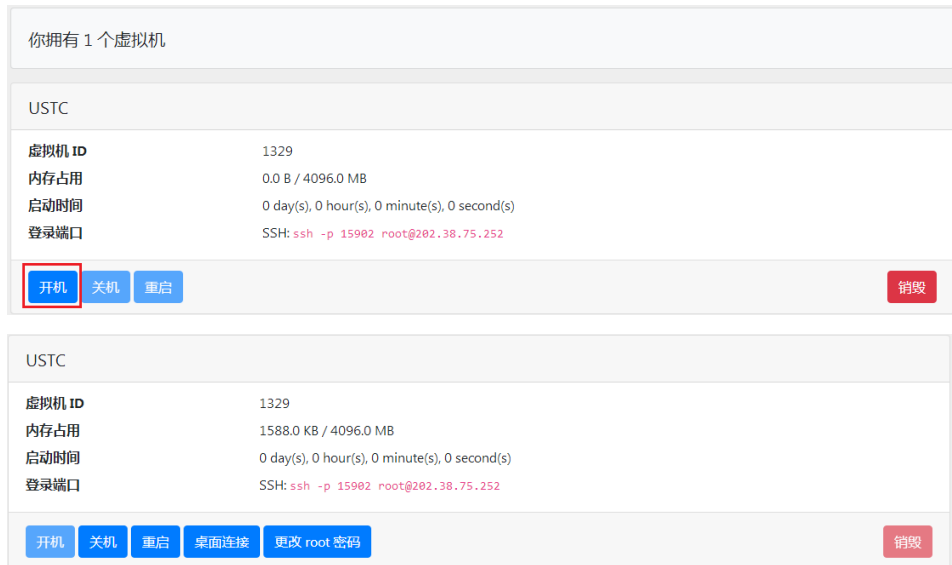
.....

请使用一个安全的密码，任何知道该密码的人都拥有虚拟机的最高权限

☒ 我已阅读并接受[服务条款](#)和[使用限制](#)*

创建

创建完成后，界面如下所示，这样你就拥有了一台已经预装了 Vivado 开发工具的远程虚拟机了，点击“开机”按钮打开虚拟机。



虚拟机开机后，可通过“桌面连接”按钮，由浏览器直接登录系统，也可以按照使用说明，配置 VNC 环境，由客户端登录（网页登录虽然方便，但不支持与本地的拷贝粘贴等操作）。

进入系统后，打开一个终端窗口，输入“vivado”，便等够打开 vivado 的 GUI 环境。至此，环境搭建工作完成。是不是很简单，小伙伴们赶紧来体验一下吧。

```
Terminal 终端 - ubuntu@VM1329-USTC: ~
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@VM1329-USTC:~$ vivado

***** Vivado v2019.1 (64-bit)
***** SW Build 2552052 on Fri May 24 14:47:09 MDT 2019
***** IP Build 2548770 on Fri May 24 18:01:18 MDT 2019
** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.
```

Step2: 远程 FPGA 实验平台（FPGAOL）

除了远程在线的虚拟机和开发环境，我们还提供了一套远程的 FPGA 实验平台（FPGAOL），同学们在 Vivado 中完成设计后，可以在该平台上进行验证。

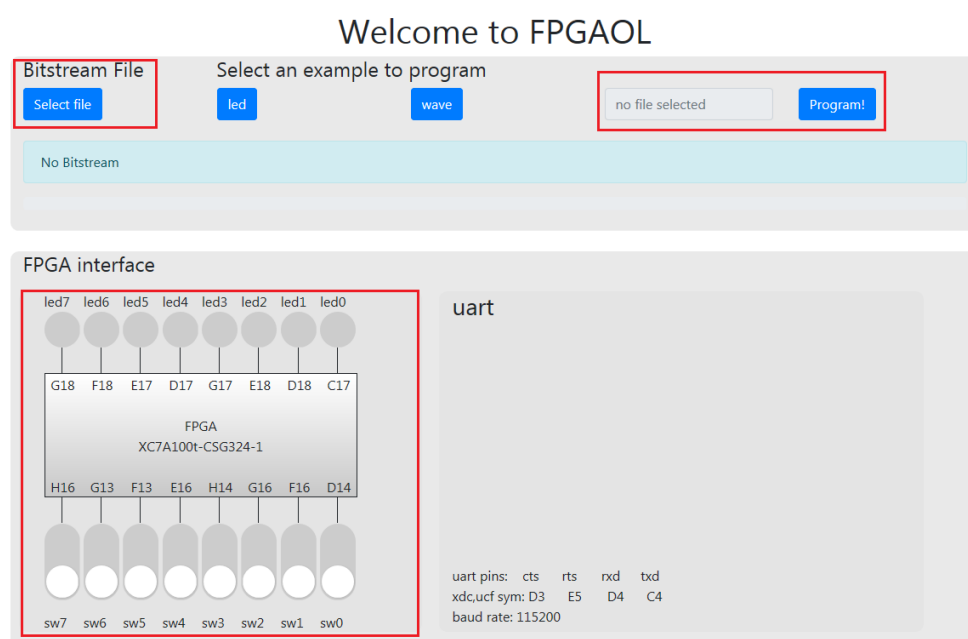
同学们可登录 `vlab.ustc.edu.cn`，点击“FPGA 在线”按钮，并通过学校的统一身份系统登录，然后点击“acquire”按钮，申请获取一个设备节点。

现在，我们后台共有 28 个设备节点供大家分时复用，所以请不要长时间占用，完成烧写验证后，请尽快退出，以供其他同学使用。



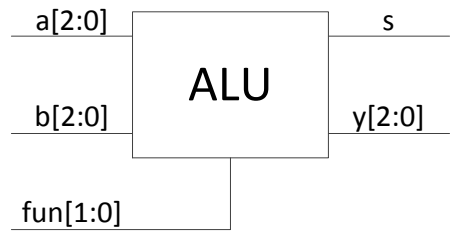
FPGAOL 的主界面如下图所示，用户可通过“Select file”按钮选中 vivado 中生成的 bit 文件，然后点击“Program”按钮进行烧写。通过模拟的开关和 LED 进行电路调试。

该平台的详细说明请参考使用说明文档。



Step3: ALU 设计

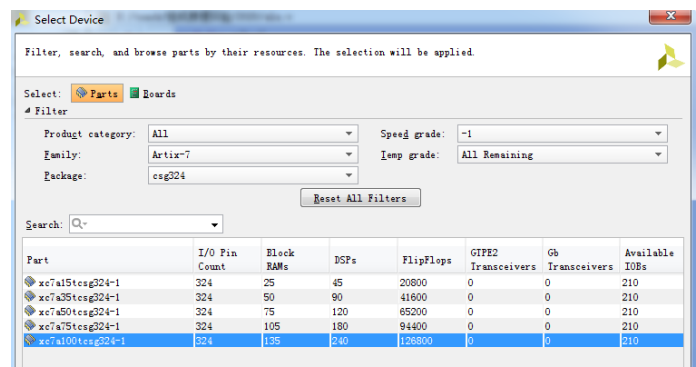
本实验中，我们完成一个简单的 ALU，并进行仿真和下载验证。
其框图如下所示：



其中，a, b, fun 为 3 个输入信号，y, s 为输出信号，其逻辑关系为：

输入:fun[1:0]	输出:y[2:0]	输出:s
2'b00	$y = a \& b$	1'b0
2'b01	$y = a \mid b$	1'b0
2'b10	$y = a + b$	进位位
2'b11	$\text{if}(b[2]) \ y = a \ll b[1:0]$ $\text{else} \quad y = a \gg b[1:0]$	移出位

在开始编码之前，需要同学们在 Vivado 中，建立一个工程，器件选择“XC7A100T-CSG324C-1”，如下图所示：



创建好工程之后，便可以按照以下要求，逐步完成实验内容了。

要求 1：根据前面的真值表，将示例代码补充完整，如下所示：

```
module alu(
```

```

input  [2:0] a,b,
input  [1:0] fun,
output reg s,
output reg [2:0] y);
always@(*)
begin
    case(fun)
        2'b00: begin s = 0; y = a & b; end
        2'b01: begin s = 0; y = a | b; end
        2'b10: begin {s,y} = a + b; end
        2'b11:
            begin
                //请将此部分补充完整
            end
        default: begin s = 0; y = 0; end
    endcase
end
endmodule

```

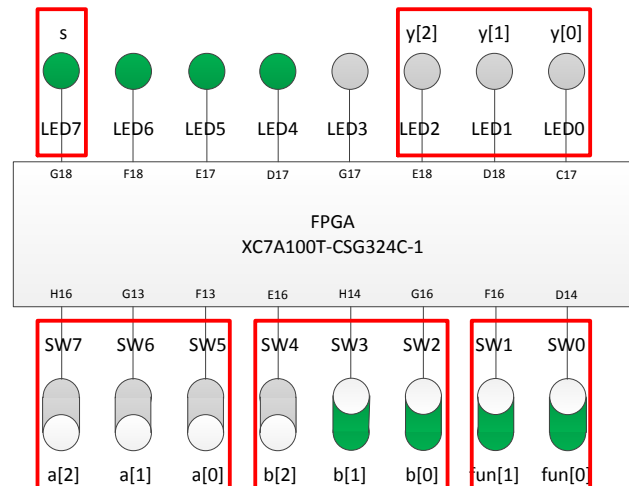
要求 2: 使用如下仿真代码，对设计文件进行仿真，根据仿真波形判断设计的正确性。

```

module tb();
reg [2:0] a,b;
reg [1:0] fun;
alu      alu(
    .a      (a),
    .b      (b),
    .fun     (fun),
    .s      (),
    .y      ()
);
initial
begin
    repeat(20)
    begin
        a = $random % 8;
        b = $random % 8;
        fun = $random % 4;
        #20;
    end
    $stop;
end
endmodule

```

要求 3: 根据要求，将 xdc 文件（管脚约束文件）补充完整，并对工程进行综合实现，最终生成可烧写到实验平台的 bit 文件，管脚对应关系如下图所示：

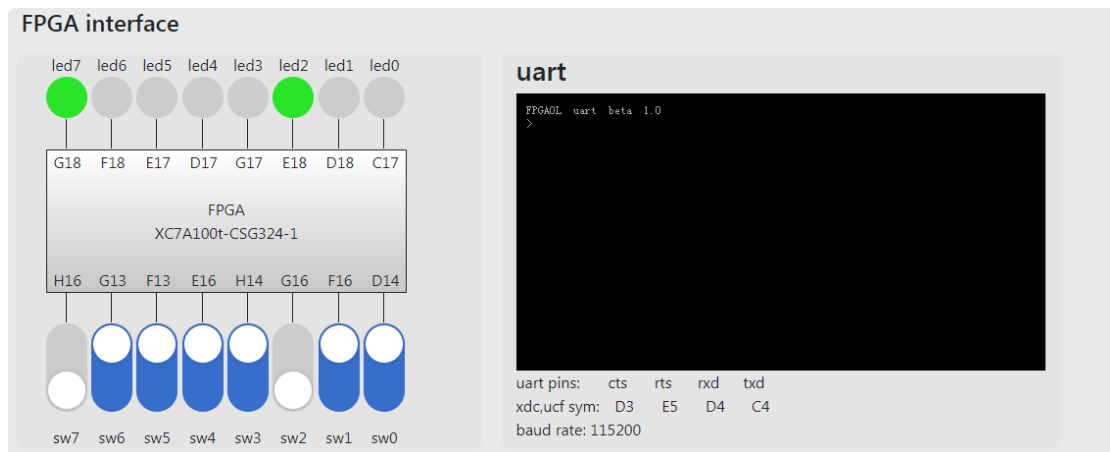


XDC 文件示例如下：

```
#INPUT a
set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { a[2] }];
set_property -dict { PACKAGE_PIN G13 IOSTANDARD LVCMOS33 } [get_ports { a[1] }];
set_property -dict { PACKAGE_PIN F13 IOSTANDARD LVCMOS33 } [get_ports { a[0] }];
#INPUT b
set_property -dict { PACKAGE_PIN E16 IOSTANDARD LVCMOS33 } [get_ports { b[2] }];
set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { b[1] }];
set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { b[0] }];
#INPUT fun

#Led
set_property -dict { PACKAGE_PIN G18 IOSTANDARD LVCMOS33 } [get_ports { s }];
set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { y[2] }];
set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { y[1] }];
set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { y[0] }];
```

要求 4: 将生成的 bit 文件烧写到 FPGAOL 平台，观察运行结果，并与仿真结果进行对比。



最后说明：我们在附加中提供了示例代码和一个测试用 bit 文件，供大家参考。

总结与思考

1. 请总结本次实验的收获
2. 请评价本次实验的难易程度
3. 请评价本次实验的任务量
4. 请为本次实验提供改进建议