

数字电路在线实验系统系统说明

系统部署

以下操作所在的环境都在 Ubuntu 14.04。

安装 modelsim

下载 ModelSimSetup-16.0.0.211-linux.run。这个版本不是最新的，但是经过检验是没有奇怪的配置问题的（当然更新的版本也不一定会有配置等问题）。可以在[官网下载](#)，也可以直接从我在校内的一台虚拟机中拷贝（也许会更加快一点吧）。

之后的安装过程请参考《ModelSim安装使用手册.pdf》。亲测安装手册中的 **P.S.** 是一定要做的。

安装完毕之后，我们主要使用 **vlib**、**vcom** 和 **vsim** 这3个命令。需要将相关的路径加入到环境变量中。同时我们需要在管理员权限下使用这些指令。在 `~/.bashrc` 中增加以下三行（假定你的 altera 安装在了默认的位置）：

```
PATH=$PATH:$HOME/altera/16.0/modelsim_ase/linuxaloem  
export PATH  
alias sudo="sudo env PATH=$PATH"
```

然后执行

```
source ~/.bashrc
```

安装 nodejs

一期工程中网站的后端使用 **nodejs**。安装过程参考官网即可：

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

mongodb的安装配置

安装

一期工程中使用 **mongodb** 作为数据库。根据[官网](#):

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F373035  
9A14518585931BC711F9BA15703C6  
echo "deb [ arch=amd64 ] http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list  
(这一句仅适用于14.04的版本)  
sudo apt-get update  
sudo apt-get install -y mongodb-org
```

开启数据库服务，并进行连接

安装完成之后我们需要对数据库进行一些配置来配合一期工程的使用。

首先，mongodb默认将数据存储在/data/db里面，你可以创建那个目录（如果没有的话）或者在启动服务的时候指定存储的路径（不指定的话默认就是 **/data/db**）

```
sudo mongod --dbpath /data/<path> --port <port no>
```

其中 port 是指定服务端口，默认为27017

开启服务之后就可以建立与这个服务的连接，同样默认的端口是27017

```
mongo --port <port no>
```

建立连接之后就可以进入一个类似于 Python 的交互界面，可以直接对数据库进行操作了

mongodb 在开机的时候的自启动，我一般会把这个给关掉。只需要修改 **/etc/init/mongod.conf**，将其中的

```
start on runlevel [2345]  
stop on runlevel [06]
```

改为

```
start on runlevel [345]
```

```
stop on runlevel [026]
```

即可。

进行一定的初始化操作

我们首先建立一个名字叫做 `ExprOnline` 的数据库

```
use ExprOnline
```

`use` 命令在数据库不存在的时候会创建并且切换到这个数据库下；数据库存在时仅完成切换功能

为了能够完成登录功能，我们可以通过以下的命令新建一个叫 `users` 的 table，并且初始化一个管理员的账号和一个普通用户的账号。具体的含义会在介绍数据库时进行详细的介绍

```
db.users.insert({ "authority" : 0, "group" : "class 5", "homeworkBox" : [], "msgBox" : [], "projectBox" : [], "userName" : "guest", "password" : "guest"})  
db.users.insert({ "authority" : 1, "group" : "", "homeworkBox" : [], "msgBox" : [], "projectBox" : [], "userName" : "admin", "password" : "admin"})
```

到此，我们创建了两个用户，一个普通用户，账号和密码都是 guest，另外一个是管理员用户，账号密码都是 admin。你可以用 `db.users.find()` 来进行查找。

一期工程将数据存在指定的文件夹中，需要实现先建立这些文件夹

```
sudo mkdir -p /home/kinnplh/projectFile  
sudo mkdir -p /home/kinnplh/cpuEntity  
sudo mkdir -p /home/kinnplh/homeworkFile  
sudo mkdir -p /home/kinnplh/homeworkFile/tem  
sudo mkdir -p /home/kinnplh/submitFile
```

克隆源代码并开启服务器

从 github 上直接克隆源代码

```
sudo apt-get install git  
git clone https://github.com/kinnplh/engineerdrawing.git
```

切换到根目录下之后安装相应的包

```
cd engineerdrawing/  
npm install
```

安装一些前端的依赖（主要是编辑器）部分

```
npm install bower -g  
cd hhEditor  
bower install
```

之后就可以开启服务器

```
node bin/www
```

在浏览器中输入 **localhost:3000** 就可以进入一期工程了。可以使用前面添加的用户进行登录。

总结

按照上面的简单的步骤就可以搭好一个一期系统的运行环境，并且将相应的系统运行起来。我这里也准备了 Ubuntu 的镜像、ModelSimSetup-16.0.0.211-linux.run 以及一个经过压缩的大小为 10G+ 的虚拟硬盘（如果你用 virtual box 装虚拟机的话就可以直接用）。有需要的话可以找我拷贝。

系统使用

这里介绍了系统使用的基本流程。根据上面的部署流程，可以通过 **localhost:3000** 对项目进行访问。

普通用户登录

使用上文中插入数据库的用户信息登录，账号是 **guest**，密码是 **guest**。当前没有提供注册的功能。

登录之后进入主界面。主界面主要分为两个部分：homework、playground。homework 是老师布置的作业，将在之后介绍管理员账户的时候介绍。homework 是用户自己创建的项目。可以点击右下角的+号进行创建。

The screenshot shows the main interface of the application. On the left is a vertical toolbar with various icons. The main area has two sections: 'Homework' (teacher-created projects) and 'Playground' (self-created projects). Both sections currently show 'No Homework' and 'No Project' respectively. A red box highlights the 'Homework' section, and a blue box highlights the 'Playground' section. In the bottom right corner, there is a green hexagonal button with a white plus sign inside.

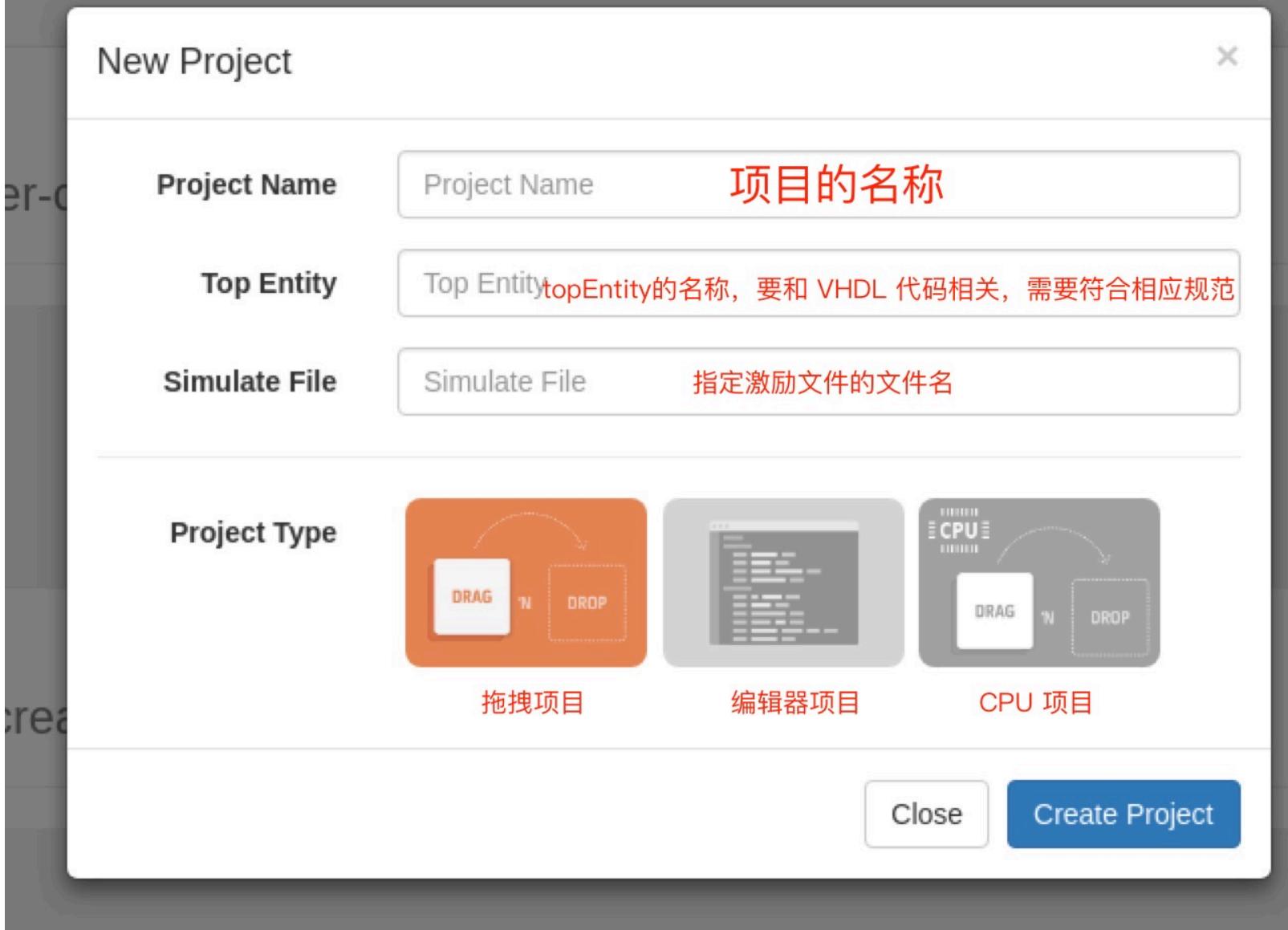
不知道为什么里面的那个加号加载不出了
我们假设这里有加号。点击创建新的 playground 项目

项目创建

点击主界面右下角可以进行项目的创建。创建项目需要填写以下4项：

1. 项目的名称（Project Name）：项目的名称不能和已有的项目的名称重合。
2. 顶层设计体名（Top Entity）：需要与 VHDL 代码相对应，因此需要满足一定的规范。
3. 仿真文件名（Simulate File）：设计使用哪个文件作为模拟仿真的输入，请以 VHD 为扩展名。
4. 项目类型（Project Type）：指定项目的类型，有拖拽项目，编辑器项目和 CPU 项目三种。

点击创建（Create Project）之后就可以在 Playground 中看到自己生成的项目。



拖拽项目

(在虚拟机环境中测试发现, 拖拽项目不能再 firefox 中使用, 原因大概是 firefox 不支持 JavaScript 的 `let` 语句。)

编辑项目仍然需要在创建项目的时候指定项目的名称、顶层设计体的名称和仿真文件名称。这三者在创建之后均不可修改。其中顶层设计体的名称和激励文件的名称只在后台创建 VHDL 代码的时候使用, 实际上对用户透明。考虑到用户体验, 后续可能需要随机生成一个顶层设计体名和激励文件名。

下图对拖拽界面进行了一定的展示。

从左到右分别是：存储电路图，进行编译仿真和查看提交列表

所有的元件
添加元件的时候，单击相应的元件，然后在画布的适当位置再点击一下即可

类型相符（输入和输出相连）的圆圈可以连接
按住 shift 键可以进行拖拽

操作 操作

导线颜色 1 2 3 4 5 6

其他元件 Vcc GND 输入 输出

芯片

- 74LS00 还剩 3 个
- 74LS04 还剩 2 个
- 六反相器
- 74LS11 还剩 3 个
- 三输入与门
- 74LS86 还剩 3 个
- 74LS90 还剩 3 个

made with □ by engineerDrawing

编辑器项目

（在虚拟机环境中测试发现，编辑器项目在 firefox 中是没有问题的，在 chrome 中由于安全性的问题，有一些库无法正常加载，导致不能正常使用。这可能是虚拟机配置的问题。）

新建一个编辑器项目之后会自动进入该项目的编辑界面。系统会自动生成一个 VHDL 文件 [Entity>.vhd](#)。

下图展示了界面上一些按钮的含义。

从左到右分别是：存储当前文件、上传本地文件、编译仿真、查看提交结果、设置

Online Experiment Example_adder

文件列表

adder.vhd
adder_tb.vhd

adder_tb.vhd x adder.vhd x +

```

100
101 -- case 2
102
103 A <= "01";
104
105 B <= "11";
106

```

标签栏，可以直接改名

linux 下 control+shift+d 可以删除文件
其他系统可以自己探索一下

注意，对于编译仿真，输入的激励就是在生成工程的时候指定的那个激励文件，你必须新建一个文件名称一模一样的文件；进行仿真的顶层设计体就是在生成工程的时候指定的那个顶层设计体（可以在“设置”选项中进行更改），这个文件在项目生成的时候已经生成好了。

在完成对源代码和激励文件进行编辑之后，就可以进行编译仿真了。

CPU 项目

CPU 项目和普通的拖拽项目相比，只是增加了一个从文件生成 VHDL 代码的功能。由于当初实现这个功能是为了能够实现一个 CPU，故名。

点击左上角最左边的圆形按钮，就可以进入一个编辑器。编辑器中的所有VHDL源代码会被解析，生成对应的可拖拽元件。

这部分还是比较不成熟的。例如没有考虑生成元件并且已经拖拽到界面上的时候，你又把对应的源文件给删除的情况。这在一期工程中属于一个扩展的功能。

编译仿真

每一次提交都会对应着一次仿真。可以在提交列表中查看。

Online Experiment Example_notop / submitList guest ▾

提交时间 编译状态
Sat Sep 23 2017 15:30:22 GMT+0800 (CST) Success

查看仿真结果 (VCD 文件)
View details Download Simulate Result Download All Files

Sat Sep 23 2017 15:29:51 GMT+0800 (CST) Error

Download All Files
下载所有文件 (包括源代码)

对于成功编译仿真的提交，可以查看显示仿真结果。

Online Experiment Example_notop / submitList / submitDetails guest ▾

in1 out1 仿真波形 150 in1 1 150 out1 0

波形缩放 某个时间点的仿真结果
随着鼠标移动而改变

Start time: 15:30:22 on Sep 23, 2017
vcom notop.vhd notop_tb.vhd
Model Technology ModelSim ALTERA vcom 10.4d Compiler 2015.12 Dec 30 2015
-- Loading package STANDARD
-- Loading package TEXTIO
-- Loading package std_logic_1164
-- Loading package std_logic_arith
-- Loading package STD_LOGIC_UNSIGNED
-- Compiling entity Chip74LS04
-- Compiling architecture arch of Chip74LS04
-- Compiling entity notop
-- Compiling architecture bhv of notop
-- Compiling entity notop_tb
-- Compiling architecture TB of notop_tb
End time: 15:30:22 on Sep 23, 2017, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
Reading pref.tcl

编译仿真时候的输出，一般没啥用，出错的时候可以看到错误信息

管理员用户登录

管理员后台简单汇总了所有用户的项目信息。除此之外的一个重要的功能是发布作业。

从左侧的导航栏中 Forms->Create Homework 可以进入布置作业的界面。

布置作业分为3个部分。

作业的逻辑是这样的：管理员在布置的时候，需要提供标准代码和激励文件。系统使用激励文件对标准代码

进行仿真，记录仿真结果。普通用户在完成作业的时候，只需要编写自己的代码。上传到系统之后，系统用管理员提供的激励文件对该代码进行仿真，将仿真结果和标准代码的仿真结果进行比较，来判断代码是否正确。

作业的基本信息

进行一些基本的设置。

其中的“Top Entity Name”和“Simulate File Name”要求和建立项目的时候的要求是一致的。



Step 1 Basic Info Step 2 Upload Files Step 3 View Simulation Result

Homework Name * Homework_adder

Project Type Drag and Drop Code Editing

Top Entity Name * adder

Simulate File Name * adder_tb.vhd

Input Port Name * A x B x in port
和代码对应的输入和输出端口名

Output Port Name * carry x sum x out port

Deadline * 09/30/2017 

Describe * Just for fun

上传相应的文件

请和上一步填写的“Top Entity Name”和“Simulate File Name”相对应。



1.2 KB
adder.vhd

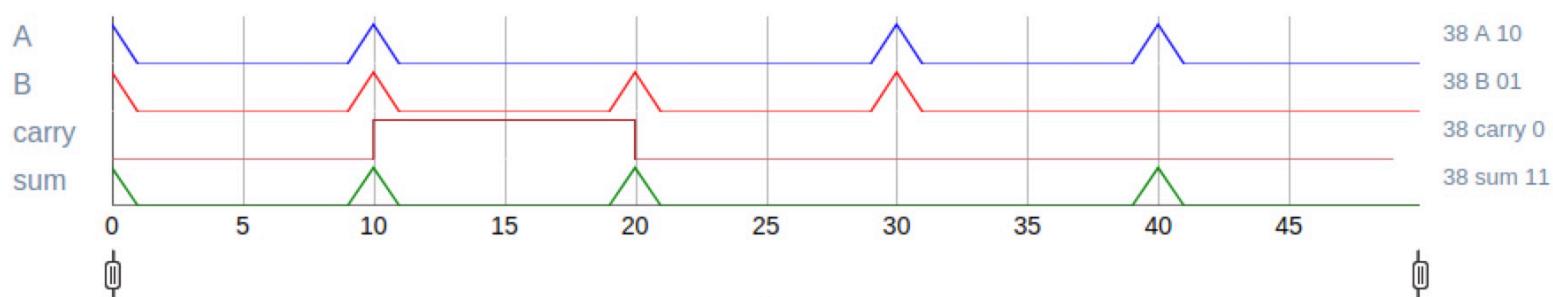
1.2 KB
adder_tb.vhd

Remove file Remove file

Finish Next Previous Loading X

查看仿真结果

如图所示。



如果和预期一致的话，点击 Finish 就可以将该作业发布给现有的所有用户。用户就可以愉快地完成这个作业了，提交之后会给评分。