[ICS-PJ REPORT 2](#_Toc26998620)

[1.前期准备： 2](#_Toc26998621)

[1.1开发环境： 2](#_Toc26998622)

[1.2文件结构： 2](#_Toc26998623)

[2.功能概述： 3](#_Toc26998624)

[3.使用方法： 3](#_Toc26998625)

[3.1打开命令行 3](#_Toc26998626)

[3.2打开网页 4](#_Toc26998627)

[3.3页面各板块介绍&板块使用 4](#_Toc26998628)

[3.3.1文件上传板块： 4](#_Toc26998629)

[3.3.2代码显示板块: 5](#_Toc26998630)

[3.3.3流水线控制板块： 6](#_Toc26998631)

[3.3.4流水线板块： 7](#_Toc26998632)

[3.3.5速度控制板块： 7](#_Toc26998633)

[3.3.6状态板块： 8](#_Toc26998634)

[3.3.7寄存器/内存板块： 8](#_Toc26998635)

[3.3.8评论区板块： 9](#_Toc26998636)

[4.流水线实现 11](#_Toc26998637)

[5.前后端交互 11](#_Toc26998638)

[6.开发流程 12](#_Toc26998639)

[第一周11.14~11.17： 12](#_Toc26998640)

[第二周11.18~11.24 12](#_Toc26998641)

[第三周11.25~11.27 12](#_Toc26998642)

[第三周11.28~12.1 13](#_Toc26998643)

[第四周12.2~12.8 14](#_Toc26998644)

[第五周12.9~12.11 14](#_Toc26998645)

[7.新技能get： 14](#_Toc26998646)

[8.感想 14](#_Toc26998647)

[9.致谢 15](#_Toc26998648)

# ICS-PJ REPORT

## 1.前期准备：

需要安装[Python 3.8.4](https://www.python.org/downloads/windows/)、 [Django 2.1](https://www.runoob.com/django/django-install.html)，在Windows上运行.

### 1.1开发环境：

* 开发语言：Javascript/HTML/CSS/Python
* 浏览器环境：Chrome
* 第三方库：jQuery

### 1.2文件结构：

***后端代码：***

**simulator/backend**

Stage/init.py 空文件-建包

Stage/Fetch.py 流水线取指阶段

Stage/Execute.py 流水线执行阶段

Stage/Decode.py 流水线译码阶段

Stage/MemoryAccess.py 流水线访存阶段

Stage/WriteBack.py 流水线写回阶段

Stage/PCUpdata.py 流水线更新阶段

y86-code 测试样例

init.py 空文件-建包

run.py 运行测试样例

cpu.py 流水线逻辑控制

const.py 常量说明

Memory.py 内存类的定义，以及加载、读取函数

Pipeline.py 流水线类的定义

register.py 寄存器类的定义，以及加载、读取函数

Alu.py 数字逻辑运算单元

StringOperation.py 字符串操作函数

***前端代码：***

**simulator/static**

font/iconfont.css 定义了icon图标的样式

backimg.jpg 网页背景图

book.jpg 网页默认图标

style.css 定义了main.html中的各个div样式

logic.js 实现安排了流水线详细数据在页面的布置+异步刷新数据

index.js 主要是使用JQuery库对评论区功能的实现

**simulator/templates**

main.html 网页主体

***django框架：***

manage.py 命令行工具

settings.py设置template、static

urls.py 配置路径

views.py 对应urls.py的路径

wsgi.py 服务器路口

***文件：***

Comment.json 储存了评论区输入的数据

Show.json 储存了流水线详细数据

## 2.功能概述：

* **实现了Y86所有指令集**
* **实现流水线控制逻辑，完整展示流水线的信息：包括条件码、CPI、寄存器、内存、各个阶段信息**

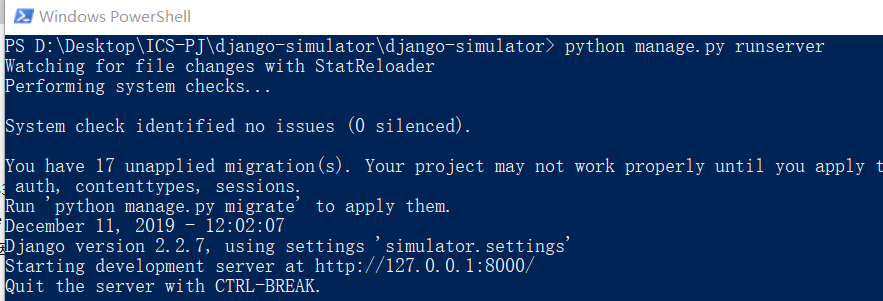
在此基础上，还添加了新的功能：

* **用于交流的评论区，可提交评论、点赞、点踩、删除、记录时间，功能强大。**
* **web页面简洁大方，每个版块精心设计，真的很养眼，人机交互及其快乐。**
* **支持运行、步进、步退、暂停、重置操作并且能以不同频率运行**
* **在代码显示框能动态查看正在运行的指令**
* **能够选择查看寄存器还是内存的数据**
* **增加了iaddq指令**

## 3.使用方法：

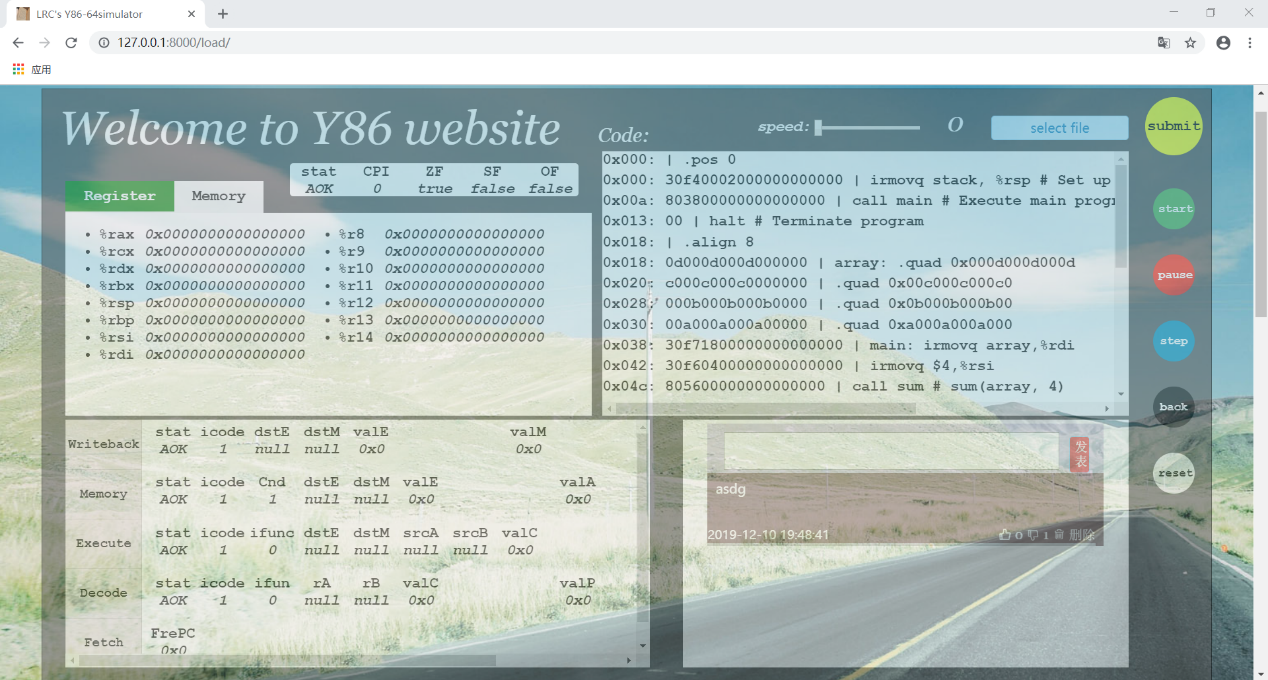
### 3.1打开命令行

进入django-simulator/simulator目录，按住键盘上面的shift键，然后在当前文件夹空白的地方鼠标右击，在弹出的选项卡中可以看到一个在此处打开命令行窗口的选项，输入python manage.py，



### 3.2打开网页

在Chrome浏览器中输入<http://127.0.0.1:8000/>，得到如下页面：



### 3.3页面各板块介绍&板块使用

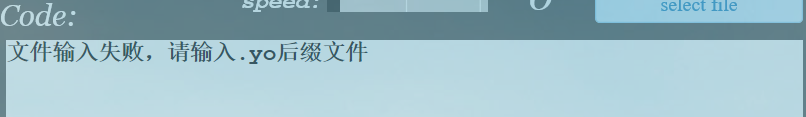
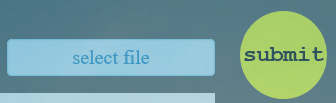
### 3.3.1文件上传板块：

**特点**：使用form表单进行前后端数据的交互。通过HTML中的input、button标签实现前端数据的传入，并美化标签，还设置了鼠标的悬停效果；

**使用&实现**：点击蓝色方框，会弹出文件选择页面，选择文件后，点击submit提交，这会将文件名传递给后端处理。

**如果文件名有效（以.yo后缀）**，后端处理得到数据类型为dict，键为流水线当前cycle数，对应的值为类型为字典，保存了流水线的详细数据，在通过json.dump（）处理为json数据写入show.txt文件中。同时后端将筛选输入文件中以0x开头代码行传递给HTML，通过模板语言显示在代码展示板块中；

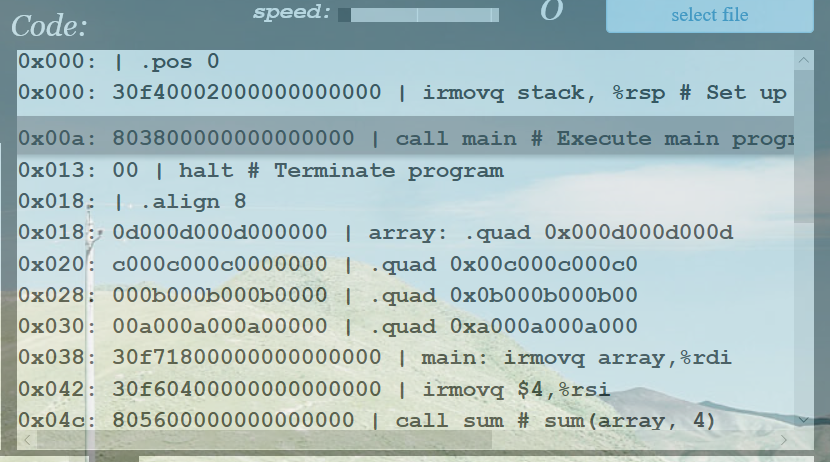
**如果文件名无效**，后端将给HTML返回一个0000000000000000，通过模板语言if判断文件无效，将在代码展示模块中显示“文件输入失败，请输入.yo后缀文件”



### 3.3.2代码显示板块:

**特点**：设置了代码运行时的突出显示效果

**实现**：通过得到当前流水线的f\_pc值可以得到当前运行指令对应的内存地址，获取每一行代码，对比地址值，相等则特别设置该行阴影、颜色属性，得到动态的突出的效果显示。



### 3.3.3流水线控制板块：

**特点**：通过Ajax异步刷新网页数据；设置了按钮悬停效果，实现开始、停止、下一步、返回、重置功能；

**实现**：主要实现了一个step函数，通过jQuery的ajax方法获取show.json中的数据，用cnt下标索引得到数据的键值对。每次step( )都会得到下一个键值对。

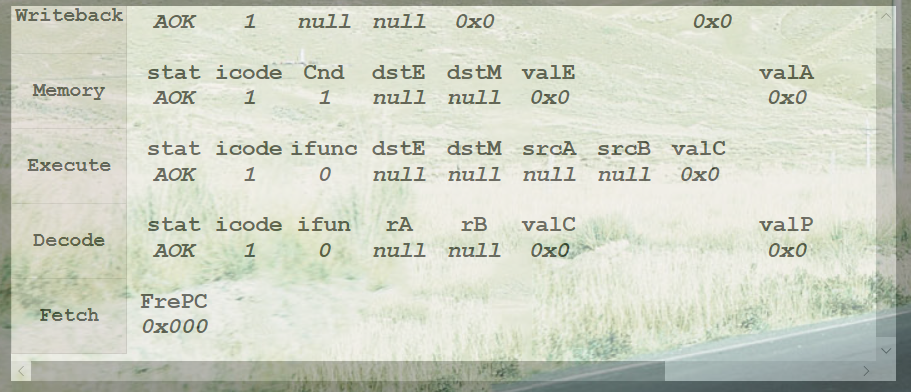
* Start函数：setInterval(step,speed)让step函数以speed( ms)/条指令速度运行。
* Pause函数：通过clearInterval杀死计时器。
* Step函数：step函数。
* Back函数：将索引cnt的值-2，运行step一次。
* Reset：将索引cnt的值设置为0，并运行step一次。

### 3.3.4流水线板块：

**特点**：通过滚动条查看，字体好看。

**实现**：通过控制板块按钮控制运行，每次调用step函数，ajax把对应的数据显示到页面上；val数据之间距离大是因为储存数据长。



### 3.3.5速度控制板块：

**特点**：动态显示滑块的值，滑条值在0~30，计算公式（500-x\*15），可以实现50~500ms/step的运行速度

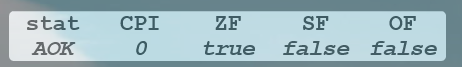
**实现**：拉动滑条可以改变setinterval（step,speed）计时器的speed，调节流水线运行速度。另外每次拉动滑条改变数据时，要求重新点击start才能改变运行速度，这样可以防止误调节滑条。

C:\Users\Dell\AppData\Local\Temp\WeChat Files\f16bbfcaf60af3b6117cd56213414c5.png

### 3.3.6状态板块：

**特点**：CPI的值设置为流水线运行结束时显示

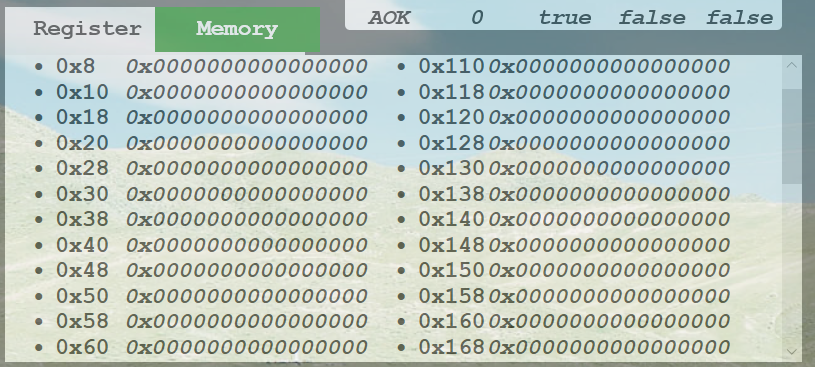
**实现**：每次调用step函数用ajax方法将数据显示到页面上



### 3.3.7寄存器/内存板块：

**特点**：通过点击板块上方的Register、Memory标签可以选择查看的内容，并且设置文件没有正确上传时不能进行选择。

**实现**：通过click事件动态设置对应标签的class值，通过css特别设置class值对应的样式就可以控制对应标签内容显示与隐藏。



### 3.3.8评论区板块：

**可吹可黑，可赞可踩。**

**灵感来源**：希望web提供一个可以交流互动分享的平台（或者说更像一个web？）

**实现**：textarea标签实现文本输入，button标签实现提交。点击事件主要通过jQuery中的.delegate( )方法来处理。数据交互ajax实现。（代码在index.js中）。

**功能特点**：处心积虑+精心设计；逻辑可以分为两部分，一部分ajax实现，控制前后端数据的交互（即js与python交互）；一部分jQuery选择器直接控制HTML的显示（即js与HTML交互）。并且因为前后端数据交互，刷新页面后评论会被重新渲染加载。

* 不输入文本内容不给submit，submit点击后，js会生成一个<div></div>插入到HTML中显示，并且每次submit都通过ajax方法将数据写入comment.json文件中；
* 赞、踩：（点击文字部分，不要点图标）可以实现数据变化，并通过ajax将后端对应数据修改。
* 删除，你删除的只是删掉你看到的，删不掉你看不到的：删除功能可以删除页面上对应评论，但无法删除后台对应数据，所以只能做到眼不见心为净。刷新页面之后删除的评论将重新显示在评论区。（您宝贵的意见、评价将被永远保存）



## 4.流水线实现

**Reference：流水线的实现非常感谢聂希瑞学长，没有他在逻辑上的指导，我不可能这么顺利地理清流水线的逻辑，以及对一些对数据的处理方法也是向他学习。**

在完成流水线的实现过程中我主要梳理了以下问题：

1. 各个阶段的逻辑。（这个通过看书可以了解）。
2. 流水线各个阶段如何更新。更新让我联想到swap函数，通过类似方式就容易解决了，就逐渐想到把前后状态封装成一个类，但对于五个阶段倒序执行这一逻辑我确实没能很快想到。
3. 原始数据（.yo文件）的处理。这一块本来不难却琢磨了很久。

## 5.前后端交互

Django框架下开发web，我没有使用django的model，而是使用了.json文件来储存后端的数据。最后进行了多种方式，不同语言之间数据的交互：

**Html与python：**通过在javascript文件的ajax方法获取数据并在js文件中处理，实现了异步刷新；通过Html的form表单标签发送请求，并通过使用django模板语言获取返回的数据。

**Javascript与Html**：通过jQuery选择器、document对象获取html的元素，以此进行web页面上数据的修改……

## 6.开发流程

### 第一周11.14~11.17：

开始阶段算是一头雾水，包括语言的选择、前后端如何交互、选择linux还是window，teminal是什么……。经过漫长的百度了解，包括python（CGI）、c自带GUI、Web、Unity都了解了一些。过程中被推荐了python，因为它里面封装了很多制作图形界面相关的库函数（虽然我也还不知道有哪些）。

在王辰浩学长的帮助下，我理解了前后端，真正明白这个PJ要做的事情并在学长的指导下，最后大概决定使用python后端，django+web前端。

### 第二周11.18~11.24

学习了python，学会使用了给的sim样例在图形界面、终端的输出，重新仔细阅读了教材熟悉了流水线的各种细节并对如何用python实现有了想法。在11.24晚上写出了大致框架，并做了memory方面的调试。

### 第三周11.25~11.27

11.25调试过prog1~prog10。

11.26调试过其它函数。

11.27学习markdownpro书写实验报告，实现了CPI的计算，实现了IADDP指令

以下是大概的问题描述,还有许多小问题遗忘了（辣眼睛）：

* prog1~4：问题多到数不清，python全局变量、变量是否可变……的python基础知识。
* prog5：writeMemory参数设置为int；PCUpdata中的条件问题
* prog6：hex（）自带“0x”问题
* prog7：由于jne的ifunc没读到4，没有设置成功Cnd，原因是ifunc 与 iFunc的统一
* prog8：d\_srcA不用mrmvoq ，d\_srcB不用rrmovq
* prog9：处理了invalid\_instr的输出
* prog10：处理地址越界的情况
* cjrr：大改ALU逻辑
* ret-hazard：修改PCUpdata，+not D\_stall
* abs-asum-movq：PCUpdata逻辑的优先级
* asumr：Execute中now.E[icode]不用,IRRMOVQ
* 加入iaddq时：不要忘了set\_cc的设置
* 实现CPI：W\_bubble不用计算，因为此时指令已经结束
* Cjr：重写了CPI计算方法

### 第三周11.28~12.1

11.28第一阶段提交，加入通过命令行控制后端功能

11.29~11.30学习html、css

12.1学习jQuery、javascript、bootstrap

解决问题：前端http请求，使得python文件读取.yo文件，再将python产生的字典数据传递到js处理，用html文件确定页面布局，将按钮、输入框等交互元素与js文件中函数绑定，因此可以将要显示出来的结果显示在页面对应位置，网页样式通过.css文件修饰。

### 第四周12.2~12.8

12.2~12.6通过菜鸟教程、csdn阅读了各种html、js、css文件，期间在学习理解ajax、jQuery上分别花了大量时间。

12.7~12.8初步完成了前端web页面的布局，吐槽一下css写的真是又臭又长。

### 第五周12.9~12.11

12.9~12.10完成了评论区功能和代码板块动态指示当前运行指令功能

12.11完善了一下评论区功能，书写实验报告。

## 7.新技能get：

* Python
* HTML5/CSS3/Javascript
* Django
* jQuery Ajax/Documnet
* JSON
* Github

## 8.感想

体会过搜寻半天也没结果的迷茫、体会过代码看不懂的忧伤、体会过苦思半天却发现走了弯路的苦涩；但也记得一步步找到解决问题逻辑的振奋、记到debug后样例正确输出的手舞足蹈、记得一朝顿悟的痛快、记得美化修饰一个个web元素的满满成就、记得实现一个个功能后与室友诉说分享的喜悦。

老实说，自己是个佛系的人，加入拔尖班就是希望挑战自己，测试自己的潜力。从第一阶段参考学习到第二阶段通过搜索百度独立完成交互、修改的所有逻辑，虽然两个阶段逻辑侧重点不同，不可一概而论，但我是真心欢喜自己的变化。还记得蔡学长在学期开始前就教导我们：自主学习、充分利用网络这项技能很重要。我一直记在心里，我想，现在自己也该有了一定进步，知道对于一个问题搜什么？怎么有效率地搜？

另外，学习新知识的过程是那么充满挑战而充实快乐。除了c++知识作为基础外，这次PJ所用的各种语言Python、Html……还有一些库都是从头到尾自学，看博客、看视频……摸索它们中适合，能够应用到这道PJ的方法。学了这么多东西，做到了自己希望做到的事，修饰出了一个还算满意的web界面。

回顾历程，其实，坚持是关键，每天的投入让我每天都在成长。

总之，和同学们之间差距仍在，“革命尚未成功，我辈仍需努力”。

## 9.致谢

感谢王辰浩学长给我的PJ指明了方向。

感谢聂希瑞学长提供他的优秀PJ供我参考。

感谢在实验报告书写方面以张作柏学长、周芯怡学姐PJ的书写格式参考。