计算机体系结构

DLXMIPSRISC-V指令格式

实验报告

|  |  |
| --- | --- |
| 学 院： | 计算机与信息技术学院 |
| 专 业： | 计算机科学与技术 |
| 指导教师： | 吴为民 |
| 班 级： | 计科1602 |
| 姓 名： | 麻锦涛 |
| 学 号： | 16281262 |

2019年 4月 3日

1. 熟悉winDLX模拟器，并确定指令格式中各个域的具体值。包括如下内容：

① 阅读winDLX目录中的wdlxtut.pdf文件（为了方便英文阅读困难的同学们，我从网上下载了一个中文版文件WinDLX教程(中文).doc），并按其中的步骤操作、学习winDLX模拟器。在完成下面问题的过程中，若有疑问，可在MIPS目录的材料中查询相关信息，或者在百度上查询相关信息。

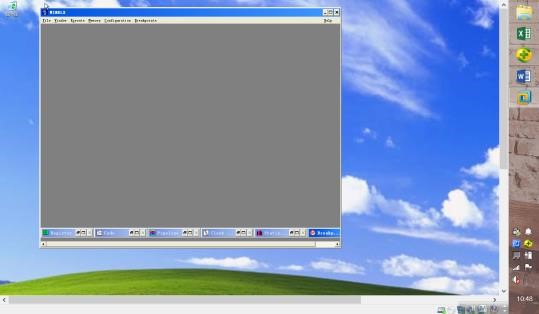
② 对于FACT.S程序，请从中选出3-5条不同的指令，并分别指出它是哪种指令（R-type，I-type，还是J-type），并参照教科书2-28页中的图2.19，填写指令格式中各个域的二进制值（提示：将程序载入到模拟器后,可在CODE子窗口中观察到）。为了清楚起见，最好用填表的形式。

③ 请通过CODE窗口的各种选项查看数据，进而判断出DLX是big-endian还是little-endian？是否是对齐的（aligned）？

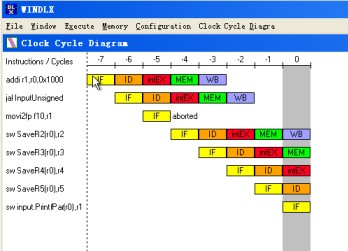
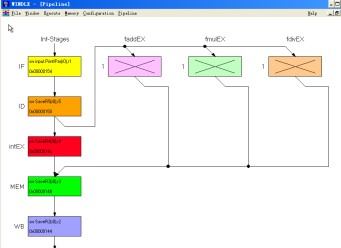
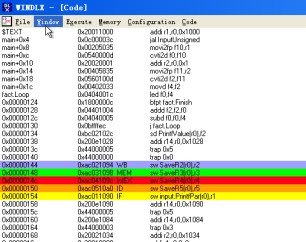
### 实验过程：

①因为在 win8 系统上 winDLX 无法运行，所首先安装了虚拟机，在

电脑中安装了 XP 系统，运行 winDLX，如下图：



# winDLX 的使用：



②选择 5 条不同指令分析指令格式：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令 十六进制 | | | | | | |
| addi r1, r0, 0x1000 | | | | 0x20011000 | | |
| 二进制 | | | | | | |
| 0010 0000 0000 0001 0001 0000 0000 0000 | | | | | | |
| 0--5 | 6--10 | 11--15 | 16--20 | 21--25 | 26--31 | Format |
| 00 10 00 | 00 00 0 | 00 00 1 | 0001 0000 0000 0000 | | | I-type |
| Opcode | Rs1(r0) | Rd(r1) | Immediate(1000) | | |  |

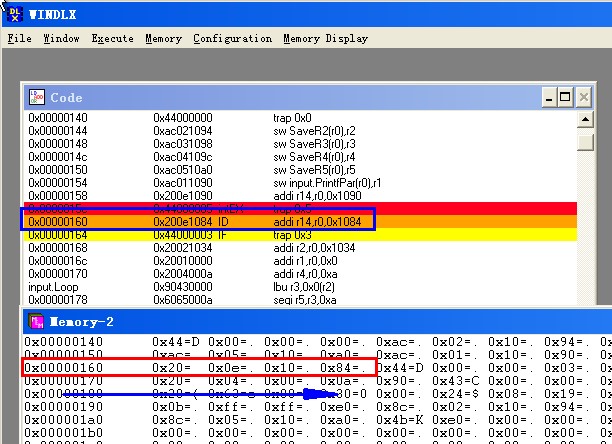
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令 十六进制 | | | | | | |
| led f0, f4 | | | | 0x0404001c | | |
| 二进制 | | | | | | |
| 0000 0100 0000 0100 0000 0000 0001 1100 | | | | | | |
| 0--5 | 6--10 | 11--15 | 16--20 | 21--25 | 26--31 | Format |
| 00 00 01 | 00 00 0 | 00 10 0 | 0000 0000 0001 1100 | | | I-type |
| Opcode | Rs1(f0) | Rd(f4) | Immediate(001c) | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 指令 十六进制 | | | | | |
|  | trap 0x3 | | | 0x44000003 | | |
|  | 二进制 | | | | | |
|  | 0100 0100 0000 0000 0000 0000 0000 0011 | | | | | |
| 0--5 | 6--10 | 11--15 | 16--20 | 21--25 | 26--31 | Format |
| 01 00 01 | 00 0000 0000 0000 0000 0000 0011 | | | | | J-type |
| opcode | Value | | | | |  |

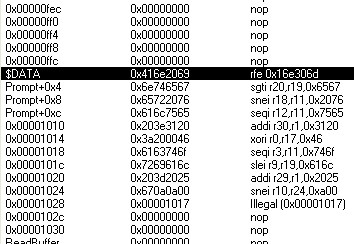
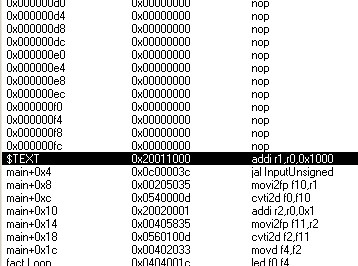
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令 | | | | 十六进制 | | |
| jr r31 | | | | 0x4be00000 | | |
| 二进制 | | | | | | |
| 0100 1011 1110 0000 0000 0000 0000 0000 | | | | | | |
| 0--5 | 6--10 | 11--15 | 16--20 | 21--25 | 26--31 | Format |
| 01 00 10 | 1111 1000 0000 0000 0000 0000 | | | | | J-type |
| opcode | value | | | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令 十六进制 | | | | | | |
| multd f2, f2, f0 | | | | 0x04401006 | | |
| 二进制 | | | | | | |
| 0000 0100 0100 0000 0001 0000 0000 0110 | | | | | | |
| 0--5 | 6--10 | 11--15 | 16--20 | 21--25 | 26--31 | Format |
| 00 00 01 | 00 01 0 | 00 00 0 | 00 01 0 | 00 00 0 | 00 01 10 | R-type |
| 0x0 | Rs1(f2) | Rs2(f0) | Rd(f2) | unused | opcode |  |

③ DLX 是 big-endian（大端字节序），是对齐的（aligned）。



* 蓝色箭头所指方向为低地址高地址，从 CODE 窗口可以看到从地址 0x00000160 开始应该存放指令 addi r14, r0, 0x1084 的二进制代码即 0x200e1084，而在对应的内存地址中，由低到高存放的顺序是 20、0e、10、84，也就是在 winDLX 中，高位存放在低地址，低位存放在高地址，因此 winDLX 是大端字节序（big-endian）。
* 对齐方式判断：



从 CODE 窗口中观察可以发现，因为地址是以字节为单位，所以实验中用到的32位的指令所存放的起始位置均放到了为4的倍数的地址上， 即使某条指令存放的地址的前面是未存放指令或数据的，该指令依然是会从地址为 4 的倍数的位置开始存放，因此 winDLX 中是对齐的。

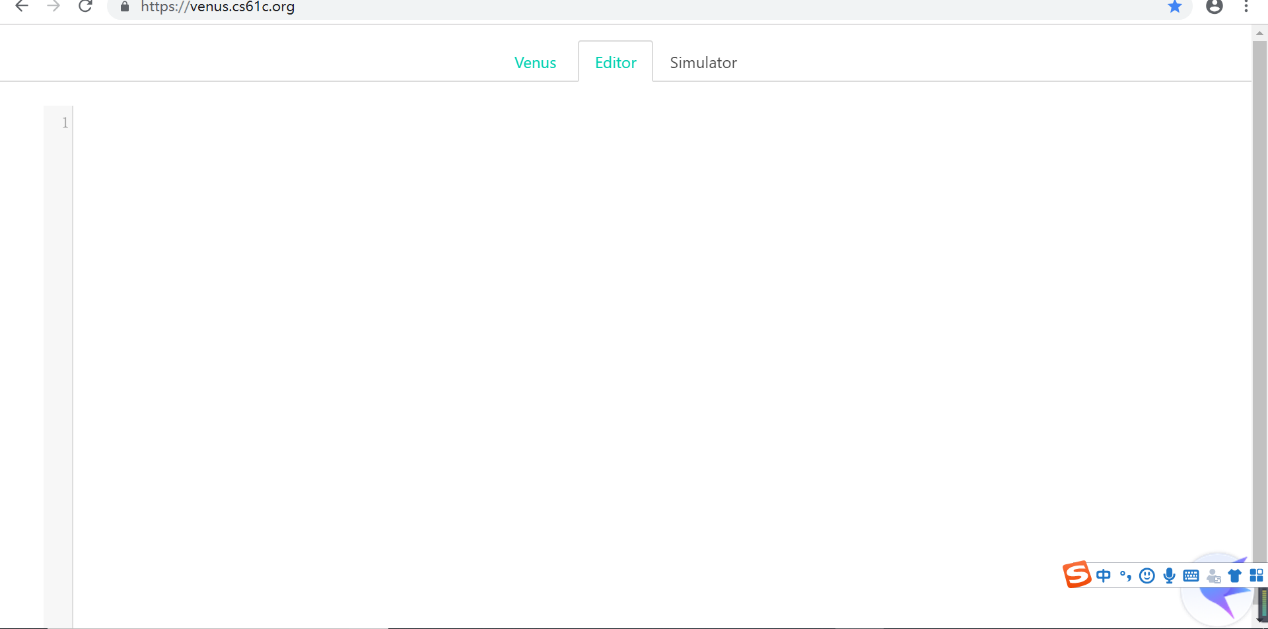
2. 了解RISC-V的在线模拟器Venus的使用方法。包括如下内容：

① 阅读Venus目录中的venustut.txt文件，并按其中的步骤操作、学习RISC-V的在线模拟器Venus的用法。在完成下面问题的过程中，若有疑问，可在RISC-V目录的材料中查询相关信息，或者在百度上查询相关信息。

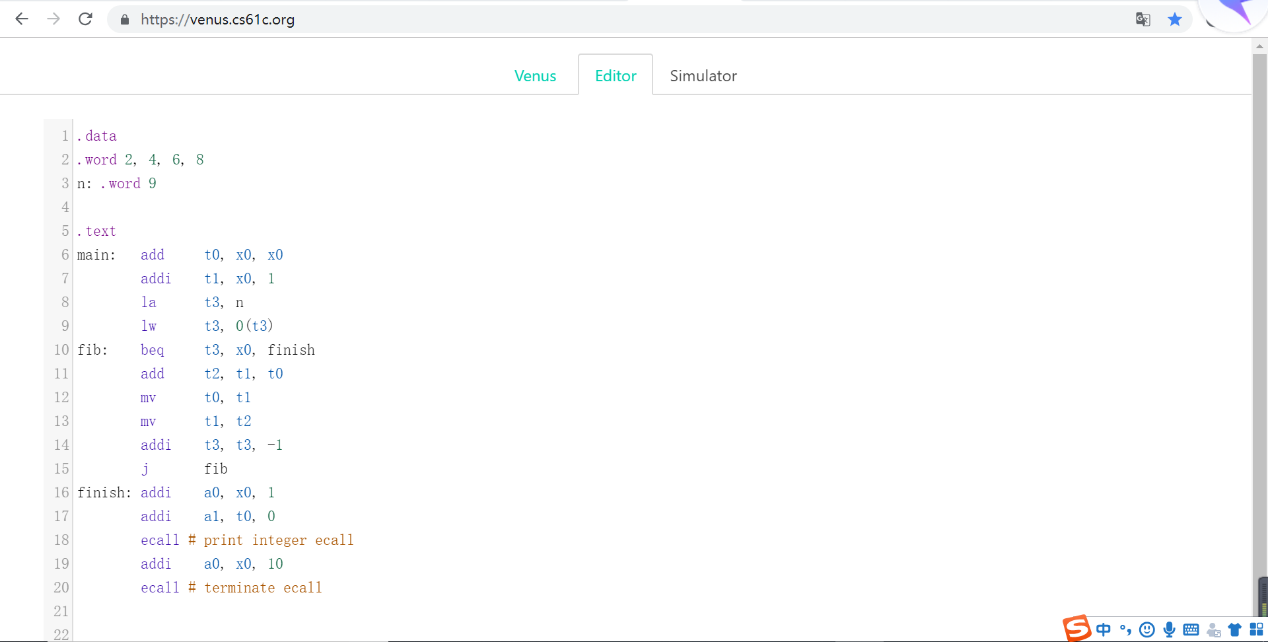
② 程序ex1.s的功能是什么？程序运行完后，得出的结果是什么数字？

③ 对于ex1.s程序，请从中选出3-5条不同的指令，并对于其中每条指令，指出它是哪种指令。请注意：RISC-V的指令格式分6种（除了R-type，I-type，J-type，还有U-type,S-type,B-type），并填写指令格式中各个域的二进制值。

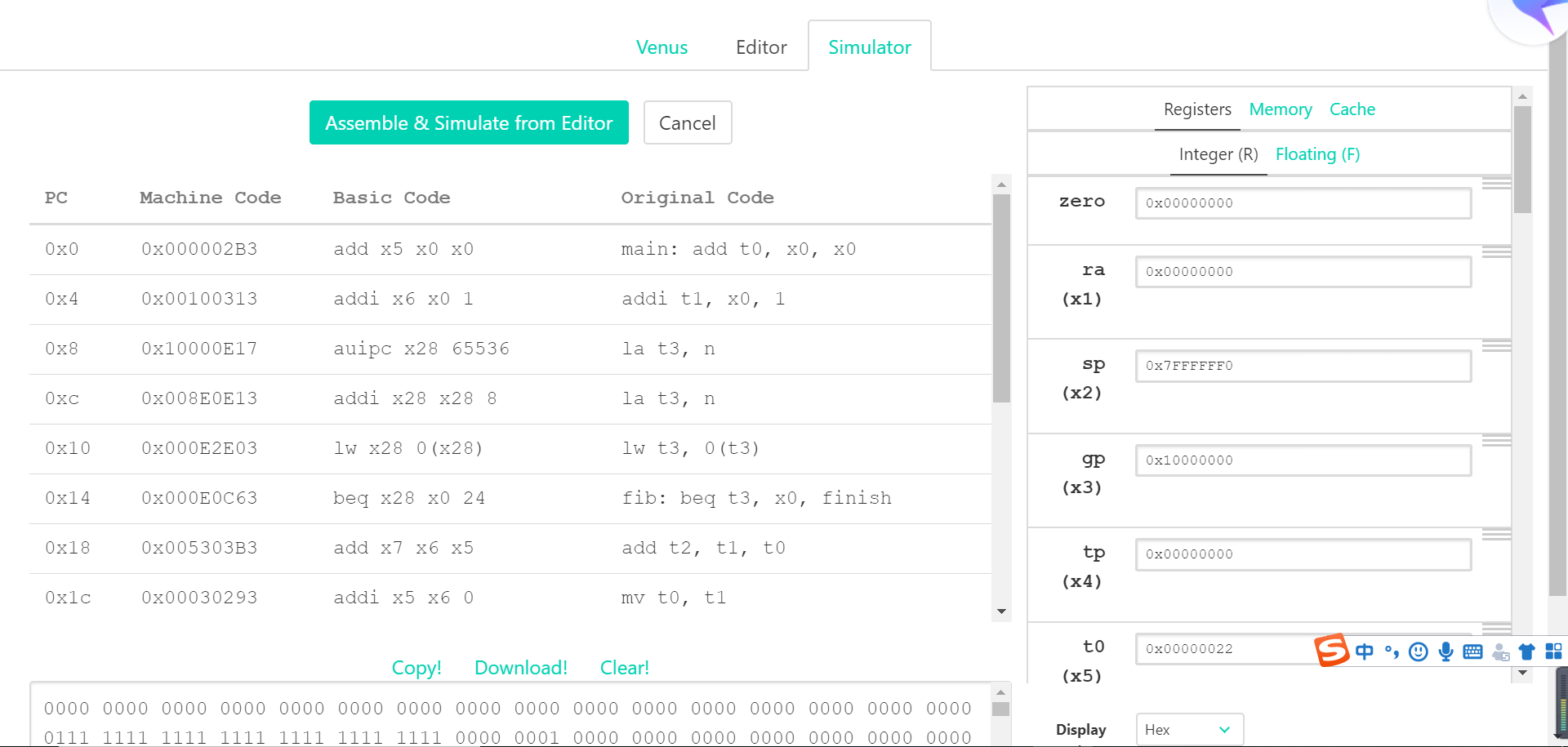
**实验过程：**



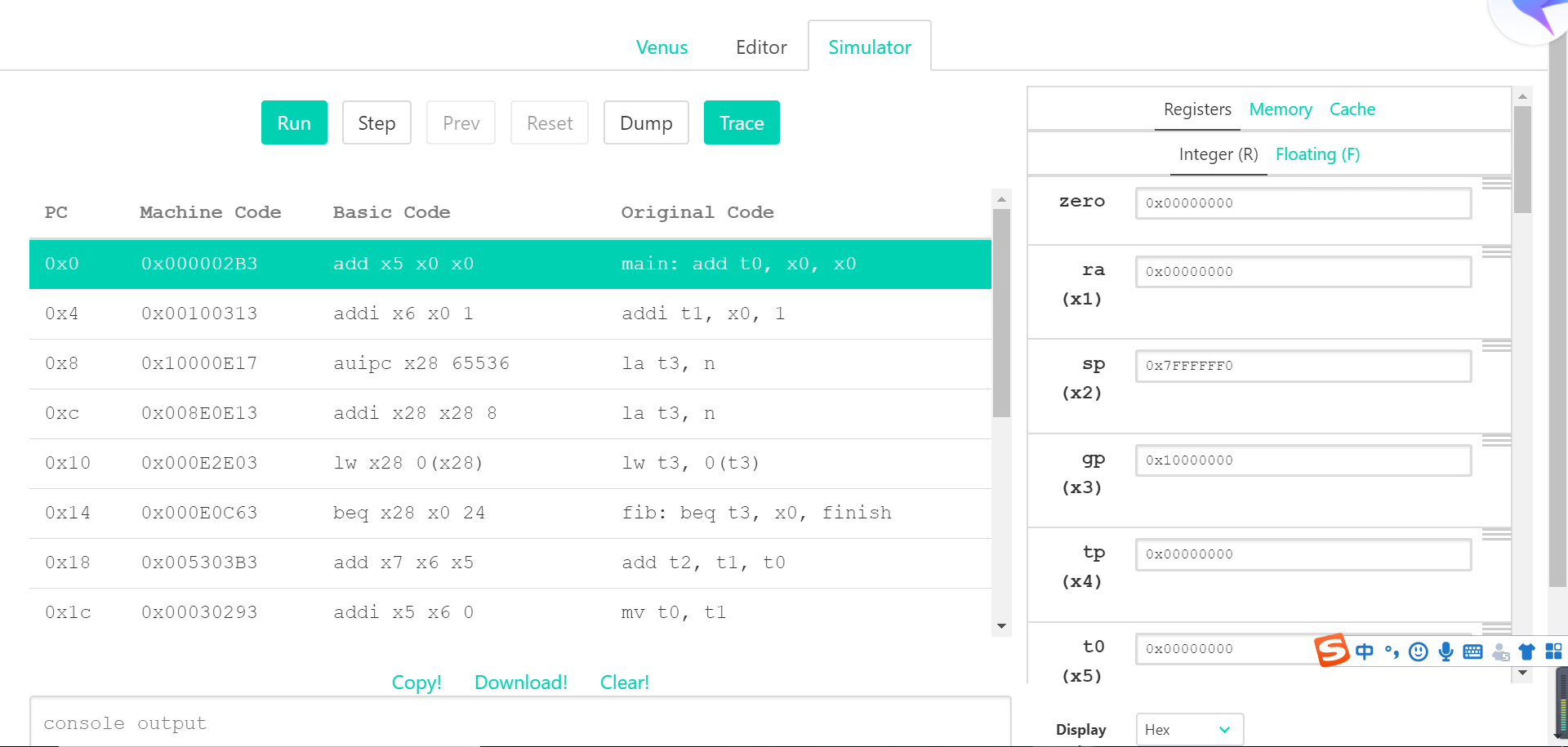
1. 1.将ex1.s的内容粘贴到编辑器中。



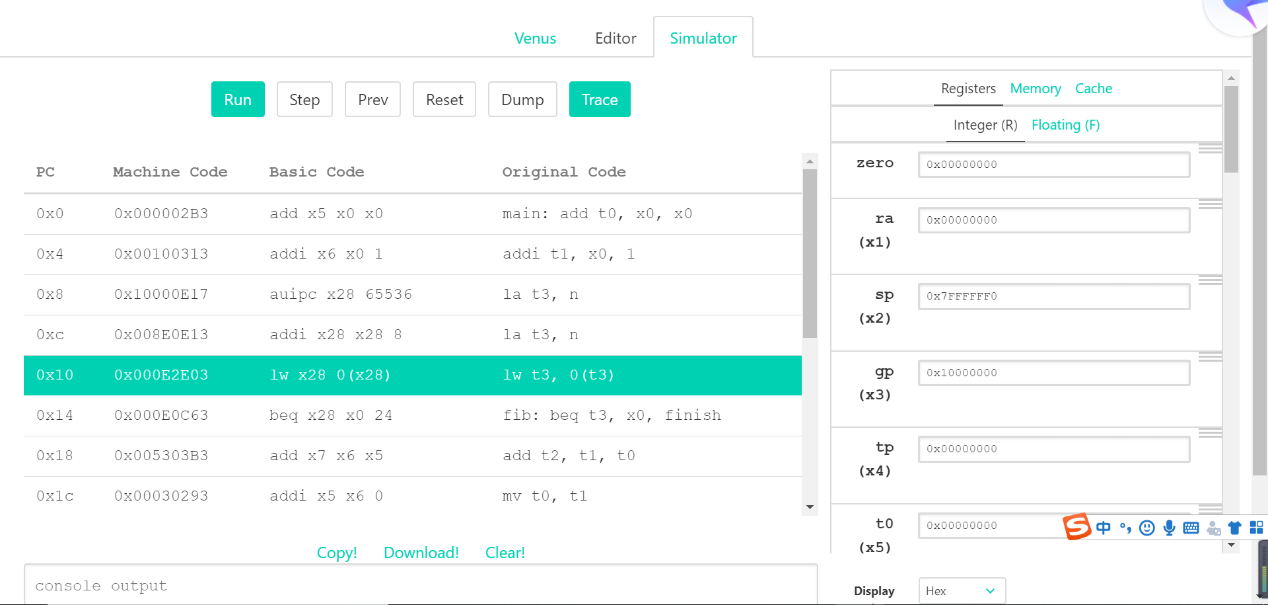
1. 单击“模拟器”选项卡。这将准备您为执行而编写的代码。



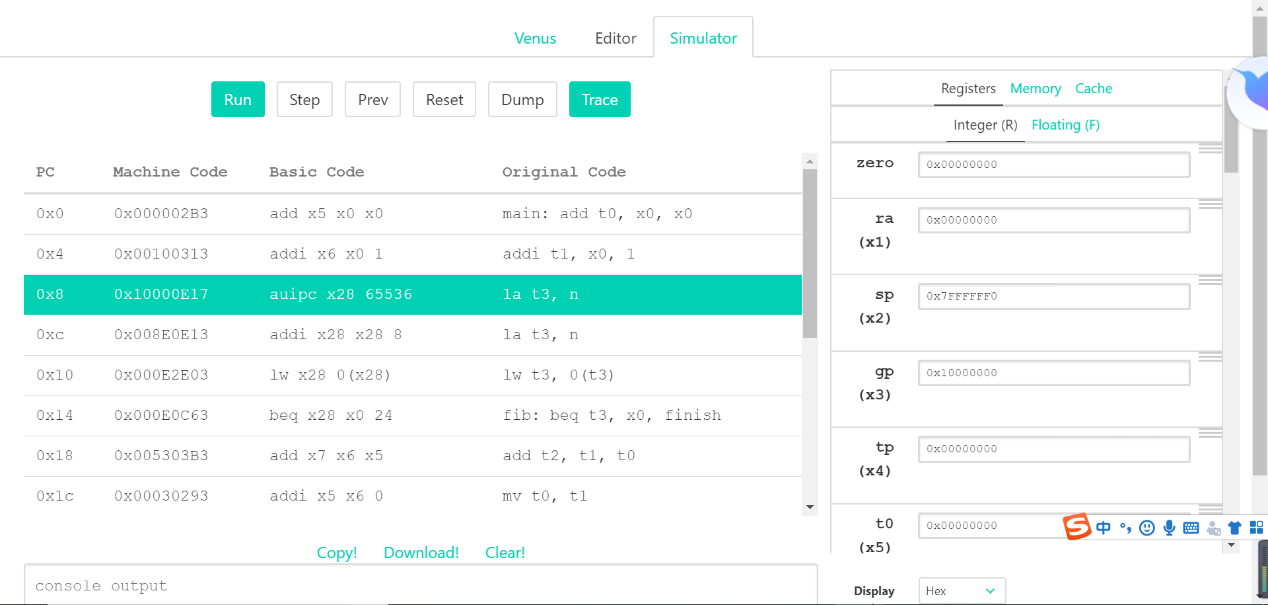
1. 在模拟器中，单击“从编辑器组装和模拟”



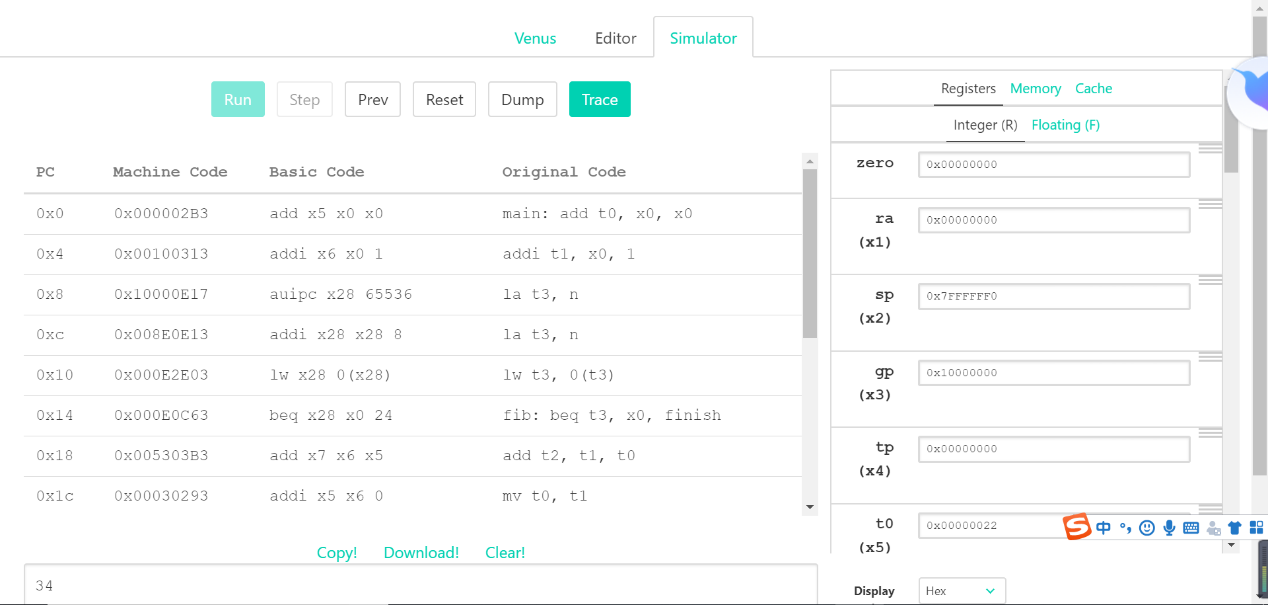
1. 在模拟器中，要执行下一条指令，请单击“步骤”按钮。



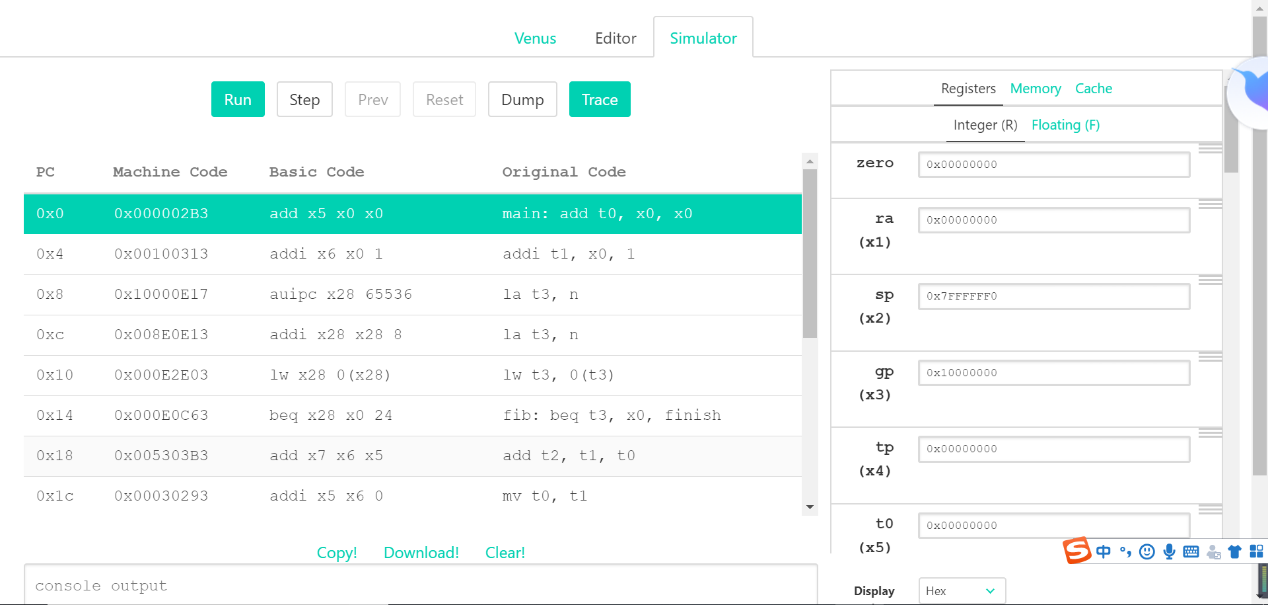
1. 要撤消指令，请单击“上一步”按钮。



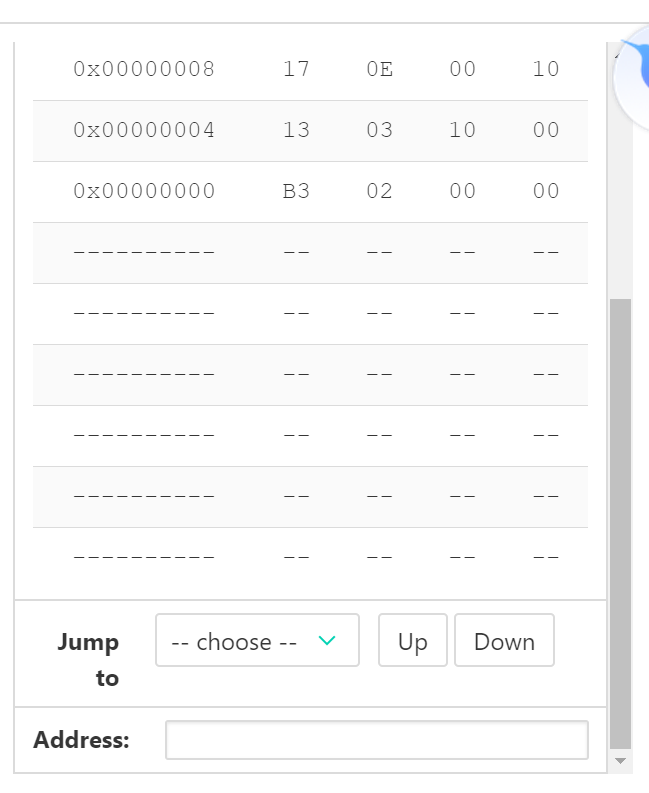
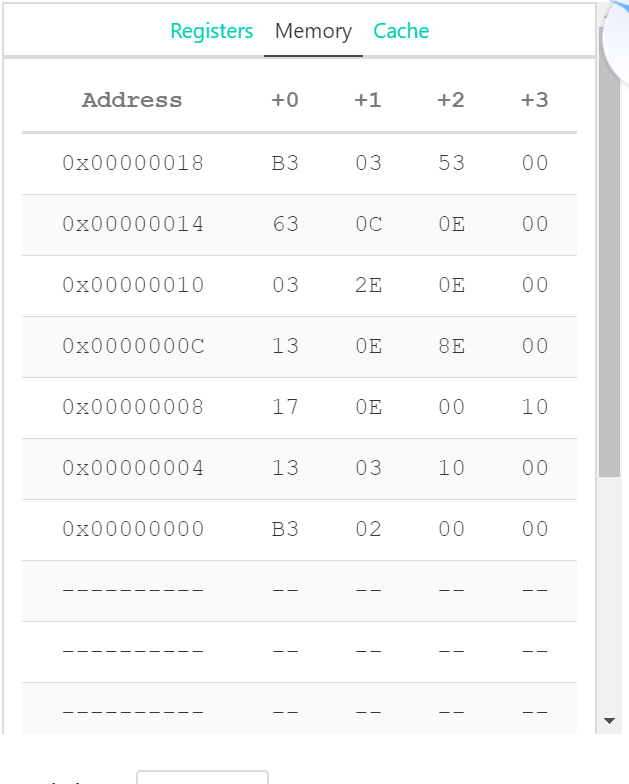
1. 要运行程序以完成，请单击“运行”按钮。



1. 要从头开始重置程序，请单击“重置”按钮。



1. 所有32个寄存器的内容位于右侧，控制台输出位于底部
2. 要查看内存的内容，请单击右侧的“内存”选项卡。可以使用底部的下拉菜单导航到内存的不同部分。



② ex1.s程序的功能是计算斐波那契数列中第9位的值.

最终结果为34。

1. 选取的指令、指令格式和二进制码如下表:

|  |  |  |
| --- | --- | --- |
| 指令 | 指令格式 | 二进制码 |
| addi x6,x0,1 | I-type | 0000 0000 0001 0000 0000 0011 0001 0011 |
| add x5,x0,x0 | R-type | 0000 0000 0000 0000 0000 0010 1011 0011 |
| Ecall | B-type | 0000 0000 0000 0000 0000 0000 0111 0011 |

程序源代码：

Fact.s 文件：

.data

.word 2, 4, 6, 8

n: .word 9

.text

main: add t0, x0, x0

addi t1, x0, 1

la t3, n

lw t3, 0(t3)

fib: beq t3, x0, finish

add t2, t1, t0

mv t0, t1

mv t1, t2

addi t3, t3, -1

j fib

finish: addi a0, x0, 1

addi a1, t0, 0

ecall # print integer ecall

addi a0, x0, 10

ecall # terminate ecall