



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

实验项目



主讲教师：章复嘉

实验六 MIPS 汇编器与模拟器实验

❖ 1、实验目的

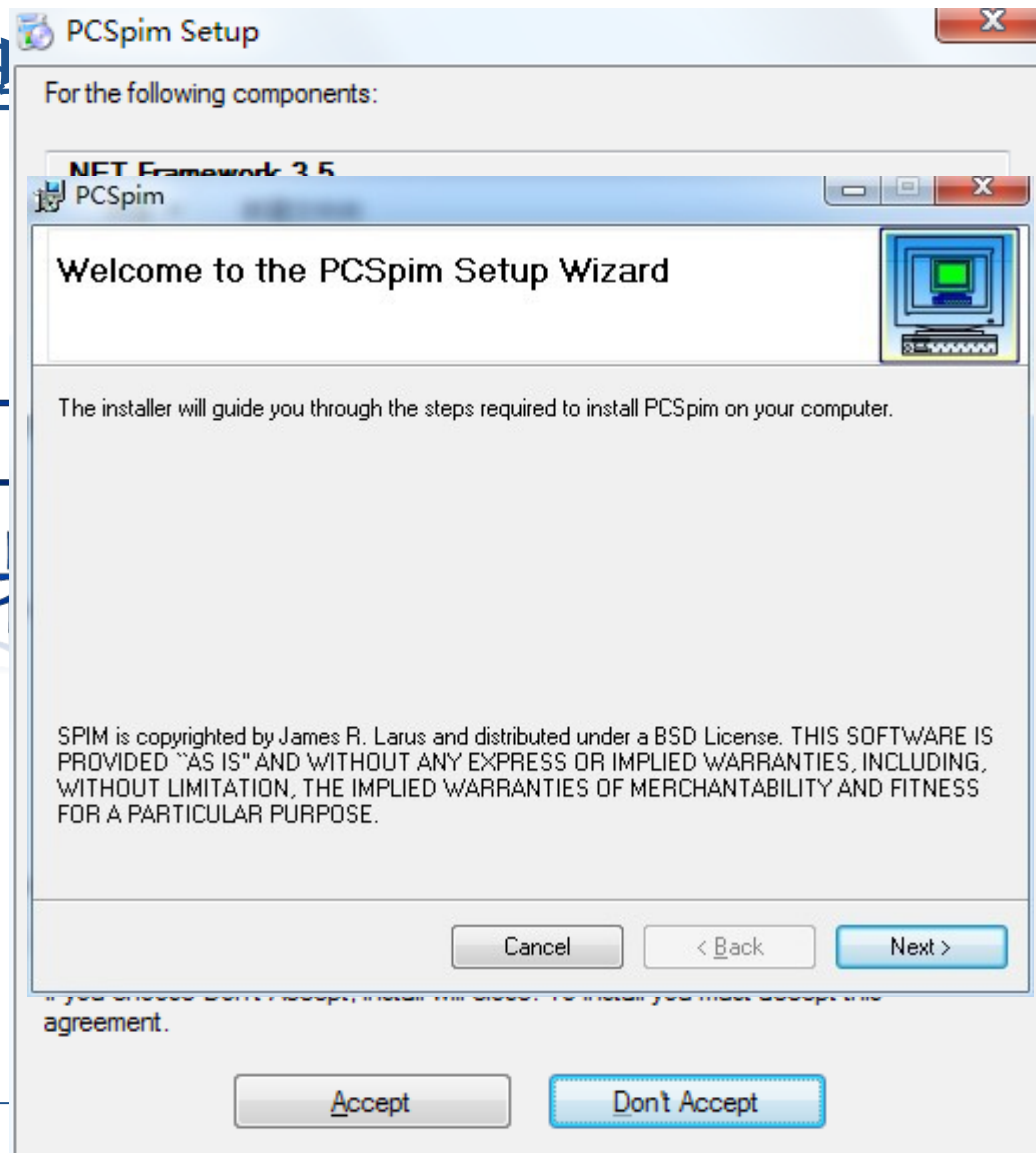
- 学习 MIPS 指令系统，熟悉 **MIPS 指令格式** 及其 **汇编指令助记符**，掌握 **机器指令编码方法**；
- 学习 MIPS 汇编程序设计，学会使用 MIPS 汇编器 **将汇编语言程序翻译成二进制文件**；
- 了解使用 MIPS 教学系统模拟器运行程序的方法；

实验六 MIPS 汇编器与模拟器实验

❖ 2、实验内容与原理

a . PCSpim 下载及安装:

学习MIPS汇编及其模拟
PCspim 模拟器也有很多版本，可以在网上免费下载。注意：最新版本的 Qtspim 不能用于 xp 系统，下载时请查看网页上相关提示信息。
下载地址：



❖ 2、实

实验结果可在
寄存器窗口、
MIPS
汇编程序
窗口、数
据区窗口
、信息输
入窗口。

The screenshot displays the PCSpim MIPS simulator interface. The main window is divided into several sections:

- Registers:** A table showing the status of various registers. The PC is 00000000, EPC is 00000000, Cause is 00000000, and BadVAddr is 00000000. The Status register is 3000fff10, HI is 00000000, and LO is 00000000. General registers R0 through R28 are listed with their current values. A red box labeled "寄存器" (Registers) highlights the R24 register.
- Instructions:** A list of instructions being executed, including lw, addiu, sll, add, jal, nop, and li. A red box labeled "指令" (Instructions) highlights the instruction "addiu \$s1, \$s2, 4".
- DATA:** A section showing memory data at address 0x10000000, with a value of 0x00000000.
- STACK:** A section showing memory data at address 0x7ffff494, with a value of 0x00000000.
- KERNEL DATA:** A section showing memory data at address 0x90000000, with a value of 0x78452020.
- Message:** A section at the bottom showing the SPIM version (8.0 of January 8, 2010) and copyright information. A red box labeled "消息" (Message) highlights this section.

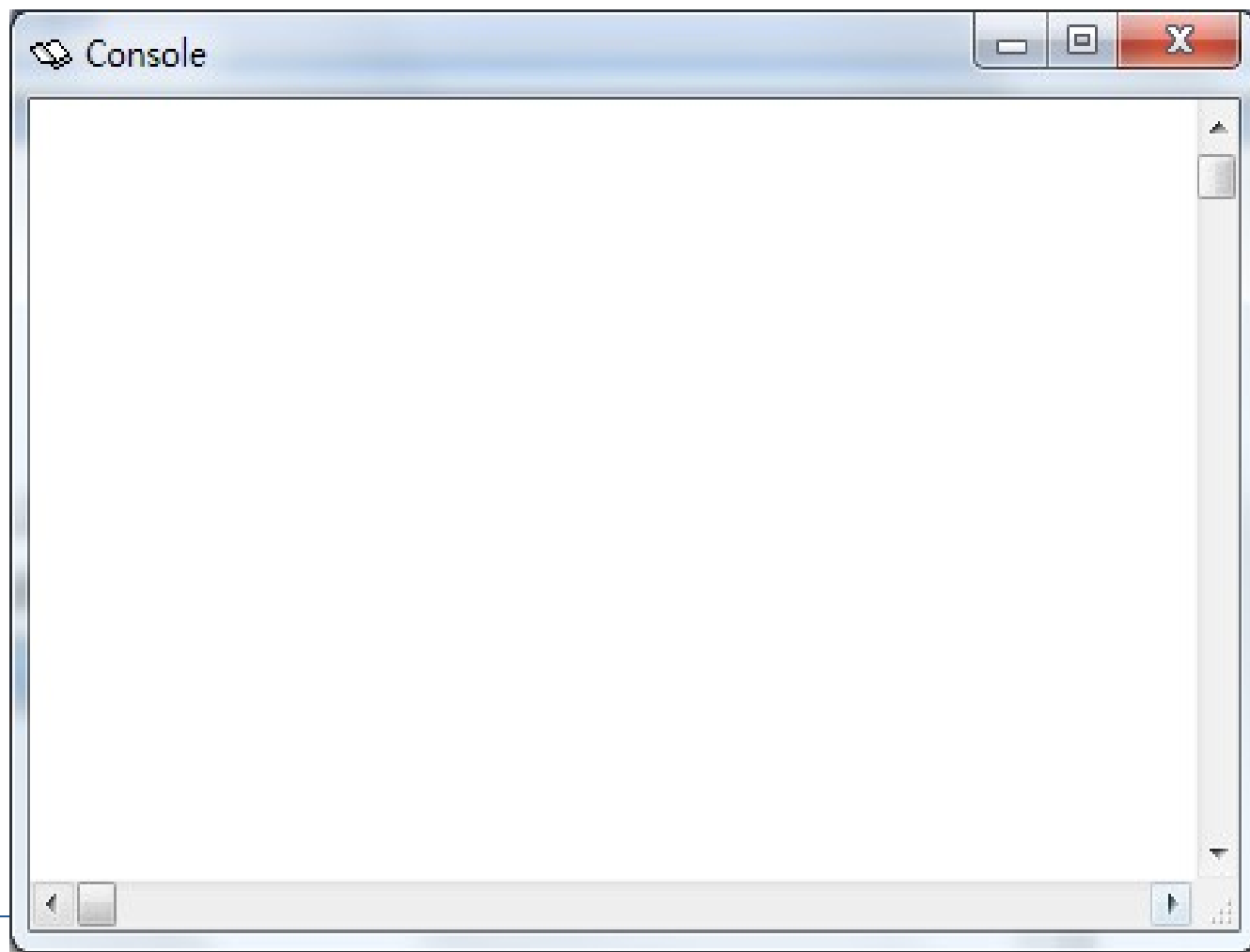
The status bar at the bottom indicates the current PC value (0x00000000), EPC value (0x00000000), and Cause value (0x00000000).

实验六 MIPS 汇编器与模拟器实验

❖ 2、实验内容与原理

除此之外，
还有一个独
立的控制台
窗口

Console，
用于程序信
息的输入、
输出，如图



实验六 MIPS 汇编器与模拟器实验

❖ 2、实验内容与原理

(1) 编辑程序

由于 PCSpim 没有自带的编辑器，用户可以选择自己喜欢的 editor 编辑文件，后缀名为 .asm 或者为 .s。在这里我们用系统自带的文本编辑器命名为 test.asm。具体程序如下：

a. PCSpim 应用实例：

读入两个整数，进行比较，并输出较大的数。

```
main: li $v0,5
```

```
syscall
```

```
move $t0, $v0
```

```
li $v0,5
```

```
syscall
```

```
move $t1, $v0
```

```
bgt $t0, $t1, t0_bigger
```

```
move $t2,$t1
```

```
b endif
```

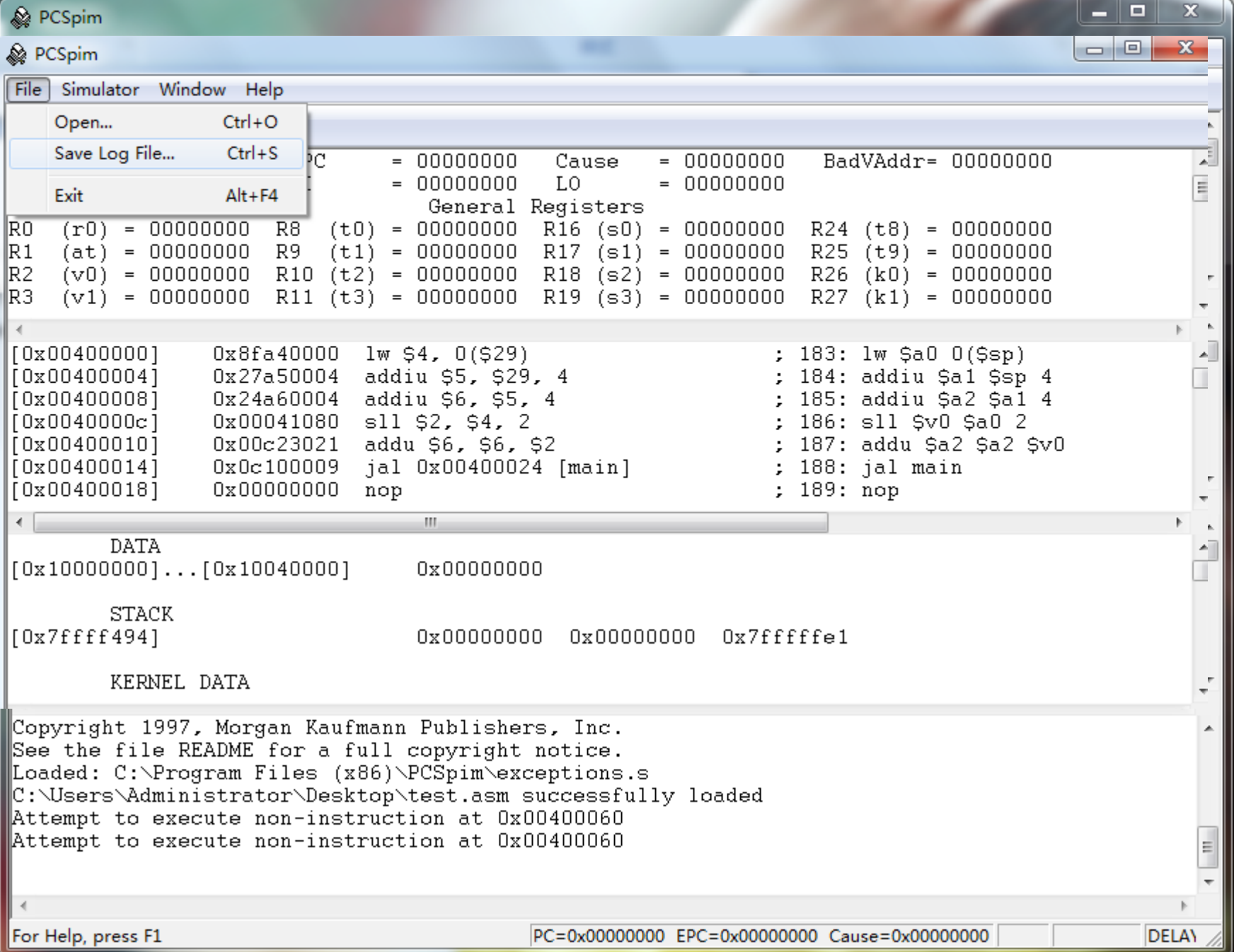
```
t0_bigger: move $t2,$t0
```

```
endif: move $a0,$t2
```

```
li $v0,1
```

```
syscall
```

```
jr $ra;
```



File Simulator Window Help

Open... Ctrl+O

Save Log File... Ctrl+S

Exit Alt+F4

PC = 00000000 Cause = 00000000 BadVAddr = 00000000
= 00000000 LO = 00000000

General Registers

| | | | |
|--------------------|---------------------|---------------------|---------------------|
| R0 (r0) = 00000000 | R8 (t0) = 00000000 | R16 (s0) = 00000000 | R24 (t8) = 00000000 |
| R1 (at) = 00000000 | R9 (t1) = 00000000 | R17 (s1) = 00000000 | R25 (t9) = 00000000 |
| R2 (v0) = 00000000 | R10 (t2) = 00000000 | R18 (s2) = 00000000 | R26 (k0) = 00000000 |
| R3 (v1) = 00000000 | R11 (t3) = 00000000 | R19 (s3) = 00000000 | R27 (k1) = 00000000 |

| | | | |
|--------------|------------|-----------------------|----------------------------|
| [0x00400000] | 0x8fa40000 | lw \$4, 0(\$29) | ; 183: lw \$a0 0(\$sp) |
| [0x00400004] | 0x27a50004 | addiu \$5, \$29, 4 | ; 184: addiu \$a1 \$sp 4 |
| [0x00400008] | 0x24a60004 | addiu \$6, \$5, 4 | ; 185: addiu \$a2 \$a1 4 |
| [0x0040000c] | 0x00041080 | sll \$2, \$4, 2 | ; 186: sll \$v0 \$a0 2 |
| [0x00400010] | 0x00c23021 | addu \$6, \$6, \$2 | ; 187: addu \$a2 \$a2 \$v0 |
| [0x00400014] | 0x0c100009 | jal 0x00400024 [main] | ; 188: jal main |
| [0x00400018] | 0x00000000 | nop | ; 189: nop |

DATA

[0x10000000]...[0x10040000] 0x00000000

STACK

[0x7ffff494] 0x00000000 0x00000000 0x7fffffe1

KERNEL DATA

Copyright 1997, Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Loaded: C:\Program Files (x86)\PCSpim\exceptions.s
C:\Users\Administrator\Desktop\test.asm successfully loaded
Attempt to execute non-instruction at 0x00400060
Attempt to execute non-instruction at 0x00400060

For Help, press F1

PC=0x00000000 EPC=0x00000000 Cause=0x00000000

DELAY

实验六 MIPS 汇编器与模拟器实验

❖ 6.6.3 实验要求

- 按照上述实验内容中的例子，编辑 test.asm 文件，在 PCSpim 中打开它，并使用单步执行和连续执行方式运行该程序；
- 将下列程序输入一个 R_CPU_Test.asm 文件，并在 PCSpim 中单步运行，观察各个寄存器的值，是否和预期的▶致；
- 将上述程序的指令代码逐条摘录出来，拷贝至 ROM IP 核关联文件 *.coe 中，以备后续实验使用；
- 撰写实验报告：含执行结果截图、实验结果记录表、实验分析和生成的 *.coe 文件内容，以及你对本实验的“思考与探索”部分所作的思考与探索；

实验六 MIPS 汇编器与模拟器实验

```
nor $1, $0, $0;    #$1=FFFF_FFFF
sltu $2, $0, $1;    #$2=0000_0001
add $3, $2, $2;     #$3=0000_0002
add $4, $3, $2;     #$4=0000_0003
add $5, $4, $3;     #$5=0000_0005
add $6, $5, $3;     #$6=0000_0007
sllv $7, $6, $2;     #$7=0000_000E
add $9, $5, $6;     #$9=0000_000C
sllv $8, $6, $9;     #$8=0000_7000
xor $9, $1, $8;     #$9=FFFF_8FFF
add $10, $9, $1;    #$10=FFFF_8FFE
sub $11, $8, $7;    #$11=0000_6FF2
sub $12, $7, $8;    #$12=FFFF_900E
and $13, $9, $12;   #$13=FFFF_800E
or $14, $9, $12;    #$14=FFFF_9FFF
or $15, $6, $7;     #$15=0000_000F
```

```
nor $16, $6, $7;    #$16=FFFF_FFF0
add $17, $7, $3;     #$17=0000_0010
sllv $18, $8, $17;   #$18=7000_0000
sllv $19, $3, $17;   #$19=0002_0000
sllv $20, $19, $7;   #$20=8000_0000
add $21, $20, $1;    #$21=7FFF_FFFF
or $22, $18, $21;    #$22=7FFF_FFFF
add $23, $20, $22;   #$23=FFFF_FFFF
sub $24, $20, $22;   #$24=0000_0001
sub $25, $22, $20;   #$25=FFFF_FFFF
xor $26, $18, $1;    #$26=8FFF_FFFF
sltu $27, $22, $20;  #$27=0000_0001
sltu $28, $26, $20;  #$28=0000_0000
add $29, $22, $2;     #$29=8000_0000
sub $30, $20, $2;     #$30=7FFF_FFFF
add $31, $11, $26;   #$31=9000_6FF1
```




实验六 MIPS 汇编器与模拟器实验


❖ 6.6.4 实验步骤

- (1) 使用记事本程序或任何纯文本编辑器，编辑 test.asm 文件，输入前述内容；
- (2) 运行 PCSpim 程序，在其中打开 test.asm，先连续执行，输入起始地址 0x0040 0000，再单步运行，按照需要在控制台输入 2 个数据，执行完毕，观察结果。
- (3) 使用记事本程序或任何纯文本编辑器，编辑 R_CPU_Test.asm 文件，输入规定指令。
- (4) 在 PCSpim 程序中打开 R_CPU_Test.asm，同上单步执行，记录执行结果。

实验六 MIPS 汇编器与模拟器实验

❖ 6.6.5 思考与探索：必做（1）

- （1）将 R_CPU_Test.asm 汇编程序执行的结果记录到表 6.13 中，分析你的实验结果是否正确；如果不正确，请分析原因；
- （2）谈谈你在实验中，碰到了什么问题？又是怎么解决的？
- （3）PCSpim 能模拟执行 MIPS 的汇编程序。还有其他真正意义上的 MIPS 汇编器，将汇编源程序直接翻译成一个只含机器码（指令编码）的代码文件，甚至还有将机器码直接自动生成 *.coe 的小工具，查找相关资料，把你的新发现和大家分享，写实验报告里。



The End!