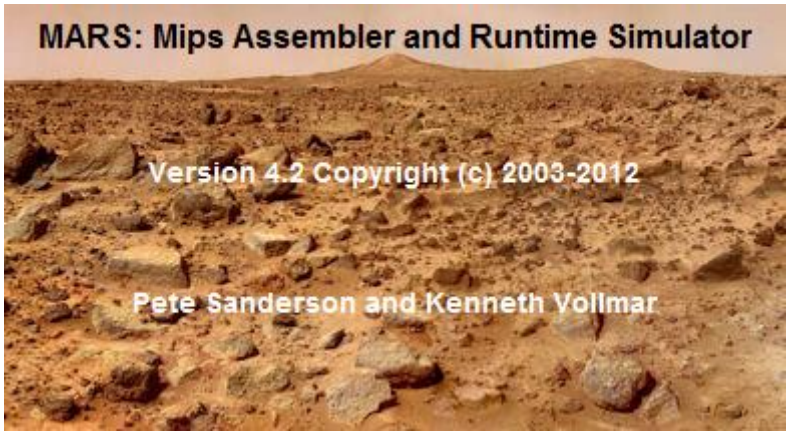
1、 写汇编语言程序的工具推荐使用 MARS，课程文件里面上传的 Mars.jar 就是了

2、 运行 jar 需要 java 环境，所以要使用 MARS 还需要安装 jdk，课程文件里面上传的 jdk-7-windows-i586.exe 就是

3、 下载下来这两个文件，先安装 jdk，安装过程中也没有特别需要说明的地方。想更改安装路径的话可以修改一下(会出现两次)，不过为了避免可能出现的问题，路径不要带中文字符。安装好会得到两个文件夹。

4、 右键单击 Mars.jar，选择打开方式，选择默认程序，定位到刚才选择的安装路径下的\Java\jre7\bin 文件夹，我这里完整的路径为 D:\Program Files (x86)\Java\jre7\bin，选择 javaw.exe，确定。以后再想运行 Mars 双击就可以了。



这里需要强调一下，在我的电脑上，Mars.jar 的放置位置路径是不能有中文的，否则会出现运行 Mars.jar 没反应的情况。

5、 有关 MIPS 汇编的语法，我上传了“5_程序与指令系统_2.pdf”这个文件。该文档为电子工程系罗嵘老师 数字逻辑与处理器基础 课程的课件，这里我们拿来学习一下，也对罗嵘老师表示感谢。

6、 .data 定义数据段，可以理解为流水线中的数据存储器。.data 结合紧跟其后的可选的地址，可以将不同的数据块存储在不同的区域，可能会方便计算访问地址。举个例子：
.data 0x10000000
.word 0x2b7e1516，0x28aed2a6，0xabf71588，0x09cf4f3c，0x3243f6a8，0x885a308d，0x313198a2 , 0xe0370734

.data 0x100000c0
.word
0x01000000,0x02000000,0x04000000,0x08000000,0x10000000,0x20000000,0x40000000,0x80000000,0x1b000000,0x36000000

.data 0x10000200
.word ……

7、 .text 定义指令段，可以理解为流水线中的指令存储器。也就是大家写汇编程序的地方。按照这个课件中所说，.text 后面紧跟的地址就定义为 0x00400000 吧

8、 label: certain instruction
certain instruction
……
汇编程序总会存在判断、跳转（类似 bne、beq 这类指令）或者是无条件跳转（类似 j、jr 这类指令）因为具体地址不方便预先确定，所以使用标签是一个很好的方法，比如说

bnez $s1, loop

这条指令，如果$s1 中数值不为 0，那么就跳转到标签为 loop 的地方，否则顺序执行下一条指令。

9、 有关栈操作，需要 jal 指令来配合，作业中对这个指令没有要求，所以栈操作可以根据情况决定是否采用。

10、 在这个课件中的一些内容，比如说系统调用指令的说明、最后的排序算法，可以先不管它。对于这次的大作业，用要求实现的指令集来做是足够的。如果觉得某个地方某条未要求指令使用起来要更方便，可以在流水线中把对这条指令的支持加进去。

11、 早先的时候，看到有一个网页对 MIPS 其中的 31 条指令做了总结，感觉使用起来较为方便，我保存成网页也一并上传到学堂上了，大家可以参考一下。

12、 另外类似地，需要查看指令的使用方式的，有这么几种途径：

a) 在 Mars 中按 F1，在 MIPS 这一栏可以查看
b) 之前上传了文档 mips32v2 指令集.pdf，这里面带有链接功能，在目录中找到后点击可以定位到对应位置
c) 如果有教材的话，教材的附录 471 页开始有对于指令的说明。除此之外，对于这 32 个寄存器各自的用途在 53 页有相关说明。方便没有教材的同学使用，我做个表格写在这里吧。

名称	寄存器号	用途	调用中是否保存
$zero	0	常数 0	n.a.
$v0~$v1	2~3	结果值和表达式求值	否
$a0~$a3	4~7	参数	是
$t0~$t7	8~15	临时变量	否
$s0~$s7	16~23	保存	是
$t8~$t9	24~25	其他临时变量	否
$gp	28	全局指针	是
$sp	29	栈指针	是
$fp	30	帧指针	是
$ra	31	返回地址	是

注：寄存器 1 称为$at，受汇编器的保护；寄存器 26~27 称为$k0~$k1，受操作系统的保护

13、 在 Mars 中打开一个测试程序，test.asm，首先编译一下（菜单栏 run 下面的 assemble，快捷键 F3），下面的信息窗口中显示 Assemble: operation completed successfully. 说明编译没有问题，可以执行了（菜单栏 run 下面的 go，快捷键 F5）。执行后的结果可以在对应的寄存器中查看。对于罗老师课件中的这个程序，是要求 4!，最后的结果保存在寄存器$s0 中，值为 0x18.

14、 上传了两张 Mars 使用的图片，用红框圈出来的是使用中信息来源位置。

15、 在编译通过后，可以设置断点(在 execute 这一栏，每一行前有 Bkpt 可以勾选)，运行到这里会停止，结合单步执行，可以查看关注位置附近的寄存器和存储区域值的变化。也方便阶段性地与标准结果进行比对。

16、 调试通过后，菜单栏选 File->Dump Memory，Dump Format 根据需要选择 Binary Text 或者 Hexadecimal Text 可以将自己的汇编代码转成对应的机器码。

17、 现在拿到的机器码光秃秃的，需要进一步的处理，因为行数比较多，可以考虑借助 MATLAB 或者其他软件对每行数据做一个包装，前面加个头，后面加个分号，变成 icache 中可用的指令。向下面这样

```
RAM[0] = 32'h34101000;
RAM[1] = 32'h00108400;
RAM[2] = 32'h22110200;
RAM[3] = 32'h22070010;
……
```

18、 写的汇编对应机器码较长的话，可能会存在原始 icache.v 定义的存储区域放不下的问题，因为不是真的在 FPGA 上面做，所以可以自行对存储区进行扩展。
指的是这一行

```
reg[31:0]  RAM[255:0];
```

19、 在后面写 Verilog 的过程中，需要注意机器码中跳转指令的地址要和.v 中对于存储区访问地址匹配，如果有必要的话，导出的机器码可以修改跳转指令的地址，从而让两者统一。

20、 对于这次的大作业，最好先把 MIPS 汇编程序调试通过了，然后在 Verilog 实现 5 级流水线的时候，需要支持哪些指令，可能会有更好的认识。在写流水线的时候，先把要支持的指令都测试一下，看看功能是不是正常，然后再把自己所写的 AES 汇编对应的机器码导入到 icache 中，进行调试，这样更有把握一些。总的来说，一步一步做，前一步做对了，后一步才有保障。

21、 很有意思，但工作量也不小。

22、 在完成过程中的问题可以到讨论区或者微信群提问，我时不时也会上去看看。

助 教