

MIPS-lite Simulator

《计算机系统结构》课程实验用户操作手册

kde9

kfirst 孔祥欣¹

deepsolo 胡玮玮²

edwardtoday 卿培³

2009/6/2

¹ 2006011299, CS62, Mobile: 13401086576, Email: kxx006@gmail.com

² 2006011293, CS62, Mobile: 13810313760, Email: huww06@gmail.com

³ 2006011291, CS62, Mobile: 15901033612, Email: edwardtoday@gmail.com

目录

1 启动界面介绍.....	5
2 主要功能.....	5
2.1 文件操作.....	5
2.1.1 新建文件.....	6
2.1.2 打开文件.....	6
2.1.3 保存文件.....	6
2.2 代码编辑.....	6
2.2.1 编辑代码.....	6
2.2.2 文本行号与指令条数提示.....	6
2.3 编译执行.....	6
2.3.1 编译.....	6
2.3.2 单步执行.....	6
2.3.3 连续执行.....	7
2.3.4 正在执行的指令提示.....	7
2.3.5 执行到指定位置.....	7
2.3.6 停止执行.....	7
2.3.7 执行周期和指令数统计.....	7
2.3.8 查看代码.....	7
2.4 结果查看.....	7
2.4.1 结果综述.....	7

2.4.2 寄存器结果.....	8
2.4.3 内存结果.....	8
2.4.4 Cache 结果.....	9
3 操作说明.....	9
3.1 文件操作.....	9
3.1.1 新建文件.....	9
3.1.2 打开文件.....	10
3.1.3 保存文件.....	11
3.1.4 另存为文件.....	11
3.2 代码编辑.....	12
3.2.1 编辑代码.....	12
3.2.2 文本行号与指令条数提示.....	12
3.3 编译执行.....	13
3.3.1 编译.....	13
3.3.2 单步执行.....	14
3.3.3 连续执行.....	15
3.3.4 正在执行的指令提示.....	16
3.3.5 执行到指定位置.....	16
3.3.6 停止执行.....	17
3.3.7 执行周期和指令数统计.....	18
3.3.8 查看代码.....	18
3.4 结果查看.....	20

3.4.1 结果综述.....	20
3.4.2 寄存器结果.....	21
3.4.3 内存结果.....	23
3.4.4 Cache 结果.....	26
4、常见问题解答.....	28
4.1 问：可以导出二进制代码文件吗？.....	28
4.2 问：可以导入二进制代码文件吗？.....	28
4.3 问：为什么点任何按钮都无反应？.....	28
4.4 问：为什么修改寄存器或者内存的值但却未修改成功？.....	28
4.5 问：为什么有跳转语句的程序执行结果会有问题？.....	28
4.6 问：为什么我的程序通不过编译？.....	28
4.7 问：请问如何联系你们？.....	29

1 启动界面介绍

启动 MIPS-lite Simulator 以后，主界面如图 1 所示：

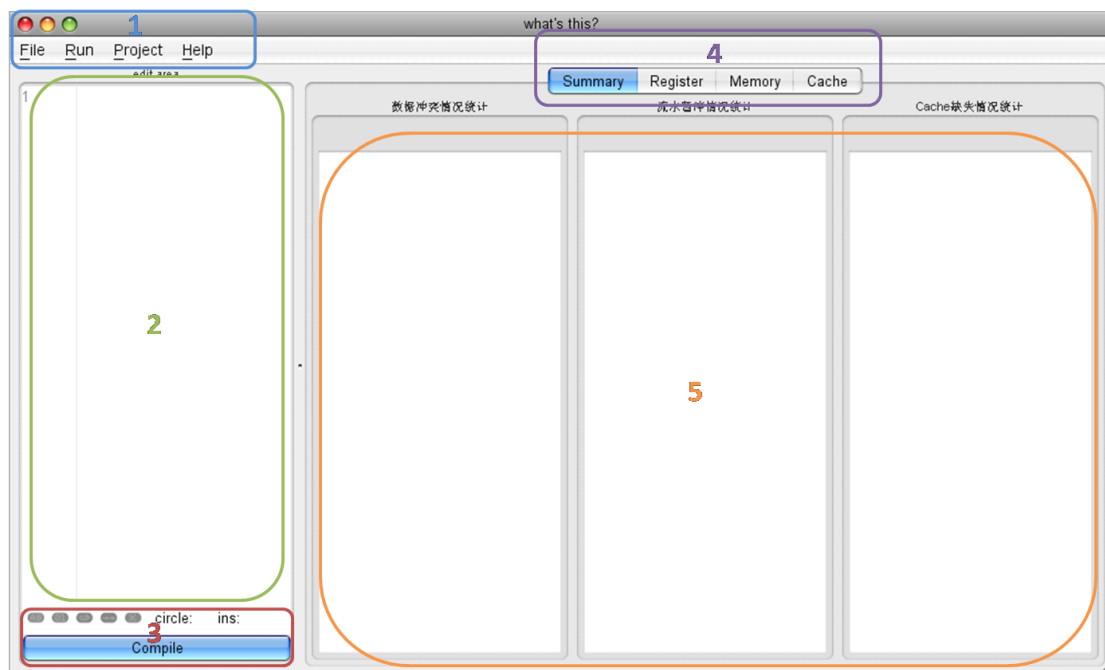


图 1

其中界面主要分成 5 个部分，在图 1 中已经用不同颜色的矩形框标识出来。其中，

1 为主菜单部分，包括了 File、Run、Project、Help 部分，提供了模拟器的各种操作；

2 为文本编辑框，可以直接在该框内输入汇编指令，也可以打开已有的文件将汇编指令显示在此处；

3 为操作模拟器的基本主要按钮，包括编译、单步执行、执行、停止等；

4 为结果显示选择按钮，可以分别进行综述、寄存器、内存、Cache 等的详细信息查看；5 为具体结果显示界面，分别对应 4 中的按钮，当 4 选择不同按钮时，这一部分显示不同。

2 主要功能

2.1 文件操作

2.1.1 新建文件

功能描述：新建文件在文本编辑框内进行代码输入。

2.1.2 打开文件

功能描述：打开已经存在的代码文件。

2.1.3 保存文件

功能描述：保存修改的代码，另存为新的文件或者保存原文件。

2.2 代码编辑

2.2.1 编辑代码

功能描述：在文本框内进行代码编辑。

2.2.2 文本行号与指令条数提示

功能描述：将文本行号与指令条数进行对应，方便查看冲突信息中的指令。

2.3 编译执行

2.3.1 编译

2.3.1.1 对汇编指令进行编译

功能描述：将汇编指令编译为二进制代码。

2.3.1.2 指令错误提示

功能描述：提供简单的指令错误提示，可以查出“不存在的指令”、“寄存器编号错误”、“寄存器格式错误”、“指令格式错误”、“立即数格式错误”等错误。

2.3.2 单步执行

功能描述：编译完成以后，单步执行程序，方便看出寄存器、内存、Cache 等的即时变化。

2.3.3 连续执行

功能描述：连续执行程序，迅速得到执行结果。

2.3.4 正在执行的指令提示

功能描述：方便查看当前执行的指令。

2.3.5 执行到指定位置

功能描述：换一种方式的“断点”功能，方便调试。

2.3.6 停止执行

功能描述：防止出现死循环的时模拟器卡死。

2.3.7 执行周期和指令数统计

功能描述：提示程序执行的周期数和指令数，方便查看各个阶段的平均 CPI。

2.3.8 查看代码

2.3.8.1 查看指令

功能描述：查看汇编指令。

2.3.8.2 查看二进制代码

功能描述：查看编译以后的二进制代码。

2.4 结果查看

2.4.1 结果综述

2.4.1.1 显示数据冲突情况

功能描述：显示程序执行过程中数据冲突的详细情况。

2.4.1.2 显示流水暂停情况

功能描述：显示程序执行过程中流水暂停的详细情况。

2.4.1.3 显示 CACHE 缺失情况

功能描述：显示程序执行过程中 Cache 缺失的详细情况。

2.4.2 寄存器结果

2.4.2.1 显示 32 个寄存器当前的结果以及过去两次结果

功能描述：直观的显示 32 个寄存器的值以及过去两次历史结果，方便查询程序执行结果。

2.4.2.2 最近被修改的寄存器提示

功能描述：方便查看最近被修改的寄存器值。

2.4.2.3 寄存器值修改

功能描述：方便用户调试使用。

2.4.2.4 十进制和十六进制两种显示方式

功能描述：方便用户查看寄存器结果。

2.4.3 内存结果

2.4.3.1 查看指定内存地址的值

功能描述：输入指定内存地址，则可以查看该地址附近的若干内存地址的值。

2.4.3.2 清空、保存以及恢复内存值

2.4.3.4 显示最近读取和写入的内存地址历史值

功能描述：返回查看内存的修改情况。

2.4.3.4 对内存进行自动跟踪

功能描述：方便查看程序执行过程中的内存变化情况。

2.4.3.5 同时修改 CACHE 值

功能描述：在修改内存值的同时修改对应的 Cache 值。

2.4.3.6 内存与 CACHE 值同步提示

功能描述：提示当前内存值与对应 Cache 值是否同步。

2.4.3.7 内存值修改

功能描述：方便修改内存值进行调试。

2.4.4 CACHE 结果

2.4.4.1 分别显示指令 CACHE 和数据 CACHE 的值以及对应的内存地址

功能描述：查看 Cache 的值，方便查看程序运行的结果。

2.4.4.2 具有清空、保存以及恢复指令 CACHE 的功能

2.4.4.3 具有清空、保存以及恢复数据 CACHE 的功能

2.4.4.4 十进制和十六进制两种显示方式

功能描述：方便用户通过十进制和十六进制两种方式进行查看程序运行结果。

3 操作说明

3.1 文件操作

3.1.1 新建文件

点击 File 菜单下的 New 按钮，如图 2 所示：



图 2

3.1.2 打开文件

点击 File 菜单下的 Open 按钮，如图 3 所示：



图 3

此时可以打开文件选择器，如图 4 所示：

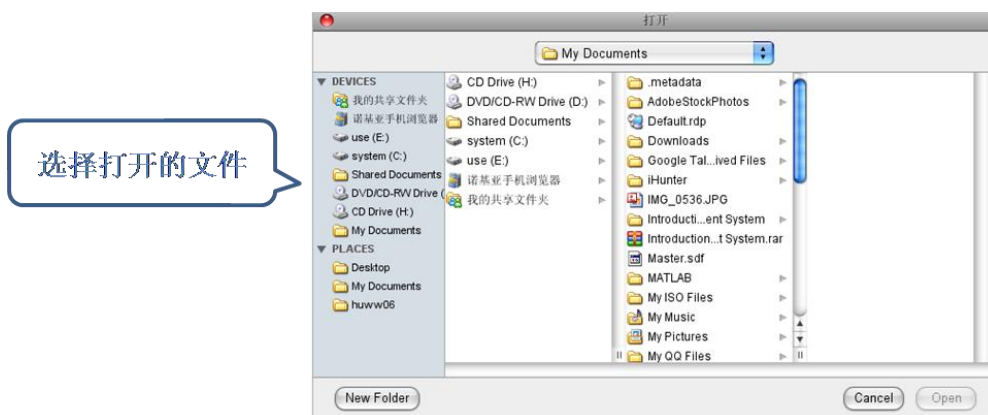


图 4

然后选择打开的文件点击 Open 即可。

3.1.3 保存文件

点击 File 菜单下的 Save 按钮，如图 5 所示：



图 5

如果该文件已经被保存过，那么将直接被保存在原有的文件中。否则，弹出文件选择器，如图 6 所示：

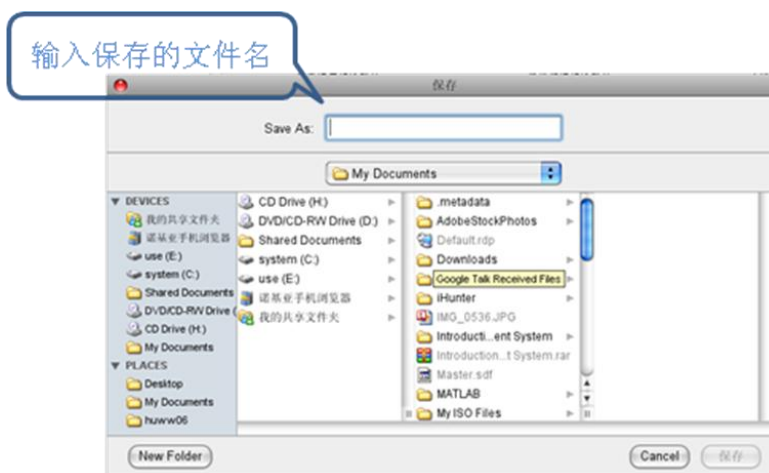


图 6

选择保存按钮即可保存文件。

3.1.4 另存为文件

点击 File 菜单下的 Save As 按钮，如图 7 所示：

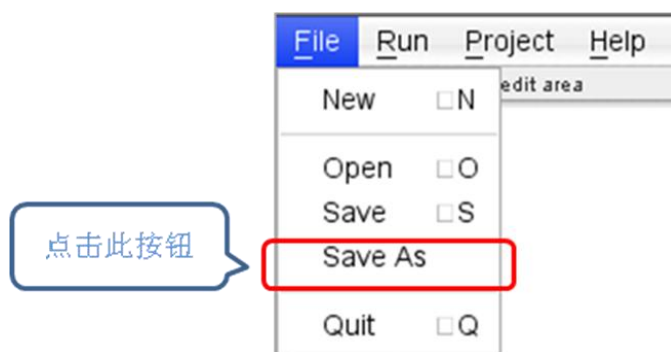


图 7

将弹出如图 6 所示文件选择器，输入文件名，点击保存按钮即可保存文件。

3.2 代码编辑

3.2.1 编辑代码

当 Compile 按钮处于按下状态时，即可在文本编辑框内进行文本输入，如图 8 所示：

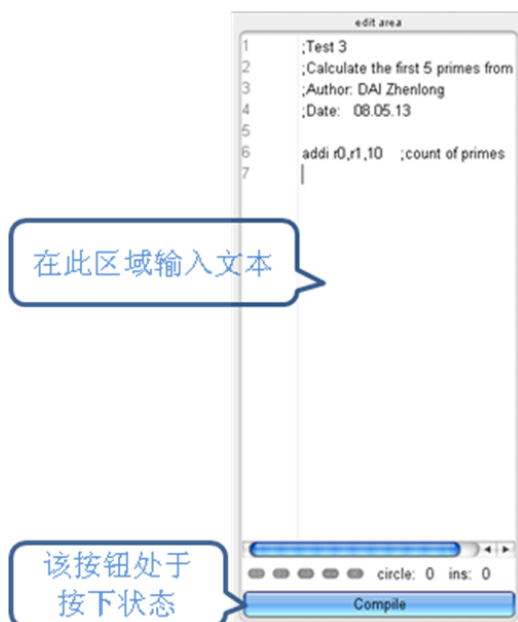


图 8

3.2.2 文本行号与指令条数提示

当 Compile 按钮未被按下且文本框中有指令时，同时显示文本行号和指令条数，如图 9 所示：

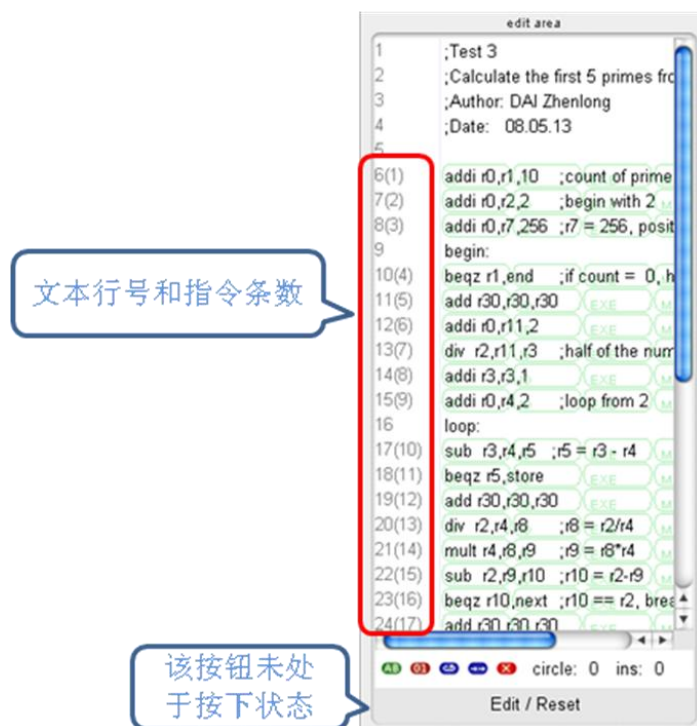


图 9

3.3 编译执行

3.3.1 编译

3.3.1.1 对汇编指令进行编译

输入指令以后，点击 Compile 按钮即可编译，如图 10 所示：

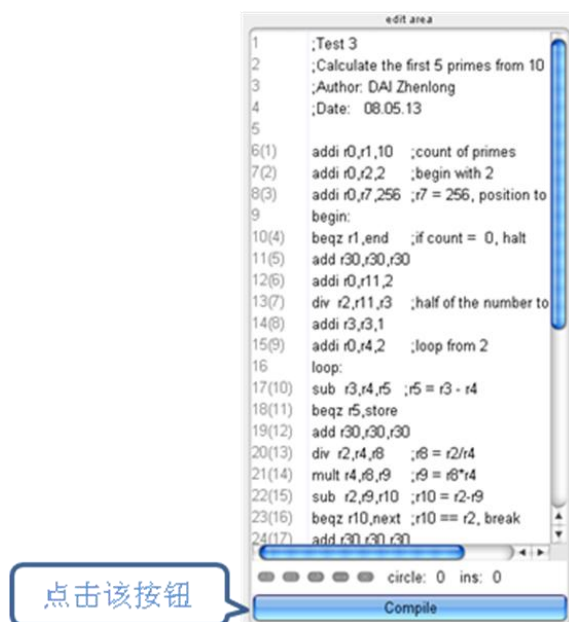


图 10

3.3.1.2 指令错误提示

如果程序代码有错，那么在点击编译的时候会出现错误提示，如图 11 所示：

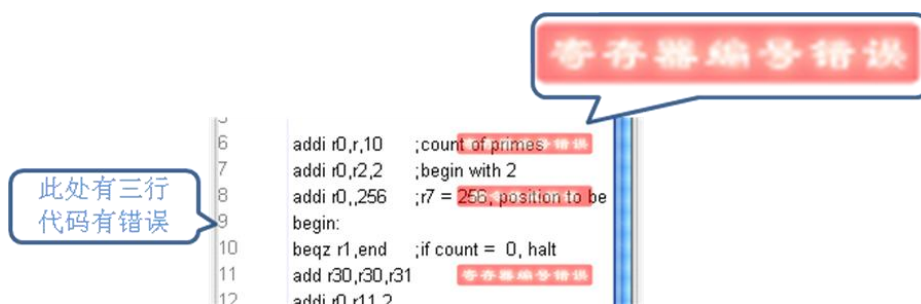


图 11

3.3.2 单步执行

编译完成以后，点击 Run 菜单中的 Run One Step，可以进行单步执行，如图 12 所示：

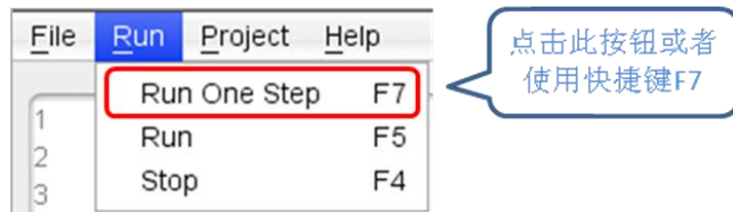


图 12

也可以点击主页面上的单步执行按钮，如图 13 所示：



图 13

3.3.3 连续执行

编译完成以后，点击 Run 菜单中的 Run，可以进行连续执行，如图 14 所示：

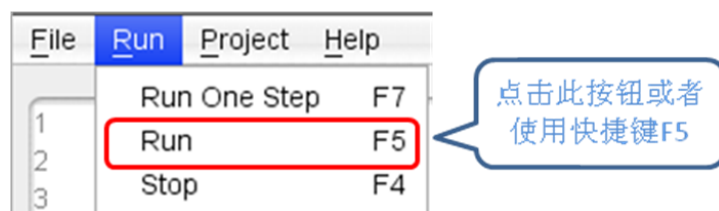


图 14

也可以点击主页面上的连续执行按钮，如图 15 所示：



图 15

3.3.4 正在执行的指令提示

编译完成以后，文本框内针对每一条指令将会产生五个流水段提示，如图 16 所示：

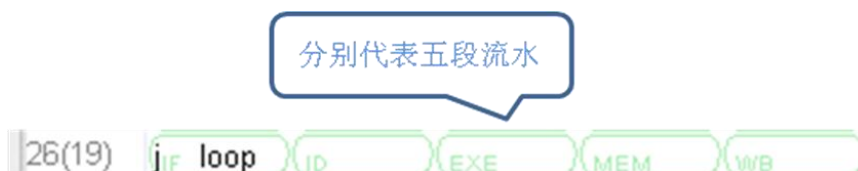


图 16

执行程序，当执行到某一条指令的某一个阶段时，会给出提示，如图 17 所示：

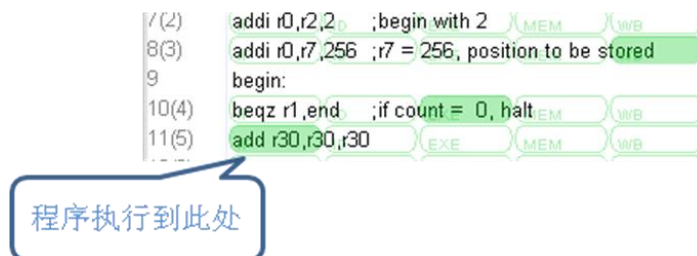


图 17

3.3.5 执行到指定位置

程序执行时，点击其中任意一个流水段，程序将会执行到此处，如图 18、19 所示：

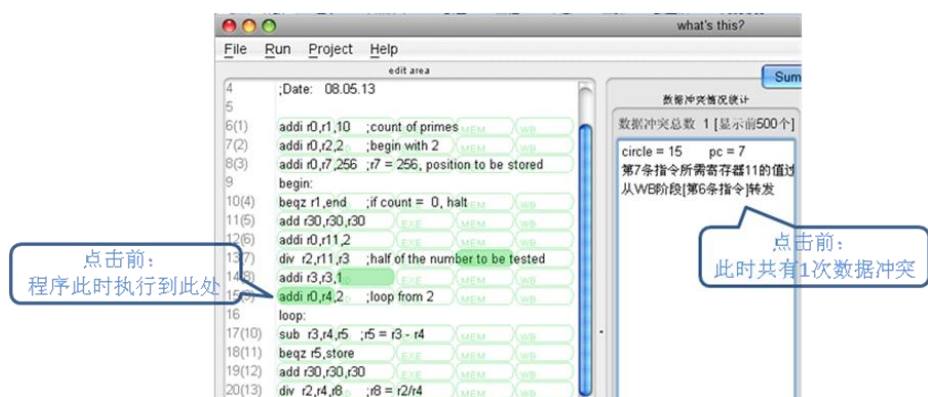


图 18

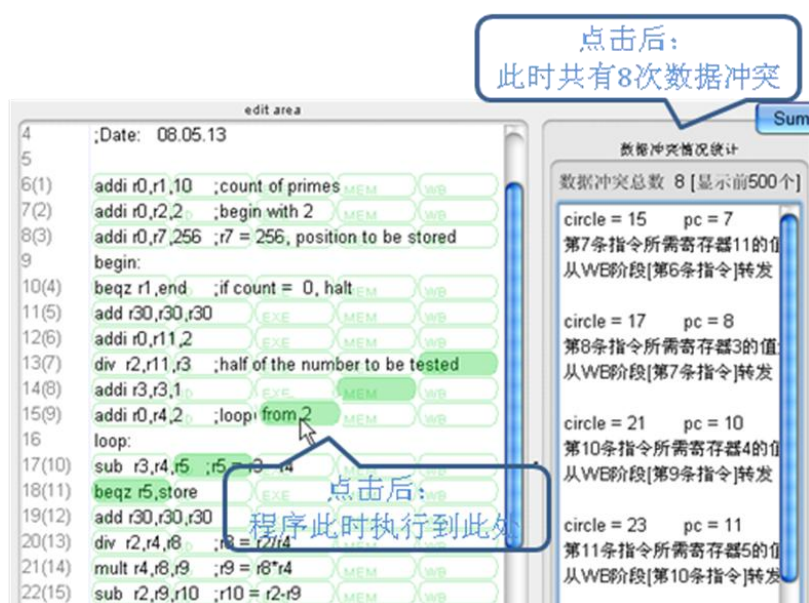


图 19

3.3.6 停止执行

当程序进入了死循环不能退出时，这时候需要停止执行。点击 Run 菜单的 Stop 按钮，如图 20 所示：

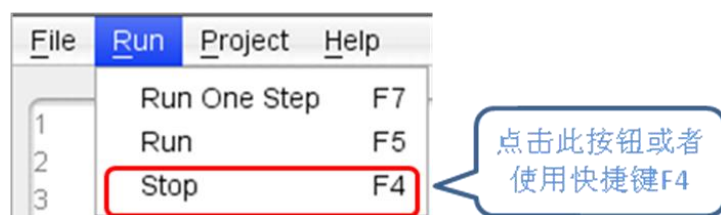


图 20

也可以点击主页面上的停止执行按钮，如图 21 所示：

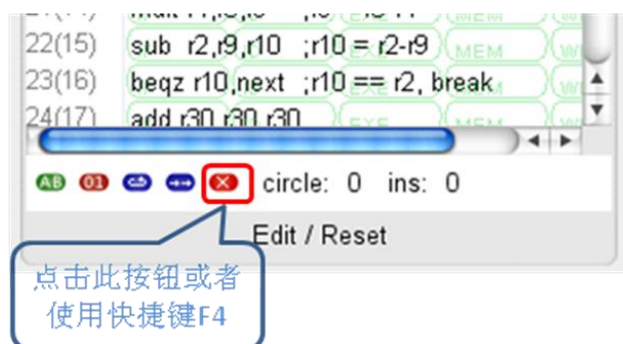


图 21

3.3.7 执行周期和指令数统计

执行程序的过程中，显示执行的周期数和指令数，如图 22 所示：



图 22

3.3.8 查看代码

3.3.8.1 查看指令

点击 Project 菜单中的 Code 按钮，可以查看指令，如图 23 所示：

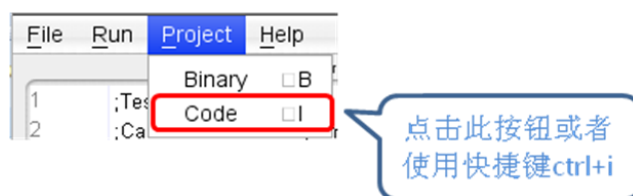


图 23

也可以点击主页面上的 AB 按钮，如图 24 所示：



图 24

3.3.8.2 查看二进制代码

点击 Project 菜单中的 Binary 按钮，可以查看二进制代码，如图 25 所示：



图 25

也可以点击主页面上的 01 按钮，如图 26 所示：

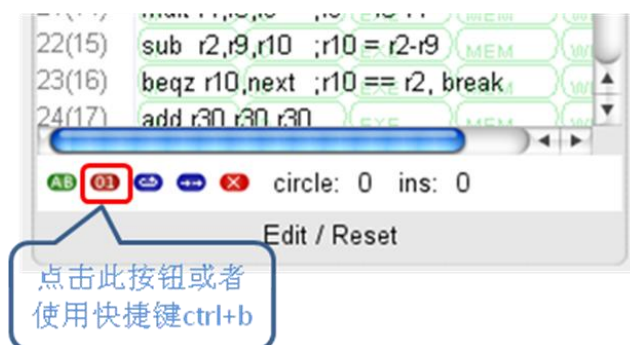


图 26

点击以后得到如图 27 结果：



图 27

3.4 结果查看

3.4.1 结果综述

在程序运行的过程中，Summary 标签下即时的显示出数据冲突、流水暂停、Cache 缺失等情况，如图 28 所示：

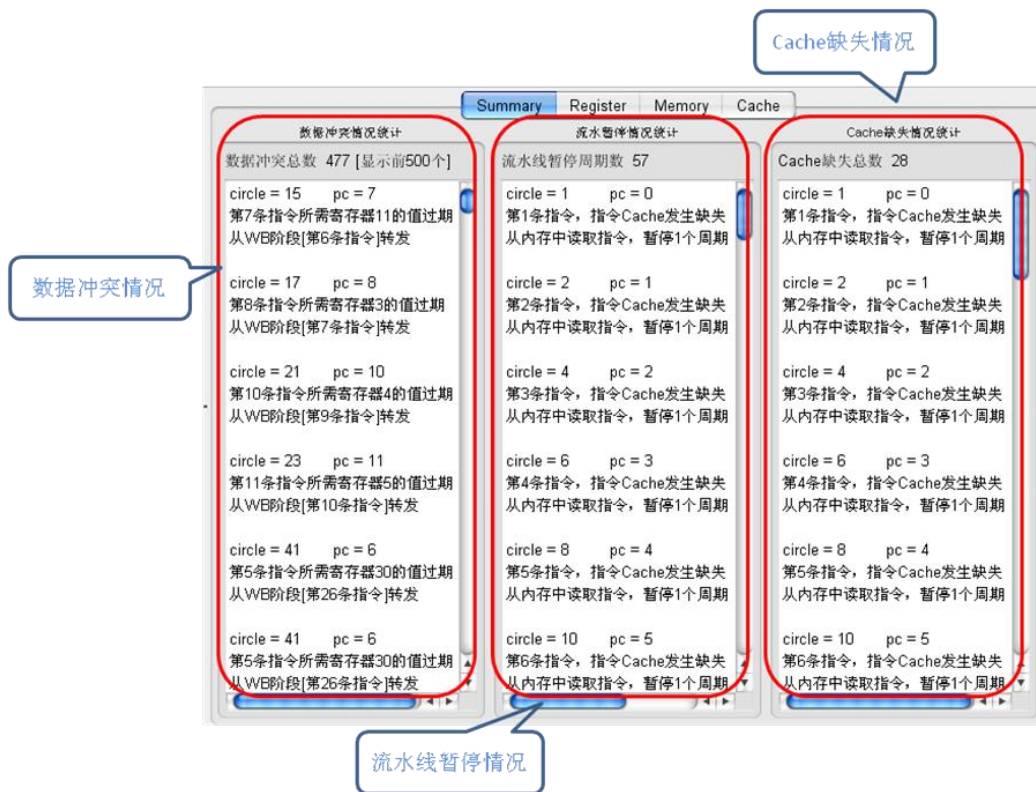


图 28

3.4.2 寄存器结果

3.4.2.1 显示 32 个寄存器当前的结果以及过去两次结果

在程序运行的过程中, Register 标签下即时的显示出各个寄存器值的变化, 如图 29 所示:



图 29

3.4.2.2 最近被修改的寄存器提示

在程序运行的过程中，会即时的显示出哪些寄存器值是刚刚被修改的，如图 30 所示：

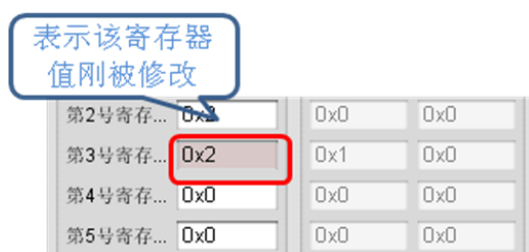


图 30

3.4.2.3 寄存器值修改

可以直接单击寄存器文本框，对寄存器的值进行修改，如果符合寄存器值的格式要求，则可以修改成功。

3.4.2.4 十进制和十六进制两种显示方式

Register 面板下方有“十进制显示”和“十六进制显示”两个按钮，点击按钮则寄存器的值分别显示为十进制和十六进制，如图 31 所示：

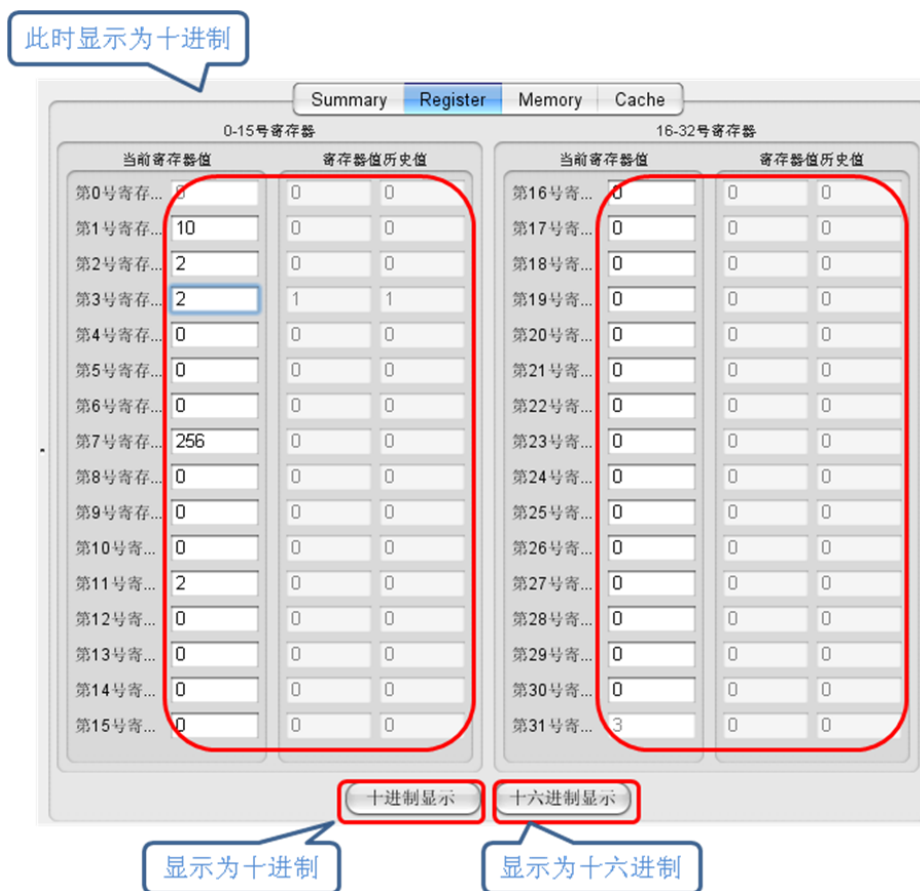


图 31

3.4.3 内存结果

3.4.3.1 查看指定内存地址的值

程序运行完成以后，可以在 Memory 标签下查看指定内存地址的值，如图 32 所示：



图 32

3.4.3.2 清空、保存以及恢复内存值

在 Memory 面板的下方有按钮“Clear Memory”和“Save Memory”，以及下拉菜单“Restore Memory to”，如图 33 所示：



图 33

3.4.3.4 显示最近读取和写入的内存地址历史值

在 Memory 面板的左右两侧分别有一部分区域显示最近读取和写入的内存地址，点击可以跳转到选中的地址，如图 34 所示：

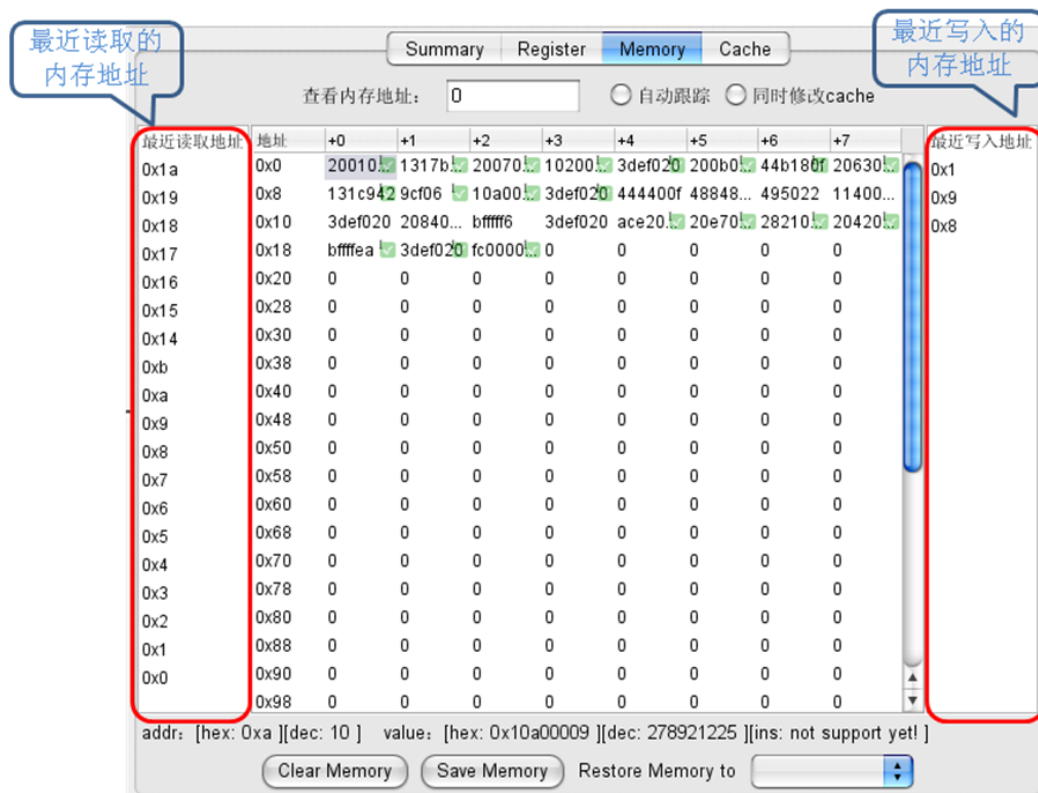


图 34

3.4.3.4 对内存进行自动跟踪

选择“自动跟踪”，则在执行程序时，内存地址将会即时的显示出运行结果，不需要选择查看内存地址。

3.4.3.5 同时修改 CACHE 值

选择“同时修改 Cache”，则在进行内存修改时，对应的 Cache 值也被修改。

3.4.3.6 内存与 CACHE 值同步提示

在内存面板上用图标显示出当前内存值与对应的 Cache 值是否同步，如图 35 所示：



图 35

3.4.3.7 内存值修改

可以直接双击某一个内存地址，对内存的值进行修改，如果符合内存值的格式要求，则可以修改成功。

3.4.4 CACHE 结果

3.4.4.1 分别显示指令 CACHE 和数据 CACHE 的值以及对应的内存地址

在 Cache 面板中分别显示了指令 Cache 和数据 Cache 的值以及对应的内存地址，如图 36 所示：

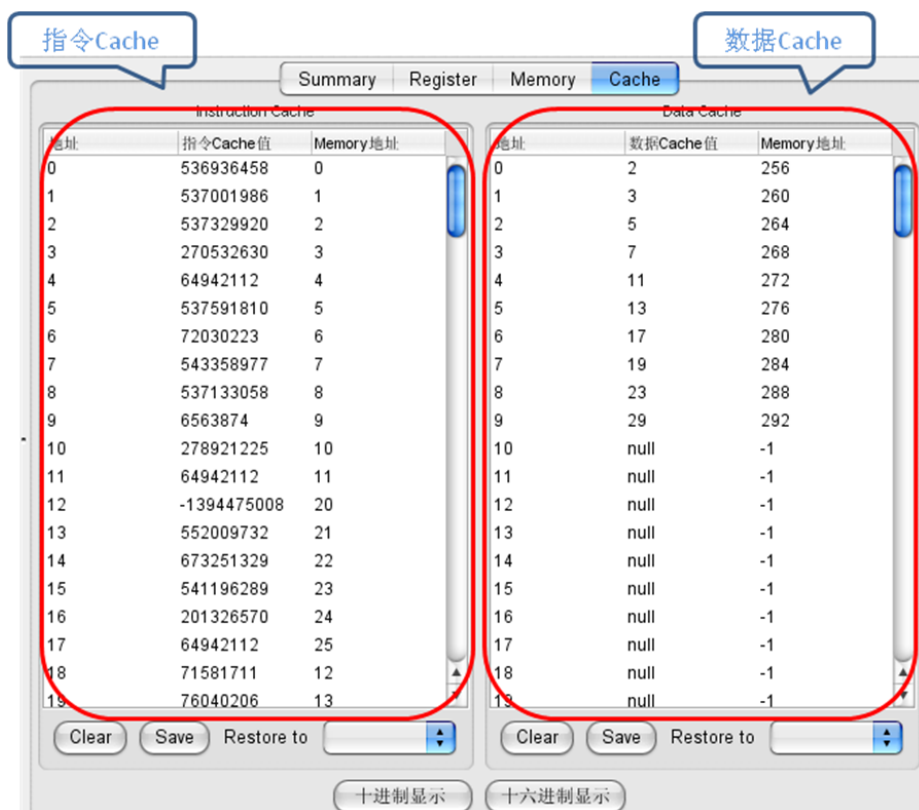


图 36

3.4.4.2 具有清空、保存以及恢复指令 CACHE 的功能

在指令 Cache 的下方有按钮“Clear”和“Save”，以及下拉菜单“Restore to”，如图 37 所示：



图 37

3.4.4.3 具有清空、保存以及恢复数据 CACHE 的功能

与上一条类似。

3.4.4.4 十进制和十六进制两种显示方式

Cache 面板下方有“十进制显示”和“十六进制显示”两个按钮，点击按钮则 Cache 地址和值分别显示为十进制和十六进制。

4、常见问题解答

4.1 问：可以导出二进制代码文件吗？

答：可以，在二进制代码模式下选择 Save 或者 Save As 可以保存二进制代码文件，但是请注意：**如果使用 Save 按钮则将覆盖源代码文件，且无提示，请小心使用，以免代码丢失。**

4.2 问：可以导入二进制代码文件吗？

答：不可以，暂不支持此功能。

4.3 问：为什么点任何按钮都无反应？

答：程序有可能遇到致命错误，请重新启动再试。

4.4 问：为什么修改寄存器或者内存的值但却未修改成功？

答：有多种可能，一、请检查代码是否已编译，点击编译按钮，系统会重置寄存器和内存，故在编译之前的改动将被忽略；二、寄存器或内存值可能会被用户程序中的赋值语句覆盖，导致没有修改成功；三、修改内存值时需要回车，请检查是否回车；四、模拟器出现错误，请联系我们，我们的联系方式见后面的问题。

4.5 问：为什么有跳转语句的程序执行结果会有问题？

答：跳转语句需要延时槽，请检查跳转语句后是否插入了 nop 语句。

4.6 问：为什么我的程序通不过编译？

答：有可能是指令格式错误，请按照我们定义的指令书写规则编写程序。指令书写规则见“指令系统 kde9.xls”中的文本指令格式一列。

4.7 问：请问如何联系你们？

答：我们 kde9 小组成员如下：



联系方式：孔祥欣，kxx006@gmail.com

胡玮玮，huww06@gmail.com

卿培，edwardtoday@gmail.com

To be updated.....