



“十二五”普通高等教育本科国家级规划教材

国家级精品课程教材



21世纪大学本科
计算机专业系列教材

张晨曦 李江峰
刘 依 沈 立 主编

计算机系统结构 实践教程 (第2版)

<http://www.tup.com.cn>

- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE *Computing Curricula 2005* 同步



清华大学出版社



普通高等教育“十一五”国家级规划教材

国家级精品课程教材

21世纪大学本科计算机专业系列教材

计算机系统结构实践教程 (第2版)

张晨曦 李江峰 刘依 沈立 主编



清华大学出版社

北京

内 容 简 介

本书设计了8个实验:指令系统和体系结构、流水线及流水线中的冲突、指令调度和延迟分支、Cache性能分析、再定序缓冲(ROB)工作原理、多Cache一致性——监听协议、多Cache一致性——目录协议。本书覆盖面广,内容丰富,有利于加深对系统结构知识的理解,书中的实验是基于MIPS指令集结构的,并提供了实验所需相关知识的介绍,可以与大多数系统结构教材配合使用。本书还提供了专门为系统结构实验开发的一套运行于Windows平台的模拟器。这些模拟器界面友好,使用方便、直观,交互性很强。

本书可作为本科生系统结构课程以及计算机组成与结构课程的实验教材,也可作为自学者的辅助教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机系统结构实践教程/张晨曦等主编.--2版.--北京:清华大学出版社,2015
21世纪大学本科计算机专业系列教材
ISBN 978-7-302-38940-8

I. ①计… II. ①张… III. ①计算机体系结构—高等学校—教材 IV. ①TP303

中国版本图书馆CIP数据核字(2015)第005674号

责任编辑:魏江江 薛 阳

封面设计:傅瑞学

责任校对:时翠兰

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:10.75 字 数:259千字

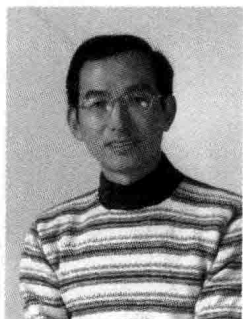
版 次:2010年5月第1版 2015年5月第2版 印 次:2015年5月第1次印刷

印 数:1~2000

定 价:29.00元

产品编号:061500-01

第一作者简介



张晨曦, 现任同济大学软件学院教授、博士生导师。国家级“中青年有突出贡献专家”, 国家杰出青年基金获得者, 上海市高校教学名师和上海市模范教师。先后主持了一个国家 973 计划课题和 5 项国家自然科学基金项目。

作为课程负责人, 张晨曦建设的“计算机系统结构”课程和“计算机组成原理”课程被评为上海市精品课程, “计算机系统结构”课程于 2008 年被评为国家级精品课程, 2013 年被评为国家级精品资源共享课。他主讲“计算机系统结构”课程和从事系统结构的研究 30 余年, 2008 年开发出了国内第一套 200 个用于本课程的动画课件, 2009 年开发出了国内第一套系统结构实验模拟器。

他负责编写出版的《计算机系统结构教程》教材于 2011 年被评为国家级精品教材, 全国至少有 100 所大学采用了该教材。他一共编写出版了 5 本“十一五”国家级规划教材, 撰写了专著两部, 其中专著《新一代计算机》由荷兰 North-Holland 出版社出版。

课程网站: www.Arch365.net

个人摄影网站: www.FotoSky.com

微信公众号: ZuiMeiJing(最美景)



前 言

计算机系统结构课程内容比较抽象、单调,不少内容要通过实验才能得到更好的理解。而且,通过实验研究对系统结构进行量化分析,是国际上流行的一种方法。

基于模拟器进行实验是一种很好的方式,有时其效果甚至比实物实验更好。因此,本教材基于编者开发的具有自主知识产权的模拟器,设计和编写了以下 8 个实验。

- (1) 指令系统和体系结构;
- (2) 流水线及流水线中的冲突;
- (3) 指令调度和延迟分支;
- (4) Cache 性能分析;
- (5) Tomasulo 算法;
- (6) 再定序缓冲(ROB)工作原理;
- (7) 多 Cache 一致性——监听协议;
- (8) 多 Cache 一致性——目录协议。

每个实验都由实验目的、实验平台、实验内容和步骤、模拟器使用手册以及相关知识 5 部分构成。“相关知识”部分系统地论述了与实验相关的知识,读者在实验前可以对其进行阅读和复习。

实验所需要的模拟器和样例程序可以从以下公众号获得。

微信公众号: Arch365

请扫描以下二维码,单击关注后,按提示操作。



本书主要由同济大学的张晨曦、李江峰、刘依和国防科技大学的沈立等编写。

本书可以跟大多数系统结构教材配合使用,也可作为自学者的辅助教材。

由于作者水平有限,书中难免有错误和不妥之处,敬请读者批评指正。

张晨曦

2014 年 12 月于上海

目 录

实验 1 MIPS 指令系统和 MIPS 体系结构	1
1.1 实验目的	1
1.2 实验平台	1
1.3 实验内容和步骤	1
1.4 MIPSsim 使用手册	4
1.4.1 启动模拟器	4
1.4.2 MIPSsim 的窗口	4
1.4.3 MIPSsim 的菜单	11
1.5 相关知识：MIPS 指令系统	15
1.5.1 MIPS 的寄存器	15
1.5.2 MIPS 的数据表示	16
1.5.3 MIPS 的数据寻址方式	16
1.5.4 MIPS 的指令格式	16
1.5.5 MIPS 的部分指令介绍	18
1.5.6 汇编程序举例	20
实验 2 流水线及流水线中的冲突	22
2.1 实验目的	22
2.2 实验平台	22
2.3 实验内容和步骤	22
2.4 MIPSsim 使用手册	24
2.5 相关知识：流水线、相关与冲突	24
2.5.1 一条经典的 5 段流水线	24
2.5.2 相关与流水线冲突	26
2.5.3 流水线的实现	31
实验 3 指令调度和延迟分支	35
3.1 实验目的	35
3.2 实验平台	35
3.3 实验内容和步骤	35

3.4	MIPSSim 使用手册	36
3.5	相关知识: 指令调度和延迟分支	37
3.5.1	指令调度	37
3.5.2	延迟分支	37
实验 4	Cache 性能分析	40
4.1	实验目的	40
4.2	实验平台	40
4.3	实验内容和步骤	40
4.3.1	Cache 容量对不命中率的影响	40
4.3.2	相联度对不命中率的影响	41
4.3.3	Cache 块大小对不命中率的影响	42
4.3.4	替换算法对不命中率的影响	42
4.4	MyCache 模拟器使用方法	43
4.5	相关知识: Cache 的基本原理	44
4.5.1	Cache 的映像规则	44
4.5.2	查找方法	46
4.5.3	替换算法	46
4.5.4	写策略	47
4.5.5	改进 Cache 性能	47
4.5.6	三种类型的不命中	48
4.5.7	降低不命中率的方法	48
4.5.8	分离 Cache 和混合 Cache	55
实验 5	Tomasulo 算法	56
5.1	实验目的	56
5.2	实验平台	56
5.3	实验内容和步骤	56
5.4	Tomasulo 算法模拟器的使用方法	57
5.5	相关知识: Tomasulo 算法	60
5.5.1	基本思想	60
5.5.2	具体算法	63
实验 6	再定序缓冲(ROB)工作原理	67
6.1	实验目的	67
6.2	实验平台	67
6.3	实验内容和步骤	67
6.4	ROB 模拟器的使用方法	68

6.5 相关知识：再定序缓冲 ROB	71
实验 7 多 Cache 一致性——监听协议	74
7.1 实验目的	74
7.2 实验平台	74
7.3 实验内容和步骤	74
7.4 监听协议模拟器使用方法	75
7.5 相关知识：监听协议	76
7.5.1 基本思想	76
7.5.2 监听协议的实现	77
实验 8 多 Cache 一致性——目录协议	82
8.1 实验目的	82
8.2 实验平台	82
8.3 实验内容和步骤	82
8.4 目录协议模拟器使用方法	83
8.5 相关知识：目录协议	84
8.5.1 目录协议的基本思想	84
8.5.2 目录协议实例	87
附录 A MIPSsim 的指令列表	91
附录 B MIPSsim 的指令系统	100
附录 C 模拟器 MIPSsim 的汇编语言	158
参考文献	161

实验 1 MIPS 指令系统和 MIPS 体系结构

1.1 实验目的

- (1) 了解和熟悉指令级模拟器。
- (2) 熟练掌握 MIPSsim 模拟器的操作和使用方法。
- (3) 熟悉 MIPS 指令系统及其特点,加深对 MIPS 指令操作语义的理解。
- (4) 熟悉 MIPS 体系结构。

1.2 实验平台

实验平台采用指令级和流水线操作级模拟器 MIPSsim。

设计: 张晨曦教授, 版权所有。

下载地址:

微信公众号: Arch365

请扫描以下二维码, 单击关注后, 按提示操作。



1.3 实验内容和步骤

首先要阅读 MIPSsim 模拟器的使用方法(见 1.4 节), 然后了解 MIPSsim 的指令系统和汇编语言(见附录 A、附录 B 和附录 C)。

- (1) 启动 MIPSsim(双击 MIPSsim.exe)。
- (2) 选择“配置”→“流水方式”, 使模拟器工作在非流水方式下。

(3) 参照 1.4 节的使用说明,熟悉 MIPSsim 模拟器的操作和使用方法。

可以先载入一个样例程序(在本模拟器所在的文件夹下的“样例程序”文件夹中),然后分别以单步执行一条指令、执行多条指令、连续执行、设置断点等的方式运行程序,观察程序的执行情况,观察 CPU 中寄存器和存储器的内容的变化。

(4) 选择“文件”→“载入程序”选项,加载样例程序 alltest.asm,然后查看“代码”窗口,查看程序所在的位置(起始地址为 0x00000100)。

(5) 查看“寄存器”窗口 PC 寄存器的值: [PC]=0x_____。

(6) 执行 load 和 store 指令,步骤如下。

① 单步执行一条指令(F7)。

② 下一条指令地址为 0x_____, 是一条 _____ (有,无)符号载入 _____ (字节,半字,字)指令。

③ 单步执行 1 条指令(F7)。

④ 查看 R1 的值, [R1]=0x_____。

⑤ 下一条指令地址为 0x_____, 是一条 _____ (有,无)符号载入 _____ (字,半字,字)指令。

⑥ 单步执行 1 条指令。

⑦ 查看 R1 的值, [R1]=0x_____。

⑧ 下一条指令地址为 0x_____, 是一条 _____ (有,无)符号载入 _____ (字,半字,字)指令。

⑨ 单步执行 1 条指令。

⑩ 查看 R1 的值, [R1]=0x_____。

⑪ 单步执行 1 条指令。

⑫ 下一条指令地址为 0x_____, 是一条保存 _____ (字,半字,字)指令。

⑬ 单步执行 1 条指令。

⑭ 查看内存 BUFFER 处字的值,值为 0x_____。

(7) 执行算术运算类指令,步骤如下。

① 双击“寄存器”窗口里的 R1,将其值修改为 2。

② 双击“寄存器”窗口里的 R2,将其值修改为 3。

③ 单步执行 1 条指令。

④ 下一条指令地址为 0x_____, 是一条加法指令。

⑤ 单步执行 1 条指令。

⑥ 查看 R3 的值, [R3]=0x_____。

⑦ 下一条指令地址为 0x_____, 是一条乘法指令。

⑧ 单步执行 1 条指令。

⑨ 查看 LO、HI 的值, [LO]=0x_____, [HI]=0x_____。

(8) 执行逻辑运算类指令,步骤如下。

① 双击“寄存器”窗口里的 R1,将其值修改为 0xFFFF0000。

② 双击“寄存器”窗口里的 R2,将其值修改为 0xFF00FF00。

③ 单步执行 1 条指令。

④ 下一条指令地址为 0x_____, 是一条逻辑与运算指令,第二个操作数寻址方式是_____ (寄存器直接寻址,立即数寻址)。

⑤ 单步执行 1 条指令。

⑥ 查看 R3 的值, [R3]=0x_____。

⑦ 下一条指令地址为: 0x_____, 是一条逻辑或指令,第二个操作数寻址方式是_____ (寄存器直接寻址,立即数寻址)。

⑧ 单步执行 1 条指令。

⑨ 查看 R3 的值, [R3]=0x_____。

(9) 执行控制转移类指令,步骤如下。

① 双击“寄存器”窗口里的 R1,将其值修改为 2。

② 双击“寄存器”窗口里的 R2,将其值修改为 2。

③ 单步执行 1 条指令。

④ 下一条指令地址为 0x_____, 是一条 BEQ 指令,其测试条件是_____, 目标地址为 0x_____。

⑤ 单步执行 1 条指令。

⑥ 查看 PC 的值, [PC]=0x_____, 表明分支_____ (成功,失败)。

⑦ 下一条指令地址是一条 BGEZ 指令,其测试条件是_____, 目标地址为 0x_____。

⑧ 单步执行 1 条指令。

⑨ 查看 PC 的值, [PC]=0x_____, 表明分支_____ (成功,失败)。

⑩ 下一条指令地址是一条 BGEZAL 指令,其测试条件是_____, 目标地址为 0x_____。

⑪ 单步执行 1 条指令。

⑫ 查看 PC 的值, [PC]=0x_____, 表明分支_____ (成功,失败); 查看 R31 的值, [R31]=0x_____。

⑬ 单步执行 1 条指令。

⑭ 查看 R1 的值, [R1]=0x_____。

⑮ 下一条指令地址为 0x_____, 是一条 JALR 指令,保存目标地址的寄存器为 R_____, 保存返回地址的目标寄存器为 R_____。

⑯ 单步执行 1 条指令。

⑰ 查看 PC 和 R3 的值, [PC]=0x_____, [R3]=0x_____。

1.4 MIPSsim 使用手册

1.4.1 启动模拟器

双击 MIPSsim.exe,即可启动该模拟器。MIPSsim 是在 Windows 操作系统上运行的程序,它需要用 .NET 运行环境。

模拟器启动时,自动将自己初始化为默认状态。所设置的默认值为:

- ◆ 所有通用寄存器和浮点寄存器为全 0;
- ◆ 内存清零;
- ◆ 流水寄存器为全 0;
- ◆ 清空时钟图、断点、统计数据;
- ◆ 内存大小为 4096B;
- ◆ 载入起始地址为 0;
- ◆ 浮点加法、乘法、除法部件的个数均为 1;
- ◆ 浮点加法、乘法、除法运算延迟分别为 6、7、10 个时钟周期;
- ◆ 采用流水方式;
- ◆ 不采用定向机制;
- ◆ 不采用延迟槽;
- ◆ 采用符号地址;
- ◆ 采用绝对周期计数。

当模拟器工作在非流水方式下(配置菜单中的“流水方式”前没有√号)时,下面叙述中有关流水段的内容都没有意义,应该忽略。

1.4.2 MIPSsim 的窗口

在流水方式下,模拟器主界面中共有 7 个子窗口,它们是代码窗口、寄存器窗口、流水线窗口、时钟周期图窗口、内存窗口、统计窗口和断点窗口。每一个窗口都可以被收起(变成小图标)、展开、拖动位置和放大/缩小。当要看窗口的全部内容时,可以将其放大到最大。

在非流水方式下,只有代码窗口、寄存器窗口、内存窗口和断点窗口。

1. 代码窗口

代码窗口给出内存中代码的列表,每条指令占一行,按地址顺序排列。每行有 5 列(当全部显示时):地址、断点标记、指令的机器码、流水段标记和符号指令,如图 1.1 所示。

图 1.1 中不同抹色的行代表相应的指令所处的执行段。黄色代表 IF 段,绿色代表

地址	断点标记	机器码	流水段	符号指令
0x00000000		0x8c01003c		LU \$1,60(\$0)
0x00000004		0x04318004	WB	BEZAL \$1,4
0x00000008		0x00000000	MEM	SLL \$0,\$0,0
0x0000000c	B. MEM	0xAC02003c		SW \$2,60(\$0)
0x00000010	B. IF	0x00000034		TEQ \$0,\$0
0x00000014		0x00000000		SLL \$0,\$0,0
0x00000018		0x00201020	EX	ADD \$2,\$1,\$0
0x0000001c		0x2021ffff	ID	ADDI \$1,\$1,-1
0x00000020		0x10200004	IF	BEQ \$1,\$0,4
0x00000024		0x00000000		SLL \$0,\$0,0
0x00000028		0x70411002		MUL \$2,\$2,\$1
0x0000002c		0x1000ffff		BEQ \$0,\$0,-5
0x00000030		0x00000000		SLL \$0,\$0,0
0x00000034	B. ID	0x03E00008		JR \$31
0x00000038		0x00000000		SLL \$0,\$0,0
0x0000003c		0x00000005		
0x00000040		0x00000000		SLL \$0,\$0,0
0x00000044		0x00000000		SLL \$0,\$0,0
0x00000048		0x00000000		SLL \$0,\$0,0
0x0000004c		0x00000000		SLL \$0,\$0,0
0x00000050		0x00000000		SLL \$0,\$0,0
0x00000054		0x00000000		SLL \$0,\$0,0
0x00000058		0x00000000		SLL \$0,\$0,0
0x0000005c		0x00000000		SLL \$0,\$0,0

图 1.1 代码窗口

ID 段,红色代表 EX 段,青色代表 MEM 段,棕色代表 WB 段。

该窗口中各列的含义如下。

- ◆ 地址：以十六进制的形式给出。内存是按字节寻址的,每条指令占 4 字节。当采用符号地址时,会在相应的位置给出汇编程序中出现的标号。
- ◆ 断点标记：如果在该指令处设有断点,则显示相应的标记。断点标记的形式为 B. X(X 为段名),表示该断点是设置在该指令的 X 段。例如,若某行的断点标记为 B. EX,则表示在该指令的 EX 段设置了断点。

当模拟器工作在非流水方式下时,断点的标记为 B。

- ◆ 机器码：该行所对应的指令的十六进制机器码。若该行无指令,则仅仅显示 4 字节数据。
- ◆ 流水段标记：表示当该指令正在执行时,它在当前周期该指令所处的流水段。当模拟器工作在非流水方式下时,它没有意义。
- ◆ 符号指令：机器代码所对应的符号指令。

在该窗口中选中某行(单击鼠标左键),然后再单击鼠标右键,就会弹出菜单“设置断点”和“清除断点”,它们分别用于在所选指令处设置断点和清除断点。

- 设置断点：

选择(单击)要设断点的指令后单击鼠标右键→“设置断点”,弹出“设置断点”对话框,在“段”的下拉框中选择断点所在的流水段(在非流水方式下,不存在该下拉框),单击“确定”按钮即可。

- 清除断点：

选择(单击)指令后单击鼠标右键→“清除断点”,则设置在该指令处的断点被删除。

2. 寄存器窗口

寄存器窗口显示 MIPSsim 模拟器中寄存器的内容。共有 4 组寄存器：通用寄存器、

浮点寄存器、特殊寄存器和流水寄存器。分为4栏来显示,每一栏下分别有各自的数据格式选项,如图1.2所示。

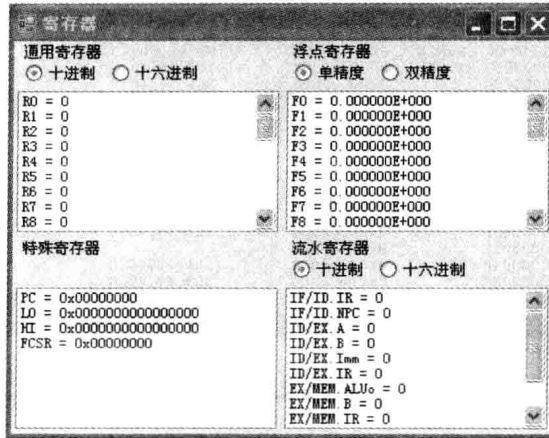


图 1.2 寄存器窗口

(1) 通用寄存器

MIPS64 有 32 个 64 位通用寄存器: R_0, R_1, \dots, R_{31} 。它们简称为 GPRs (General-Purpose Registers), 有时也称为整数寄存器。R0 的值永远是 0。

通过数据格式选项, 可以选择显示的格式是十进制还是十六进制。

(2) 浮点寄存器

共有 32 个 64 位浮点数寄存器: F_0, F_1, \dots, F_{31} 。它们简称为 FPRs (Floating-Point Registers)。它们既可以用来存放 32 个单精度浮点数(32 位), 也可以用来存放 32 个双精度浮点数(64 位)。存储单精度浮点数(32 位)时, 只用到 FPR 的一半, 其另一半没用。

(3) 特殊寄存器

特殊寄存器有以下 4 个。

- ◆ PC: 程序计数器(32 位)。
- ◆ LO: 乘法寄存器的低位。
- ◆ HI: 乘法寄存器的高位。
- ◆ FCSR: 浮点状态寄存器。

(4) 流水寄存器

- ◆ IF/ID. IR: 流水段 IF 与 ID 之间的指令寄存器。
- ◆ IF/ID. NPC: 流水段 IF 与 ID 之间的下一指令程序计数器。
- ◆ ID/EX. A: 流水段 ID 与 EX 之间的第一操作数寄存器。
- ◆ ID/EX. B: 流水段 ID 与 EX 之间的第二操作数寄存器。
- ◆ ID/EX. Imm: 流水段 ID 与 EX 之间的立即数寄存器。
- ◆ ID/EX. IR: 存放从 IF/ID. IR 传过来的指令。
- ◆ EX/MEM. ALUo: 流水段 EX 与 MEM 之间的 ALU 计算结果寄存器。
- ◆ EX/MEM. IR: 存放从 ID/EX. IR 传过来的指令。

- ◆ MEM/WB. LMD: 流水段 MEM 与 WB 之间的数据寄存器,用于存放从存储器读出的数据。
- ◆ MEM/WB. ALU_o: 存放从 EX/MEM. ALU_o 传过来的计算结果。
- ◆ MEM/WB. IR: 存放从 EX/MEM. IR 传过来的指令。

除了流水寄存器外,其他寄存器都可以修改。只要双击某寄存器所在的行,系统就会弹出一个对话框。该对话框显示了该寄存器原来的值,在新值框中填入新的值,然后单击“保存”按钮,系统就会将新值写入该寄存器。

3. 流水线窗口

流水线窗口显示流水线在当前配置下的组成以及该流水线的各段在当前周期正在处理的指令,如图 1.3 所示。

非流水方式下,没有该窗口。

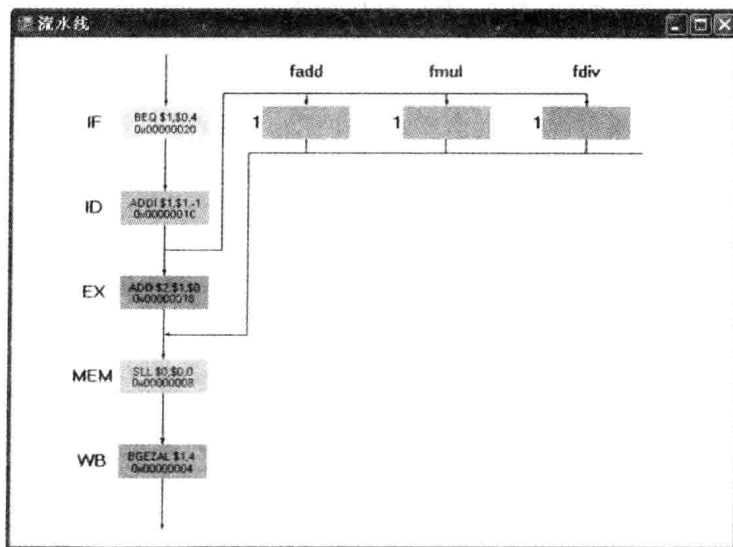


图 1.3 流水线窗口

在该窗口中,每一个矩形方块代表一个流水段,它们用不同的颜色填充。在该窗口的左侧是 IF 到 WB 段,其右边为浮点部件。浮点部件分有浮点加法部件(fadd)、浮点乘法部件(fmul)和浮点除法部件(fddiv)三种。在菜单“配置”→“常规配置”中修改浮点部件个数,可看到该窗口中对应类型的浮点部件个数会发生相应的变化。

在运行过程中,各段的矩形方块中会显示该段正在处理的指令及其地址(十六进制)。当双击某矩形方块时,会弹出窗口显示该段出口处的流水寄存器的内容(十六进制)。

4. 时钟周期图窗口

该窗口用于显示程序执行的时间关系,画出各条指令执行时所用的时钟周期。非流水方式下,没有该窗口。

以窗口左上为原点,横轴正方向指向右方,表示模拟器先后经过的各个周期(列),纵

轴正方向指向下方,表示模拟器中先后执行的各条指令(行),如图 1.4 所示。

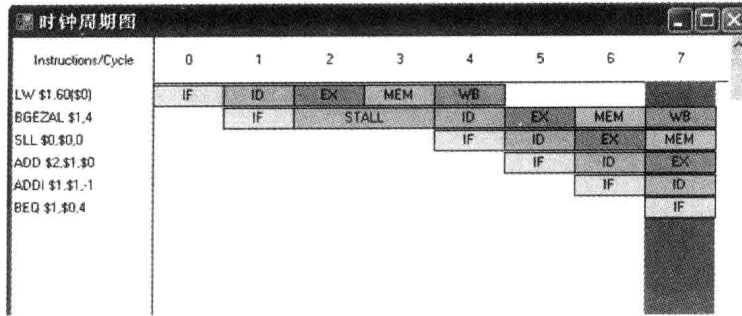


图 1.4 时钟周期图窗口

横坐标有相对周期计数和绝对周期计数两种不同的表示形式。在默认的绝对周期计数下,按 0、1、2……依次递增的顺序计数。而在相对周期计数下,当前周期记为第 0 个周期,而其余周期(在左边)则按其相对于当前周期的位置,分别记为-1、-2、-3……等。

在由指令轴和周期轴组成的二维空间下,坐标 (n, i) 对应的矩形区域表示指令 i 在第 $n+1$ 周期时所经过的流水段(假设采用绝对周期计数)。

双击某行时,会弹出一个窗口,显示该指令在各流水段所进行的处理。

该窗口中还显示定向的情况。这是用箭头来表示的。若在第 m 周期和第 $m+1$ 周期间产生从指令 i_1 到指令 i_2 的定向,则在坐标 (m, i_1) 和 $(m+1, i_2)$ 表示的矩形区域之间会有一个箭头。

5. 内存窗口

该窗口显示模拟器内存中的内容,左侧一栏为十六进制地址,右侧为数据,如图 1.5 所示。可以直接通过双击来修改其内容,这时会弹出一个“内存修改”对话框,如图 1.6 所示。对话框的上部区域为数据类型与格式选择区,通过勾选其中的一项,就可以指定所采用的数据类型与格式。

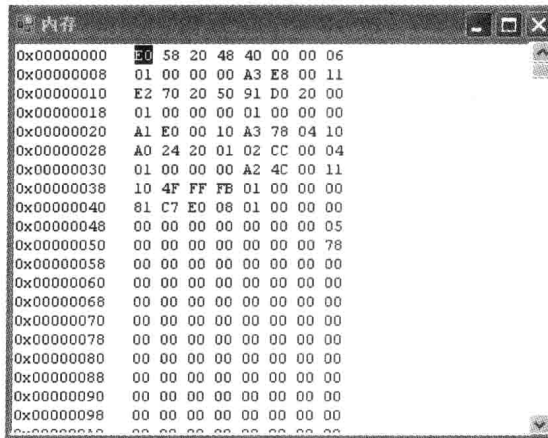


图 1.5 内存窗口

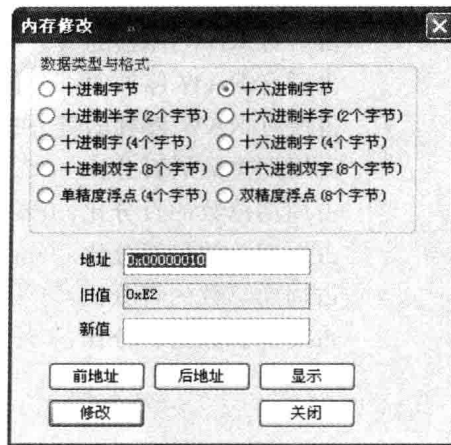


图 1.6 “内存修改”对话框

在该“内存修改”对话框中，地址框最开始显示的是被双击单元的地址，用户可以直接修改该地址。在新值框中输入新值，然后单击“修改”按钮，模拟器就会把新值写入内存中相应的单元。新值的格式必须与所选的数据类型和格式一致。

“前地址”与“后地址”按钮分别将当前地址减少和增加一个数据长度（字节数），并显示当前地址所指定单元的内容。“前地址”和“后地址”用于连续修改一片的内存数据。“显示”按钮用于显示当前地址所指单元的内容。在修改地址后，单击该按钮就可以显示内存单元的内容。

6. 统计窗口

该窗口显示模拟器统计的各项数据，如下所示。

（非流水方式下，没有该窗口）

汇总：

执行周期总数：0

ID 段执行了 0 条指令

硬件配置：

内存容量：4096 B

加法器个数：1

执行时间(周期数)：6

乘法器个数：1

执行时间(周期数)：7

除法器个数：1

执行时间(周期数)：10

定向机制：不采用

停顿(周期数)：

RAW 停顿：0

占周期总数的百分比：0%