

ICS: Y86 PIPELINE SIMULATOR

实验报告

夏亦婧 13307130414

一、实验要求：

通过文件输入Y86指令，模拟Y86处理器，显示每个周期各个流水线寄存器的值、寄存器的值、指定地址和长度存储器的值及其他在流水线中的变量值。

二、具体实现过程：

1. 实现平台和语言：

操作系统：OSX Yosemite 10.10

集成开发环境：Eclipse Luna

开发语言：Java

JDK版本：Java 1.7.0_80

2. 实现功能

(1)内核功能：

- 读入输入文件(*.yo)的Y86指令内容，对指令地址、指令、指令内容进行储存。
- 产生输出文件(*.txt)，模拟Y86处理器每个时钟周期的运行，保存每个周期各个流水线寄存器的值。
- 实现Y86指令集中的所有指令包括：
nop, halt
rrmovl, irmovl, rmmovl, mrmovl
addl, subl, andl, xorl
jmp, jle, jl, je, jne, jge, jg
call, ret
pushl, popl
- 实现流水线控制逻辑(forwarding, stall, bubble)，避免数据冒险
- 检测对非法地址的读写(ADR)

(2)界面功能：

a. 显示内容：

左1: 显示程序运行的状态，包括运行频率和当前周期。

左2: 显示程序运行时栈中的内容/地址，和其对应的内容（如没有则显示0x00000000），栈指针所指向的位置用灰色底色表示。

中1: 显示各个流水线寄存器的各个变量的值，显示在流水线各个阶段所正在处理的指令内容，当出现bubble和stall时，显示bubble和stall标志。

中2左: 显示CC的值。

中2右: 显示各个寄存器的值。

右1: 显示输入文件中的指令地址和指令内容，当前在流水线中处理的指令用蓝色表示。

右2: 所查询地址存储器中的内容。

b. 按键功能：

load 载入文件 *.yo

save 导出当前输入文件所对应的每个时钟周期的流水线寄存器的值

frequency滑动条 调节运行频率

run/stop 开始/暂停运行

back/next 单步后退/前进

set 跳转至jump to cycle文本框中所输入的周期

reset 重置

check memory 查看起始地址(from address)起所输入长度(length)位的存储器内容

程序运行截图：

Y86 PIPELINE SIMULATOR ©YIJING XIA 2015

Y86 PIPELINE SIMULATOR

LOAD SAVE

FREQUENCY: 9 hz

RUN STOP

CURRENT CYCLE: 51

BACK NEXT

JUMP TO CYCLE:

SET RESET

[STACK]

ADDR	CONTENT
0x100	0x00000000
0xfc	0x00000004
0xf8	0x00000014
0xf4	0x00000039
0xf0	0x00000100

WRITE BACK: addl %ebx,%edx

stat	icode	valE	valM	dstE	dstM
AOK	0x6	0x00000000	0x0000a000	0x2	0x8

MEMORY: jne Loop

stat	icode	Cnd	valE	valA	dstE	dstM
AOK	0x7	false	0x00000000	0x00000074	0x8	0x8

EXECUTE: [bubble] nop

stat	icode	ifun	valC	valA	valB
AOK	0x0	0x0	0x00000000	0x00000000	0x00000000

dstE	dstM	srcA	srcB
0x8	0x8	0x8	0x8

DECODE: [bubble] nop

stat	icode	ifun	rA	rB	valC	valP
AOK	0x0	0x0	0x8	0x8	0x00000000	0x00000000

FETCH: popl %ebp

predPC
0x5f

[CC]

ZF: 1

SF: 0

OF: 0

%eax:	0x0000abcd	%esp:	0x000000f0
%ecx:	0x00000024	%ebp:	0x000000f0
%edx:	0x00000001	%esi:	0x0000a000
%ebx:	0xffffffff	%edi:	0x00000000

[CHECK MEMORY]

FROM ADDRESS: 0x 14

LENGTH: 4 BYTES

RESULT: 0x 0000000d

CHECK

ADDR INSTRUCTIONS

0x0	init: irmovl Stack, %esp
0x6	irmovl Stack, %ebp
0xc	jmp Main
0x24	Main: irmovl \$4,%eax
0x2a	pushl %eax
0x2c	irmovl array,%edx
0x32	pushl %edx
0x34	call Sum
0x39	halt
0x3a	Sum: pushl %ebp
0x3c	rrmovl %esp,%ebp
0x3e	rrmovl 8(%ebp),%ecx
0x44	rrmovl 12(%ebp),%edx
0x4a	irmovl \$0,%eax
0x50	andl %edx,%edx
0x52	je End
0x57	Loop: rrmovl (%ecx),%esi
0x5d	addl %esi,%eax
0x5f	irmovl \$4,%ebx
0x65	addl %ebx,%ecx
0x67	irmovl \$-1,%ebx
0x6d	addl %ebx,%edx
0x6f	jne Loop
0x74	popl %ebp
0x76	ret

三、测试：

1. 对Y86所有基础指令进行测试，查看输出文件的流水线寄存器的值。
2. 处理UI界面上按键的冲突，当程序在自动运行(RUN)过程中，单步前进/后退功能自动失效，其他按键功能将首先使程序停止运行之后再实现。
3. 程序未对非法指令、非法地址、无halt文件等进行处理，如遇到此类情况，程序将报错并自动关闭。