



北京航空航天大学

BEIHANG UNIVERSITY

计算机组成原理 Project7 实验报告

MIPS 微体系 – 支持异常和中断

支持指令集

{add,addu,and,div,divu,mult,multu,nor,or,sub,subu,xor,

addi,addiu,andi,lui,ori,xori,

beq,blez,bltz,bgez,bgtz,bne,

slt,sltu,slti,sltiu,sll,sllv,sra,srav,srl,srlv,

lw,lh,lhu,lb,lbu,sw,sh,sb,jal,jr,j,jalr,mthi,mtlo,mfhi,mflo,mfc0,mtc0,eret}.

北京航空航天大学

计算机学院

陈麒先

16061160

二〇一八年一月

郑重声明

关于诚实守信公约：

本实验报告由本人独立完成，全部内容均为本人通过查找互联网资料、翻阅课件、课堂笔记和教材后独立思考的结果。特此声明。

16061160

陈麒先

原创性声明

作业中出现的公式、图片、代码段以及图片的文字注释信息，均为作者原创。抄袭行为在任何情况下都被严格禁止 (COPY is strictly prohibited under any circumstances)！转载或引用须征得作者本人同意，并注明出处！

16061160

陈麒先

第一章 测试验证

new instr test

```
ori $1 $0 0x1234
mtc0 $1 $14
mfc0 $2 $14
ori $1 $0 0
mtc0 $1 $14
mfc0 $2 $14
ori $1 $0 0x7c01 #0111_1100_0000_0001
mtc0 $1 $12
mfc0 $2 $12
ori $1 $0 0
lui $3 0x8000
ori $1 $0 0x007c #0000_0000_0111_1100
addu $1 $1 $3
mtc0 $1 $13
mfc0 $2 $13
mtc0 $0 $13
ori $1 $0 0
ori $1 $0 0x3054
mtc0 $1 $14
eret
lui $4 0x1234
ori $3 $0 0x5678
```

CP0 forward

```
ori $3 $0 0x007c
ori $1 $0 0x8666
mtc0 $1 $14
```

```

ori $2 $0 0x7c01
mtc0 $2 $12
sw $3 0($0)
lw $4 0($0)
mtc0 $4 $13
mfc0 $5 $13
sw $5 4($0)
mfc0 $6 $14
subu $6 $0 $6
mfc0 $7 $12
blez $7 if_1_else
nop
if_1:
j end
nop
if_1_else:
sll $8 $1 7
end:
sll $8 $1 3
# PC error
lui $3 0x8000
ori $1 $0 0x007c
addu $1 $1 $3
ori $4 $0 4
jr $4
nop
# illegal instr
lui $3 0x8000
ori $1 $0 0x007c
addu $1 $1 $3

```

```

ori $4 $0 0x0008
movz $1 $6 $0 #illegal
add $7 $6 $3
nop
# load_EXC
#beyond the boundary
ori $1 $0 0x3000
ori $2 $0 3
lw $4 0($1)
addu $5 $4 $2
ori $1 $0 0x7F0c
ori $2 $0 4
lw $4 0($1)
addu $5 $4 $2
ori $1 $0 0x7F00
ori $2 $0 5
lw $4 0($1)
addu $5 $4 $2
# unaligned
ori $1 $0 0x7F00
ori $2 $0 6
lw $4 1($1)
addu $5 $4 $2

ori $1 $0 0x1234
lui $2 0x8765
add $2 $2 $1
sw $2 0($0)
lh $10 0($0)
lh $11 1($0)

```

lh \$12 2(\$0)

lh \$13 3(\$0)

lw \$14 0(\$0)

lw \$15 2(\$0)

lb \$16 3(\$0)

store_EXC

#beyond the boundary

ori \$1 \$0 0x3004

ori \$2 \$0 0x1234

lui \$3 0x9876

add \$4 \$3 \$2

sw \$4 0(\$1)

ori \$1 \$0 0x0004

sw \$4 0(\$1)

ori \$1 \$0 0x7F10

ori \$6 \$0 5

sw \$6 0(\$1)

sw \$6 4(\$1)

sw \$6 8(\$1)

ori \$1 \$0 0x7F00

sw \$6 8(\$1)

sw \$6 0(\$1)

#unaligned

ori \$1 \$0 0x1234

lui \$2 0x8765

add \$2 \$2 \$1

sb \$2 1(\$0)

sw \$2 1(\$0)

sb \$2 6(\$0)

sw \$2 6(\$0)

sb \$2 11(\$0)

sw \$2 11(\$0)

sb \$2 13(\$0)

sh \$2 13(\$0)

sb \$2 18(\$0)

sh \$2 18(\$0)

sb \$2 23(\$0)

sw \$2 23(\$0)

sb \$2 28(\$0)

sh \$2 28(\$0)

第二章 思考题

1. 我们计组课程一本参考书目标题中有“硬件/软件接口”接口字样，那么到底什么是“硬件/软件接口”？

答：I/O 系统接口：它是 I/O 系统与上层系统之间的接口，向上层提供对设备进行操作的抽象 I/O 命令，以便于高层对设备的使用

软件/硬件接口：在接口之上是中断处理程序和用于不同设备的设备驱动程序，在此之下是各种设备的控制器，如 CD-ROM 控制器、硬盘控制器、键盘控制器、打印机控制器、网络控制器等，它们都属于硬件。由于设备种类繁多，故该接口相当复杂。

2. 在我们设计的流水线中，DM 处于 CPU 内部，请你考虑现代计算机中它的位置应该在何处。

答：在 CPU 外部的主板上。

3. BE 部件对所有的外设都是必要的吗？

答：不是必要的。在我们 P7 所设计的外设 COCO 中，就只允许对 COCO 按字访问，即只能使用 sw 指令。故对 COCO，无需使用 BE 部件。

4. 请开发一个主程序以及定时器的 exception handler。整个系统完成如下功能：

- ① 定时器在主程序中被初始化为模式 0;
- ② 定时器倒计时至 0 产生中断;
- ③ handler 设置使能 Enable 为 1 从而再次启动定时器的计数器。2 及 3 被无限重复。
- ④ 主程序在初始化时将定时器初始化为模式 0, 设定初值寄存器的初值为某个值, 如 100 或 1000。(注意, 主程序可能需要涉及对 CP0. SR 的编程, 推荐阅读过后文后再进行。)

答:

```
.text
ori $1,$0,0x7f00
ori $2,$0,0x7f04
ori $5,$0,0x0009
ori $6,$0,0x0005
ori $7,$0,0xff1
mtc0 $7,$12
sw $6,0($2)
sw $5,0($1)
ori $3,$0,0x7f10
ori $4,$0,0x7f14

.ktext 0x00004180
ori $1,$0,0x7f00
ori $2,$0,0x0001
sw $2,0($1)
nop
nop
eret
```


5. 请查阅相关资料，说明鼠标和键盘的输入信号是如何被 CPU 知晓的？

答：IO 设备的输入输出有好几种方式，键盘、鼠标这类的低速设备是通过中断请求的方式进行 IO 操作的。即当键盘上按下一个按键的时候，键盘会发出一个中断信号，中断信号经过中断控制器传到 CPU，然后 CPU 根据不同的中断号执行不同的中断响应程序，然后进行相应的 IO 操作，把按下的按键编码读到寄存器（或者鼠标的操作），最后放入内存中。