实验3报告

第24小组

张传奇、谈清扬、孔静

1. 实验目的

 理解计算机系统启动时的初始化过程

 理解 TLB、CACHE 的结构，掌握其初始化的原理和方法

 了解串口的工作原理，掌握波特率的设置方法

 认识 PMON 并掌握简单的命令

 进一步提高 MIPS 汇编编程的能力

 进一步提高软件的调试经验

1. 实验任务

（一）设计

1、串口初始化模块

（1）工作内容

编写汇编程序，使之能完成比特率设置。

（2）工作目标

完成串口初始化。

2、Cache初始化模块

（1）工作内容

编写汇编程序，使之能完成Cache的初始化。

（2）工作目标

完成Cache初始化。

3、TLB初始化模块

（1）工作内容

编写汇编程序，使之能完成TLB的初始化。

（2）工作目标

完成TLB初始化。

（二）实现

1、串口初始化模块

计算所需比特率，设置好线路控制器，并设置好相应的分频锁存器。

2、Cache初始化模块

固定规模的版本比较好写，通过循环完成Cache的初始化。

3、TLB初始化模块

同上，固定规模的版本，通过循环完成TLB的初始化。

（三）验证

将bin文件烧写到SPI flash芯片中，结合在线烧写，查看是否初始化成功。

1. 实验设计

（一）设计方案

1、总体设计思路

即根据所学知识以及所给任务书，完成三个模块的初始化汇编代码。

2、串口初始化模块设计

已经知道，UART\_LCR线路控制器，当它仅最高位是1的时候，即0x80的时候，偏移量为0x0、0x1的寄存器UART\_TLL和UART\_TLH就能组合成为16位的分频锁存器。

实验中SOC中CPU接的时钟频率是33MHz，将其记为CPU\_CLK，串口控制器的时钟频率记为UART\_CLK，则UART\_CLK=CPU\_CLK / {UART\_TLH, UART\_TLL}。波特率计算公式为：UART\_CLK / 16。所以通过计算，我们知道我们让分频锁存器的值为18就可以了，即给偏移量为0x0的UART\_TLL赋值18=0x12，给偏移量为0x1的UART\_TLH赋值0x0即可。

（实在凑不到五页，贴代码了。对应681-694行。）

########

##TODO: CODE 1: set baud rate

########

li v1, 0x80

sb v1, 0x3(v0)

li v1, 18

sb v1, 0x0(v0)

li v1, 0

sb v1, 0x1(v0)

nop

########

##TODO: CODE 1: end

########

3、Cache模块设计

因为要初始化的Cache很多，所以通过循环即可完成。利用cache指令，和Index\_Store\_Tag来初始化。代码已经提供了Index\_Store\_Tag\_D，即对数据cache的初始化，我们再此基础上，增加了Index\_Store\_Tag\_I，来对指令cache初始化化。

固定版本的写相对容易。

循环外部：我们用a0保存了cache的基地址，a1存放了cache的总数128，首先将COP\_0\_TAG\_LO和COP\_0\_TAG\_HI清零。然后令v0和v1成为循环开始的指针和结束的指针，当v1=v0的时候，表示循环结束，跳出。

循环内部：利用store清零，数据cache和指令cache都清零，对偏移0x0和0x1000的一起初始化。每一块cache有32位，所以每循环一次基地址+0x20。

（不多说了，贴代码。新增宏在第757行，其余代码在774-800行。）

#define Index\_Store\_Tag\_I 0x08

########

##TODO: CODE 2: cache init

########

#mfc0 a0, COP\_0\_CONFIG,1

#a0 for flag

li a0, 0x80000000

li a1, 0x1000

mtc0 zero, COP\_0\_TAG\_LO

mtc0 zero, COP\_0\_TAG\_HI

move v0, a0

addu v1, a0, a1

1:

slt a2, v0, v1

beq a2, zero, 2f

nop

cache Index\_Store\_Tag\_D, 0x0(a0)

cache Index\_Store\_Tag\_D, 0x1000(a0)

cache Index\_Store\_Tag\_I, 0x0(a0)

cache Index\_Store\_Tag\_I, 0x1000(a0)

b 1b

addiu v0, v0, 0x10

2:

nop

########

##TODO: CODE 2: end

########

4、TLB初始化模块设计

写完Cache的循环之后，再写TLB，仍旧可以利用循环，完成初始化。

循环外部：设置好循环的参数，a1为循环的总次数32，a0作为游标，从0开始循环一次加1，直到a0=a1结束循环。给PageMask赋值0。另外设置好一个参数a2=0x4000，用于给EntryHi赋值。

循环内部：依次给 EntryHi、EntryLo0、EntryLo1以及Index赋相应的值，然后利用TLBWI指令，将 EntryHi、PageMask、EntryLo0、EntryLo1 写入到寻址到的 TLB 项中。

（依旧是代码部分，对应482-502行。）

########

##TODO: CODE 3: TLB init

########

move a0, zero

li a1, 32

lui a2, 0x4000

mtc0 zero, COP\_0\_TLB\_PG\_MASK

1:

mtc0 a2, COP\_0\_TLB\_HI

mtc0 zero, COP\_0\_TLB\_LO0

mtc0 zero, COP\_0\_TLB\_LO1

mtc0 a0, COP\_0\_TLB\_INDEX

addiu a0, a0, 1

tlbwi

bne a0, a1, 1b

nop

########

##TODO: CODE 3: end

########

（二）验证方案

1、总体验证思路

2、验证环境

3、验证计划

1. 实验实现

（一）实验交付说明

修改了 start.s 文件，放到 pmon\_lab3/Targets/LS1B/ls1b/ 目录下

（二）实现说明

完成了3个模块的初始化。对应代码在，串口：681-694行。Cache：757行，774-800行。TLB：482-502行。

1. 实验测试

（一）测试过程

（二）测试结果

1. 成员分工

小组成员三人，分别看完了任务指导书，一起糊完了代码，糊完了报告，目前等待上板验证正确与否，约2+2+2+？小时。

1. 实验总结

（一）张传奇

（二）谈清扬

（三）孔静

这报告能不能不要有限制5页的要求啊，太特么难写了，简直比实验本身还难写，香菇蓝瘦。

不过给这次用心准备的助教满分！满分！这个指导书，太棒了！Excellent！又详细又讲的很清楚，除了里面可以忽略的一些小细节错误，给满分不过分！

谢谢体系结构研讨课给我们的期中考试留了时间，让我们这次不用花太多时间，再次点赞，O(∩\_∩)O谢谢！

八、参考文献