LAB6: CPU初始化代码

- 王华强
- 2016K8009929035

---

1. 实验内容

> 补充完 PMON 源代码中被删除的 Cache 初始化、TLB 初始化、串口初始化部分，编译后在一个已完成的 SoC 设计上正常启动并成功装载和启动 Linux 内核。
> 要求刚下载完 bit 文件后，PMON 运行时打印的信息无“TLB init Error!!! ”和“cache init Error!!! ”。且能正确装载并启动 Linux 内核。

完成CPU初始化代码的编写，初始化串口波特率，Cache以及TLB.

---

2. 具体实现

代码共计30行，这里结合代码讲解实现的原理.

2.1. 初始化波特率

此实验中使用9600的波特率. 根据公式计算出分频寄存器应该配置为215. 因此，按照讲义对分频寄存器进行配置如下：

```
    # 直接设置指定寄存器的值

    #define M_SERIAL_BASE 0xbfe40000
    #define M_UART_TLL 0x0
    #define M_UART_TLH 0x1
    #define M_UART_LCR 0x3

        li  v0, M_SERIAL_BASE
        li  v1, 0x80
        sb  v1, M_UART_LCR(v0)
        li  v1, 215
        sb  v1, M_UART_TLL(v0)
        li  v1, 0
        sb  v1, M_UART_TLH(v0)
```

2.2. 初始化Cache

在Cache初始化部分，我们使用一个长循环来完成Cache的初始化。此实验构架中Cache行的长度为`M_CACHE_ALINE`, 经过计算得知(四路Cache)需要初始化的缓存地址空间为`M_CACHE_UPPERBOUND`. 于是初始化代码设计如下：

```
    #define M_CACHE_BASE 0x80000000
    #define M_CACHE_UPPERBOUND 0x00002000
    #define M_CACHE_ALINE 0x00000020

        li    v0, 0
        mtc0  v0 ,COP_0_TAG_LO  # set tag to 0
        li    a0, M_CACHE_BASE  # a0 is abse
        li    a1, 0             # a1 is offset

cinit_loop:
        li    a2, 0             # a2 is addr now
        addu  a2, a1, a0
        cache 0x08, 0(a2)       # init data$, inst$ at the same time
        cache 0x09, 0(a2)
        addi  a1, M_CACHE_ALINE # goto next $ line to be inited
        bne   a1, M_CACHE_UPPERBOUND, cinit_loop
        nop
```

在上面的代码中， 同时初始化了数据Cache和指令Cache来简化代码量。

2.3. 初始化TLB

此实验中TLB共计32项，因此只要在初始化代码中做32次循环来将所有项初始化即可。初始化的重点在于将TLB的valid位设成0.

```
    li   v1, 0          # v1 is tlb index
    li   k0, 0x80000000

tlbinitloop:
    li   v0, 0
    # set cp0 regs
    mtc0 v0, COP_0_TLB_LO0
    mtc0 v0, COP_0_TLB_LO1
    mtc0 v0, COP_0_TLB_PG_MASK
    mtc0 v1, COP_0_TLB_INDEX
    mtc0 k0, COP_0_TLB_HI
    addi v1, 0x1
    tlbwi
    bne  v1, 32, tlbinitloop
    nop
```

2.4. Linux启动配置代码

某次测试中的配置如下：

```
ifconfig dmfe0 169.254.17.33
load -r -f bfc00000 tftp://169.254.17.34/gzrom.bin
```

---

3. BUGS与实验日程

20分钟写代码，2小时配环境。

本实验中没有遇到任何BUG。

---

4. 补充说明

本学期最简单实验没有之一。 感谢老师们送分。因为没有BUG也没有什么别的好写的... 就这样吧，像室友一样30行代码1300字实验报告我是学不来...

这一学期就这么结束了吗... ~~虽然Cache估计会很有挑战性不过感觉还是可以一试的~~

~~没性能测试不够快乐啊~~
