计算机综合大实验——CPU部分

刘亚宁

（武祥晋 秦同）

CPU部分

1. 开发环境

硬件环境： 装有Windows7及Linux的微机

软件环境： FPGA开发工具软件Xilinx ISE

Linux下的串口工具minicom

Windos下的串口工具串口调试精灵

Flash的写入工具

1. 任务实现及使用方法

实现： 完成了基本目标，实现了一块能够将自己的Ucore成功运行的32位基于简化mips的多周期CPU，主频25M，支持Flash、串口等基本外设。自带硬件debug小工具，可以指定断点、单步调试、继续运行以及输出通用寄存器和部分其他寄存器的值。

使用方法：烧录CPLD和FPGA以及Flash中的ucore后，将拨码开关设置为0x00010000，按RST可以开始运行。

Debug： 拨码开关高16位为断点控制，当PC低16位值与之相同时即可自动暂停，根据SW(15)值为’1’可以控制系统单步调试。SW(6)控制寄存器组（即通用还是非通用），SW(5 downto 1)控制寄存器编号，非通用编号可查阅代码。SW(0)控制寄存器高位还是低位，此时被选中寄存器的指定16位信息由LED给出。

1. 代码模块
2. multiplication.vhd

使用Xilinx自带的IP核生成的乘法器模块接口。

1. alu.vhd

运算器部分。

1. registers.vhd

通用寄存器组，同时也控制LED的输出

1. extend.vhd

立即数扩展

1. phy\_mem.vhd

访存管理，包括串口、Flash、ram、rom的物理内存映射管理以及串口的读入缓存

1. uart.vhd txmit.vhd rcvr.vhd clkcon.vhd

串口，波特率9600，8O1

1. mmu

tlb模块

1. rom

内置ROM

1. cpu.vhd

主结构体，包括CP0寄存器的定义，时钟的自动实现；指令译码和状态机流转控制。

1. 问题及解决方案
2. 访存地址对齐问题

对于B系列和J系列的指令，立即数的扩展应当是左移两位后进行扩展，而S/L系列的指令不会进行左移。此时应当判断地址是否对齐的问题。通常情况下地址应当低两位均为0，但对于LHU,LB,LBU,SB则不用，其中LHU应当低1位为0。但是在实际的访存过程中，都必须将最低两位置0。

1. J/JAL地址问题

J/JAL都是绝对跳转指令，但事实上并不完全如此，因指令中的立即数并没有达到32位，甚至没有达到30位，故实际上仍需要扩展，而此处的扩展则是相对扩展，即高位保留PC值即可。

1. 串口通信时序问题

老师给定的串口并不稳定，故将txmit及rcvr中开始时序作了微调，以达到较稳定的目的。而同时读入设置了一个缓存。这样在访问串口状态地址时给出的数据就包括可写(1 bit)、可读(1 bit)、可读数据总量(10 bits)。

1. 时钟中断

时钟中断是由COMPARE和COUNT两个寄存器来控制的，在标准规范中COMPARE需在每次时钟中断中读入COUNT值加1写入，但这样控制比较麻烦，所以我们选择完全由硬件控制，软件基本不作处理。

1. Flash写入的问题

由于Flash的写入需要先擦除，而擦除其实是对一整块进行的擦除，故在硬件上给出如下的实现规则：写块首地址的时候进行擦除后再写入，而对于其他地址则直接进行写入。

1. 异常处理

对于中断设置在每一次取指之前，判断并进入中断处理，而其他异常则发现立即进入。所以实质上的EPC都是当前指令。