实验2报告

第24小组

张传奇、谈清扬、孔静

一、实验目的

巩固提高 VerilogHDL 电路设计能力。

进一步提高 MIPS 汇编编程的能力。

理解处理器里软硬件协同工作机制。

进一步提高软硬件的调试经验。

二、实验任务

（一）设计

1、汇编代码设计

（1）工作内容

添加中断部分，以及设置时钟部分内容。

（2）工作目标

能通过中断让时钟暂停，并能完成相应的小时/分钟/秒钟 + 1操作。

2、Verilog代码设计

（1）工作内容

添加硬件连线，以及处理相应硬件信号。

（2）工作目标

消除抖动信号，获取所需要的信号，以及将获得信号处理好并存储，并定义新寄存器处理中断。

（二）实现

1、汇编代码部分

按照中断设置给Cause Status寄存器初始化，打开中断。

在Lab1代码已经完成时钟显示的基础上，增添了暂停功能，根据查询到的信息决定是否暂停。

另写中断处理程序，实现了复位和设置功能。

2、Verilog代码部分

声明了新的输入作为硬件按键输入。

实现消抖模块处理信号。

根据各个控制器的要求重新编写各个寄存器。

（三）验证

模拟键盘信号仿真，以及在FPGA板上运行，检验复位、暂停、设置（包括小时、分钟、秒钟 + 1）是否成功实现。

三、实验设计

（一）设计方案

1、总体设计思路

在Lab1基础上，要求添加按键控制器、中断路由等逻辑。

首先需要增加输入接口，获得输入。其次需要正确获得输入，考虑到实际信号和理想信号的差距，需要设置消抖模块来获取所需信号。然后按需求设置所需寄存器：设置模式寄存器、中断请求寄存器、按键信号寄存器。最后在汇编部分，打开中断，以清零方式复位，以轮询方式处理暂停，以中断方式处理时间 +1 的操作。

2、archlab\_sopc\_top.v设计

（1）设计描述

增加了输入接口KEY\_in，接到键盘上，将KEY\_in[0]取反（复位是低电平有效），接到confreg.v模块的cr06\_in上。

修改了中断路由，将中断请求寄存器缩位或，接到中断路由的0位上。

（2）接口定义

| **名称** | **方向** | **位宽** | **功能描述** |
| --- | --- | --- | --- |
| KEY\_in | IN | 5 | 输入接口，连接键盘的按键 |

3、confreg.v设计

（1）设计描述

增加了三个输入接口，其中cr04为设置模式的标记位（1是设置模式，0不是），cr05为中断请求的寄存器，cr06为按键记录器。

消抖模块：通过比对相邻周期的信号，当获取有效信号并持续20ms时（目的消除抖动），将按键寄存器写入相应位置。

设置寄存器控制逻辑：将cr04[0]取反，设置成信号上升沿触发（了解到复位是低电平有效）。

用cr06保存按键状态。

cr05在平时不断复制cr06。形成中断以后，中断处理程序会向cr05写数据清中断，通过异或完成，使cr05[4:0]，00001复位，00010暂停，00100小时+1，01000分钟+1，10000秒钟+1。同时设置标志位software\_change，当软件清中断以后software\_change为1，当software\_change为1时，cr05不会复制cr06，避免重复中断。software\_change会在侦测到没有按键输入时不断清0。

（2）接口定义

| **名称** | **方向** | **位宽** | **功能描述** |
| --- | --- | --- | --- |
| cr04\_in | IN | 32 | 设置模式寄存器 |
| cr05\_in | IN | 32 | 中断请求寄存器 |
| cr06\_in | IN | 32 | 按键记录寄存器 |

3、start.S设计

按照实验指导书开中断。

在Lab1的基础上修改，显示时钟后，查询cr04设置模式寄存器，如果是设置模式，跳转相应代码，不会进行+1操作，即时钟数字保持不动。

增加中断处理程序，根据cr05中断请求寄存器（00001复位，00010暂停，00100小时+1，01000分钟+1，10000秒钟+1），作出相应处理：复位，清零；暂停，轮询；其余，即相应寄存器+1。

（二）验证方案

1、总体验证思路

因为在Lab1的基础上，所以不用考虑精度，仅需要考虑增添的几个功能。模拟键盘信号仿真，以及在FPGA板上运行，检验复位、暂停、设置（包括小时、分钟、秒钟 + 1）是否成功实现。

2、验证环境

仿真验证中需要我们修改godson system 这个预先配置好的模块来进行测试，和上次不同的是，模拟了按键信号的输入，模拟复位，暂停，恢复，设置按键，观察波形确认功能是否正常。

FPGA 系统验证环节，我们需要直接检查数码管能否正常显示，按键功能是否正常实现。

3、验证计划

| **编号** | **功能点描述** | **考核标准** |
| --- | --- | --- |
| 1 | 复位 | 时钟成功清0。 |
| 2 | 暂停/恢复/设置 | 第一次点，时钟暂停；再次点，时钟恢复。 |
| 3 | 小时/分钟/秒钟 +1 | 能判断是否在设置模式下，然后再决定是否进行相应+1操作。 |

四、实验实现

（一）实现交付说明

修改的文件有：

archlab\_sopc\_top.v，放置于\rtl\_led\TOP\

confreg.v，放置于\rtl\_led\CONFREG\

start.S，放置于\simu\soft\func\_led\

（二）实现说明

1、archlab\_sopc\_top.v设计

13行：增加了输入接口KEY\_in作为键盘的按键输入。

203-205行：写入中断，并将中断请求寄存器缩位或，接到中断路由0位上。

504行：增加cr系列输入连线。

510行：将键盘输入连接到cr06\_in，并将末位取反（实际上复位信号低电平有效）。

2、confreg.v设计

116行：增加cr系列输入连线。

196-234行：消抖逻辑，通过比对相邻周期信号，以及有效信号持续20ms才写入按键寄存器来消抖。

236-243行：设置寄存器控制逻辑，目的是让设置键信号上升沿时触发。

246-281行：写入逻辑，对cr05中断请求寄存器的写入。

3、start.S设计

60-63行：开中断

185行：修改Lab1逻辑，查询cr04，并作出相应操作。

316行：中断处理程序，读取cr05中断请求寄存器，按照原因分别处理。

五、实验测试

（一）测试过程

1、ISE仿真

（1）Problem

秒+1中断后行为不正常，会卡死，并且发现只是处理秒+1的时候才会发生。

Solution：

秒+1是中断处理程序的最后一行，其下一条指令是yourcodeend上面的jr，所以会发生错误的行为，eret后应该加nop

（2）Problem:

秒+1的时候小时+1。

Solution：

发现是写入内存时的偏移量写错。

（3）Problem：

访问写入cr等寄存器时不正常。

Solution：

忘记按字节编址，一个寄存器4个字节，改成\*4之后正常。

2、FPGA

（1）Problem：

运行时，什么都不显示。

Solution：

忘记复位键是低电平有效，对其取反就解决问题了。

（二）测试结果

全部实现。

六、成员分工

组员共同完成了代码+仿真调试+上板+实验报告四个部分，共计大约15+小时。

（一）组员：张传奇

主要参与了代码+仿真调试。

（二）组员：谈清扬

主要参与了代码+上板调试。

（三）组员：孔静

主要参与了调试+实验报告。

七、实验总结

（一）组员：张传奇

实验最重要的是理解实验要求，我们在设计的时候做了极大的简化，注意到设置和暂停的一致性，时间的周期循环性用+1实现设置，让逻辑变得十分简单并且易于实现，节约了许多时间。

（二）组员：谈清扬

第一次仿真验证成功之后觉得可以上板成功了，但是上板之后毫无反应，甚至数字管灯都不亮了。后来发现是硬件按键的逻辑理解错误，正负相反，改正之后就可以使用。要仔细联系软硬件之间的关系。

（三）组员：孔静

我认为该实验的主要难点，在于如何将键盘上的信号连接到我们设计的寄存器，并获得正确的稳定的有效信号，以及了解FPGA的信号，比如复位键是低电平有效。

八、参考文献

无。