北京工业大学

2022- 2023学年 第二学期

软件学院

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称： | 嵌入式软硬件综合设计 | | |
| 报告性质： | □作业报告 █实验报告 | | |
| 学号： | 20080216 | 姓名： | 黄嘉祺 |
| 任课教师： | 李达 | 课程性质： | 专业限选课 |
| 学分： | 2 | 学时： | 32 |
| 班级： | 软件工程200802 | 成绩： |  |
| 小组成员： | 独立完成 | | |
| 教师评语： |  | | |

2023 年 5 月 8 日

|  |
| --- |
| 实验名称 软件动态配置流水灯 实验日期 2022.03.28  实验报告要求：1.实验目的 2.实验内容 3.实验需求  4.实验步骤 5.实验结果 6.程序源码  7.实验体会 |
| 1. 实验目的   1.了解Zedboard开发板，掌握并熟悉开发流程；  2.基于UART和GPIO实验基础上，根据设计功能建立应用软硬件工程。  3.通过软件代码实现流水灯的动态配置控制。   1. 实验内容   1.根据设计功能建立Vivado硬件工程（可以直接利用实验project2工程）  2.在SDK环境中创建软件代码工程，分部实现指令配置、参数配置；然后生成构bit流文件下载到开发板中，测试检验功能；   1. 实验需求   //根据作业要求自己定义具体参数  1.通过UART接收字符指令  ‘l’：程序进入配置状态；  num：数值配置流水灯sleep周期；  1对应0.1s，2对应0.5s，3对应1s，4对应2s；  ‘s’：程序进入运行状态；  2.通过switch 8位开关，配置LED循环码  4.实验软件功能结构设计简图  5.实验步骤   1. **创建vivado项目**   启动了Vivado后，点选Create New Project 。    接下来指定好你的项目名称与路径    选择 RTL Project，勾选Do not specify sources at this time    在开发板选项中，选择board->ZedBoard,单击next;    单击finish,完成项目建立.   1. **建立Block Design**   当我们的设计需要用到 Zynq 的处理器系统(Processing System, PS)时候，就需要透过 Block Design 来建立我们的电路设计，首先点选IP Integrator -> Create Block Design。接着点选OK，建立我们的block design;    点选Add IP按钮去增加我们需要的IP核;    输入Zynq ，双击将ZYNQ7 Processing System 加入到我们的Block Design，并点选 Run Block Automation 对Zynq处理器进行一些设定；进入到 Run Block Automation 的设定页面后，确认processing\_system7\_0有被勾选到，并且 Cross Trigger In以及Cross Trigger Out都是Disable的状态，点选Ok结束设定。然后双击zynq，定义中断，选择interrupts-- interrupts port--fabric interrupt—PL-PS interrupts port 下勾选IRQ\_F2P[15:0]，这样Zynq处理器就使能了中断变量。      可以看到zynq上多了IRQ\_F2P[0:0]和fixed\_io、ddr外扩接口。   1. **添加ZYNQ7 Processing System、2个GPIO IP核（控制LED及SWITCH）**   点选 Add IP 按钮继续增加我们需要的IP核，输入gpio，添加AXI\_GPIO用来对可程序逻辑(Programmable Logic, PL)区域的LED进行控制，这里我们连续添加3个，即axi\_gpio\_0、axi\_gpio\_1、axi\_gpio\_2，完成后点选上方的Run Connection Automation 按钮;    在Run Conenction Automation窗口内，我们选择gpio类型配置：所有gpio都挂接axi勾选上；按照前面设计设定的axi\_gpio\_0选择led\_8bits，axi\_gpio\_1选择sws\_8bits，axi\_gpio\_2选择btns\_5bits，这样gpio都挂接上了。    双击axi\_gpio\_2，勾选中断enable interrupt    将zynq和axi\_gpio\_2，中断连接。这里可以直接在图选中中断引脚IRQ\_F2P[0:0]，鼠标右键选择make connection，选择    按快捷键Ctrl+S保存，接下来点选Validate Design 按钮，我们要确认我们的Block Design 没问题才能够继续往下走。    我们检查一下设计是否有遗漏，zynq+led+sw+btn，中断配置，，正常来讲不会有啥问题才对，我们结束Block Design的工作。  选中.bd文件，右键点击Generate Output Products；    然后就ok自动编译生成，当中warning可以先不管它。透过Block Design产生我们的HDL wrapper，对Block Design档案点选右键，选择Create HDL Wrapper。它会根据你项目设定的语言(VHDL或是Verilog)来产生相对的HDL程序代码。选择第二个 let vivado manage wrapper；    好了后，假设你的Block Design档案叫做 design\_1.bd，那就会产生design\_1\_wrapper.v或是 design\_1\_wrapper.vhdl这样的档案。   1. **自动连接后生成bit流**   前面的处理都好了后，接下来点选 Program and Debug -> Generate Bitstream 去让Vivado将这个项目产生出比特流 (bitstream)，ZYNQ会根据bitstream的信息对FPGA进行设定。在点击Generate Bitstream后会提示没有Implement文件，单击“Yes”进行综合执行。当bitstream产生完成后，将刚刚产生的硬件信息输出给Xilinx SDK去    当bitstream产生完成后，将刚刚产生的硬件信息输出给Xilinx SDK去点选 File -> Export -> Export Hardware,确定勾选Include bitstream后，点选 Ok；     1. **导入bit流在SDK中进行软件部分开发**   点选 File ->Launch SDK, 完成后，Xilinx SDK程序会自动启动。        选择File -> New -> Application Project去建立新的项目; 命名这个项目叫做gpioproj，其他默认单击next;    选择Hello world,单击finish。当项目建立完成后，会自动打开gpioproj\_bsp里面的 system.mss，里面会显示我们所用的外围范例程序代码以及使用手册的链接,可以点选这些链接来了解这些外围要怎样使用。  双击helloworld.c这是模板自带打印的程序，我们可以直接在里面编程。    6.程序设计源代码（关键节点注释）          7.实验调试和结果验证  C:\Users\23969\AppData\Local\Temp\WeChat Files\21a2cfe341ac5b6a1ff2eb08e2e2455.jpg C:\Users\23969\AppData\Local\Temp\WeChat Files\65b7ce947ea4a0b4c99bf8cddecd2e5.jpg  C:\Users\23969\AppData\Local\Temp\WeChat Files\dd124e93382996347b83dbe3a8039bf.jpg C:\Users\23969\AppData\Local\Temp\WeChat Files\b2e4178c95141eccc2c11fb79896bc3.jpg    8.实验总结  在实验中我完成了跑马灯的实现，并且通过Xilinx系统搭了一个框架出来，通过在C语言中引入库函数来控制架构中的引脚，控制中断等，最后实现了跑马灯的功能。  这次试验让我对开发的流程基本熟悉，知道了怎么编辑架构，怎么用C语言操作开发板，以及如何编写中断处理程序等，受益颇多。 |

|  |
| --- |
|  |
|  |