

公司名称: _____ ID: _____ 开始日期: 2021年8月17日, 星期二
截止日期: 2021年8月19日, 星期四

与Zybo的软件和硬件联合设计, 2021年夏季高峰 实验室#7中断驱动乒乓球游戏

这是一个单独的实验室。每个学生必须自己表演, 并演示这个实验室以获得学分。延迟的实验室提交将被接受, 每天延迟的评分将降低10%。

1 所实现的目标

- 第1部分: 执行Zynq书教程2, ZynqSoC设计的下一步, 并提交关于这部分的备忘录。
- 第二部分: 修改实验室#6LED乒乓球游戏, 使用输入更改中断和计时器中断。输入更改中断服务将处理四个按钮和四个滑动开关。
- 计时器中断服务将处理LED显示。主程序只有在初始化后才能处理终端显示。
- 每当滑动开关发生变化时, 应发生中断, 以立即改变计时器的延迟时间。按钮中断会导致游戏播放状态发生改变。
- 创建一个ASM图表或伪代码来描述中断驱动的乒乓球程序的行为。

2 可用的源代码作为一个模板

interrupt_counter_tut_2D_new.c是启动本实验室第2部分的模板。请注意, 代码比原始的教程程序更模块化, 因为每个设备都由一个子例程初始化, 该子例程也会初始化本地中断。GIC初始化是仅用于GIC和ARM核心CPU中断配置的一个单独的子例程。

3 在2021年8月19日星期四之前的演示和报告



提交一份关于ZynqBook教程2的一些典型屏幕截图的pdf备忘录, 以显示您已经完成了它作为这个实验室的第一部分。演示您的棋盘与中断驱动的乒乓球游戏, 并提交您的ASM图表或伪代码的pdf副本以及您最终的源代码。

4 第一部分: Zynq书教程2: ZynqSoC设计的下一步

获取本教程的源代码文件夹, The_Zynq_Book_Tutorials_Sep_14.zip从

<http://www.zynqbook.com/download-tuts.html>. 将

The_Zynq_Book_Tutorial_Sources/sources/zynq_interrupts文件夹复制到C:\Xilinx\SHCodesign2021/。

Xilinx > SHCodesign2021 > zynq_interrupts			
Name	Date modified	Type	Size
 interrupt_counter_tut_2B	8/15/2021 11:17 PM	C File	5 KB
 interrupt_counter_tut_2D	8/15/2021 11:17 PM	C File	6 KB

按照2014年4月[6]的ZynqBook教程v1.2中的教程#2，完成中断教程示例练习2B和2D。写一份包含本实验室典型屏幕截图的备忘录，以显示您已经完成了本教程。

5 第二部分：中断驱动的乒乓球游戏

通过中断服务实现实验室6的乒乓球游戏。你的实验室#7游戏应该有与实验室#6相同的行，但响应计时器溢出和按钮和滑动开关改变通过中断。

5.1 中断驱动的LED乒乓球游戏的体系结构

这个游戏由四个主要的子例程组成：主()，按钮中断处理程序，滑动开关中断处理程序和计时器中断处理程序。主程序将设置中断，并初始化所有端口和计时器。这三个中断处理程序将处理按钮、滑动开关和计时器中断。然后，主程序将打印在游戏状态和终端的结果。所有xil_printf()或printf()必须在主()循环中，以便中断处理程序花费尽可能少的时间。同时使用xil_printf()和printf()不是一个好主意，因为它们独立于每个()，并且可能不会以正确的顺序显示文本。

按钮中断处理程序将决定按下哪个按钮，并设置游戏状态和相应的结果。幻灯片开关中断处理程序将加载计时器加载寄存器。注意：XGPIO_IR_CH1_MASK指定中断的来源来自gpio接口的通道1。如果源来自gpio接口的通道2，则应该使用XGPIO_IR_CH2_MASK。

计时器中断处理程序将更新LED显示。

全局状态和结果变量用于这三个主要子程序之间的通信。您的代码不应该有任何魔法数字或魔法端口。

5.2 ASM图表或伪代码

在开始编写代码之前，请为中断驱动的乒乓球代码创建一个ASM图表或一个伪代码。将此图表或伪代码作为可交付成果的一部分。

5.3 对乒乓球游戏的要求

游戏的外部行为主要与实验室#6相同，除了按钮和滑动开关，它们是中断驱动的。一旦按下重置按钮，游戏就会被重置。无论游戏是否结束，只要按下发球按钮，游戏发球就会开始。滑动开关的更改将立即改变计时器的延迟时间。

- 1) 暂时按下重置按钮，BTN2应该会重置游戏的分数。分数应该显示在笔记本电脑的SDK串行端口窗口中。
- 2) 瞬时间按发球按钮BTN1应提供“球”，即四个LED线(LD3)的左端出现一个点亮LED，球(点亮LED)开始从LED线的左端(LD3)向右端(LD0)移动。每个球员将轮流发球。

- 3) 当球（点亮的LED）到达最右位置（LD0）时，应按下可视为右球手的桨的右按钮（BTN0）（此时不能提前推！）为了把球打回另一个方向。如果在这段时间里没有按下正确的按钮，或者在球到达LD0之前按得太早，或者在球超过LD0之后按得太晚，球应该从4个led线的末端“掉下来”（球消失……没有led灯仍然亮），发球结束。如果发生这种情况，比赛将结束，在左边球员的得分中应该加一分。
- 4) 假设球被正确地向左回击，那么当球最终回到最左的LED位置（LD3）时，必须按下左按钮（BTN3），以便将其送回另一个方向。如果在球处于LD3的关键时刻没有按下左按钮，那么右球员的得分就会增加。
- 5) 一旦发球结束，右边或左边的球员得到一分，玩家必须等待发球按钮（BTN1）再次被击下。然后球必须在另一个方向发球。服务的方向应交替与每按下中心按钮。
- 6) 游戏速度（游戏难度）应该在16个步骤中使用4个车载数码数码开关进行调整。15的设置应该非常慢，0的设置应该相当快，尽管仍然（几乎）专家玩！这个速度变化是瞬间的，通过滑动开关改变中断，一旦滑动开关改变他们的状态。

6 中断驱动的LED乒乓球游戏的体系结构

这个游戏由三个主要的子程序组成：主（），按钮中断处理程序和计时器中断处理程序。主程序将设置中断，并初始化所有端口和计时器。然后，主程序将打印在游戏状态和终端的结果。这两个中断处理程序将处理按钮和计时器中断。

按钮中断处理程序将决定按下哪个按钮，并设置游戏状态和相应的结果。

计时器中断处理程序将更新LED显示。

全局状态和结果变量用于这三个主要子程序之间的通信。

7 参考文献

1. 专用定时器设备驱动程序的皮层A9，磨损定时器v2_0，Xilinx。
2. 通用I/O(XGpio)设备驱动程序，gpiov4_0，Xilinx。
3. 如何使用ZynqSoC上的中断，Xcell期刊87
4. xil_exception.c, standalone_v4_2, 来自C:
\\Xilinx\\SDK\\2014.4\\data\\embeddedsw\\lib\\bsp\\standalone_v4_2\\src\\co
rtexa9
5. 第三章：中断控制器，Cortex™-A9MPCore，版本：r2p2，技术参考手册
6. Zynq图书教程，v1.2-2014年9月，英国斯特拉斯克莱德大学，格拉斯哥，苏格兰