

# 西安邮电大学

## 毕业设计（论文）

题目： 基于 FPGA 的拔河游戏机设计

学院： 自动化学院

专业： 自动化

班级： 自动 1403

学生姓名： 杨磊

学号： 06141099

导师姓名： 姚霁 职称： 讲师

起止时间： 2017 年 12 月 5 日至 2018 年 6 月 10 日

---

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于FPGA的拔河游戏机设计》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：      年    月    日

## 西安邮电大学本科毕业设计(论文)选题审批表

申报人	姚霖	职称	讲师	学院	自动化学院			
题目名称	基于 FPGA 的拔河游戏机设计							
题目来源	科研				教学	√	其它	
题目类型	硬件设计	√	软件设计		论文		艺术作品	
题目性质	应用研究		√		理论研究			
题目简述	(为什么申报该课题) 设计一个能进行拔河游戏的电路。使用硬件描述语言进行设计，并仿真，最终在 FPGA 平台上实现。							
对学生知识与能力要求	1.熟练掌握 FPGA 的设计流程及方法； 2.具有硬件电路调试基础； 3.有较强的自学能力和分析能力。							
具体任务以及预期目标	(应完成的具体工作，预期目标和成果形式) 使用发光二极管表示拔河的“电子绳”，开机后只有中间一个发亮，即拔河的中心点。游戏甲乙双方各持按钮，谁按得快，亮点向谁的方向移动，亮点移到任一方终端发光二极管则获胜，并发出声响，用数码管显示获胜者的盘数。							
时间进度	2017.12.5~2018.1.15 了解课题的背景知识，具体细节，明确所涉及的内容，确立毕业设计的题目，撰写开题报告； 2018.1.16~2018.3.12 学习有关本课题设计的相关知识，包括 FPGA 硬件电路及相关 EDA 软件的学习； 2018.3.13~2018.4.1 规划设计方案，硬件选型及购买，搭建硬件电路； 2018.4.1~2018.5.10 编写拔河游戏机设计的硬件代码、仿真，硬件调试； 2018.5.10~2018.5.30 系统联调，完成实物设计，同时撰写毕业设计论文； 2018.6.1~2018.6.10 修改、打印毕业设计论文，准备 PPT，完成毕设答辩。							
系(教研室)主任签字	2017 年 12 月 9 日			主管院长签字		2017 年 12 月 9 日		

## 西安邮电大学本科毕业设计（论文）开题报告

学生姓名	杨磊	学号	06141099	专业班级	自动 1403
指导教师	姚霖	题目	基于 FPGA 的拔河游戏机设计		
<p>选题目的（为什么选该课题）</p> <p>随着 FPGA 技术的发展，基于 FPGA 的产品设计越来越多样化，促使人们使用硬件编程语言设计各种各样的器件，通过 Verilog 设计产品最后在 FPGA 上实现逐渐成为 IC 设计的主流。拔河游戏机作为一种竞赛工具，可以很好的保证比赛的公正性，因此越来越多用于各种比赛场合。相较于其他产品，基于 FPGA 的抢答器具有开发周期短、成本低、稳定性高等优点。根据需要，在 FPGA 系统中可以更方便的实现比分显示、去抖等功能。所以，基于 FPGA 的设计在锻炼硬件编程以外对于拔河游戏机的应用与推广有着很现实的意义。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：Verilog HDL 编程，数字电路基础，微机原理与接口技术</p> <p>掌握工具：Modelsim 仿真软件的应用，EDA 软件 QUARTUSII 的使用方法，FPGA 的结构和设计流程</p> <p>资料积累：参考资料《数字系统设计与 Verilog HDL》</p> <p>软硬件条件：Verilog 编译环境，FPGA 开发板</p>					
<p>要研究和解决的问题（做什么）</p> <p>用 FPGA 设计一个拔河游戏机。实验电路使用 7 个发光二极管显示拔河结果，上电后只有中间的一个发亮。两个输入按键，按一次按键，亮点就向按键方移动一次，谁按得快，亮点就向谁的方向移动。两点移到任一方最后一个二极管时，表示该方获胜，此时按键不起作用、输出保持并用数码管显示比分，复位后亮点才回到中心。</p>					
<p>工作思路和方案（怎么做）</p> <p>（1）需要计数器对双方按键次数进行计数，通过比较计数结果来判断亮点的移动方向；</p> <p>（2）设计一个分频器对外部时钟进行分频；</p> <p>（3）按键控制：因为要对按键方有效按键次数进行计数，需要在按键部分增加去抖电路来增加比赛的公平性；</p> <p>（4）亮灯控制：通过比较计数结果判断亮点移动方向，移到终端时比赛结束。</p>					
<p>指导教师意见</p> <p style="text-align: right; margin-top: 50px;">签字：_____ 年 月 日</p>					

## 西安邮电大学毕业设计 (论文)成绩评定表

学生姓名	杨磊	性别	男	学号	06141099	专业 班级	自动 1403
课题名称	基于FPGA的拔河游戏机设计						
指导教师 意见	(从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核)   评分(百分制):            指导教师(签字): _____ 年    月    日						
评阅 教师 意见	(从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核)   评分(百分制):            评阅教师(签字): _____ 年    月    日						
验收 小组 意见	(从毕业设计质量、准备、操作情况等方面进行考核)   评分(百分制):            验收教师(签字): _____ 年    月    日						
答辩 小组 意见	(从准备、陈述、回答、仪表等方面进行考核)   评分(百分制):            答辩小组组长(签字): _____ 年    月    日						
评分比例	指导教师评分 20(%) 评阅教师评分 30(%) 验收小组评分 30(%) 答辩小组评分 20(%)						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	毕业论文(设计)最终成绩(等级):   学院答辩委员会主任(签字、学院盖章): _____ 年    月    日						

---

# 目 录

摘 要 .....	I
Abstract.....	II
第一章 绪论 .....	1
1.1 课题背景 .....	1
1.2 课题研究的内容 .....	1
1.3 论文结构 .....	1
第二章 理论基础 .....	3
2.1 FPGA 简介 .....	3
2.2 硬件描述语言 Verilog 简介 .....	3
2.3 FPGA 设计流程 .....	3
第三章 系统设计与实现 .....	5
3.1 设计要求 .....	5
3.2 设计思路 .....	5
3.3 分模块实现系统功能 .....	6
3.3.1 分频模块.....	6
3.3.2 按键处理模块.....	7
3.3.3 按键频率比较模块.....	9
3.3.4 亮点控制模块.....	11
3.3.5 蜂鸣器控制模块.....	12
3.3.6 分数显示模块.....	13
3.4 系统仿真 .....	15
3.4.1 Modelsim 软件介绍 .....	15
3.4.2 在 Modelsim 中完成软件仿真 .....	15
第四章 硬件实现 .....	18
4.1 Quartus II 软件介绍 .....	18
4.2 程序下载与设计完成 .....	18
第五章 结论 .....	21
5.1 总结: .....	21
5.2 展望: .....	21

---

致 谢 .....	22
参考文献 .....	23
附录 1 .....	24

## 摘 要

FPGA的开发相对于传统PC、单片机的开发有很大不同，相较于其他设备，FPGA具有更强的稳定性和更短的开发周期；与DSP相比，FPGA打破了顺序执行的模式处理，处理能力是DSP解决方案的多倍；与ASIC相比成本更加低廉。

本毕业设计是一个基于FPGA设计的拔河游戏机，利用EDA技术，自顶向下设计工程，系统由时钟分频、按键去抖、按键计数、按键比较、亮点输出、分数显示等六个模块组成。通过各个模块分别Verilog HDL为硬件语言，用Modelsim进行功能仿真，Quartus2 工具软件进行系统联调和程序的下载。

本系统可实现拔河游戏机的全部功能。用一排亮灯模拟拔河情况；数码管来显示比分，可实现每局的清零和系统的复位等功能，在输入部分，可用压力传感器或脑电波灯设备进行控制，本毕业设计通过两个按键的按下频率决定胜负。

基于FPGA的拔河游戏机设计将电子系统设计和娱乐生活结合在一起，使电子技术有了现实意义，在实际应用中更体现了FPGA相比其他设备的优势所在，在运用中又对原计划提出了更高的要求。

**关键词：**拔河游戏机；FPGA；Verilog HDL



## Abstract

The development of FPGA is much different from that of traditional DSP, and it has stronger stability and shorter development cycle than other devices, compared with DSP, it breaks the mode of sequential execution. Processing power is many times that of DSP solutions; it is cheaper than ASIC.

This graduation design is a tug-of-war game based on FPGA design. The system is composed of six modules: clock frequency division, key shake, key count, key comparison, bright spot output, score display and so on. Through each module respectively Verilog HDL as the hardware language, the Modelsim carries on the function simulation and the Quartus2 tool software carries on the system link reconciliation program download.

The system can realize all the functions of tug of war game. A row of lights is used to simulate tug of war, the digital tube to display the score, can realize the function of clearing every game and the reset of the system, etc. In the input part, it can be controlled by pressure sensor or brainwave lamp equipment. The graduation design through the press of two buttons frequency to determine the victory or defeat.

The tug-of-war game design based on FPGA combines the electronic system design with the entertainment life, which makes the electronic technology have practical significance. It also reflects the advantages of FPGA compared with other devices in the practical application. In the application of the original plan put forward higher requirements.

**Key words:** tug of war, FPGA, Verilog HDL

## 第一章 绪论

### 1.1 课题背景

现代社会，信息产品作为时代的产物逐步成为生活、生产中重要的应用工具，而信息技术也在这种趋势下飞速发展，更新换代也越来越快，其中，不仅微电子工艺水平被提高，电子设计开发技术也跟随潮流得到发展，这就是电子设计自动化即 EDA 技术的出现。

EDA (electronic design automatic) 作为电子制造业的一种新兴技术，主要借助计算机的运算速度，通过 EDA 技术自动完成电子系统的设计，包括程序的综合以及电子线路的搭建<sup>[1]</sup>。开发人员可以利用它来完成软件或硬件的开发、测试，辅助生成电路设计，在开发周期以及生成系统的稳定性各方面都有了长足的发展。EDA 技术的引进与应用，节省了不少时间与人力，渐渐成为电子系统设计必不可少的一种工具。

随着信息技术在生产、生活、娱乐等方面的应用，信息产品越来越多样化，而且性能越来越稳定，迭代更新速度越来越快。使得越来越多的功能可以由电子产品实现。

基于 EDA 技术，拔河游戏机作为一种竞赛工具，可以很好的保证比赛的公正性，因此越来越多用于各种比赛场合。目前，已有基于单片机或数字电路设计的拔河游戏机，相比较而言，基于 FPGA 的拔河游戏机比单片机系统开发周期短，稳定性高，更擅长处理多线程任务；而直接基于数字电路的设计难度大，稳定性差；用 arm 实现拔河游戏机成本太高，需要投片生产，且不利于维护与修改配置。根据需要，在 FPGA 系统中可以更方便的实现比分显示、去抖等功能。所以，基于 FPGA 的设计在锻炼硬件编程以外对于拔河游戏机的应用与推广有着很现实的意义。

### 1.2 课题研究的内容

本毕业设计使用 FPGA 器件实现了一个由两个按键控制的拔河游戏机，通过按键的快慢来决定亮点的移动，当亮点移动到某一段时一局结束并通过数码管显示比分。有复位按键可对亮点和比分进行清零。可以用于娱乐、比赛等场合。

### 1.3 论文结构

本毕业论文针对 FPGA 的设计过程，共分为六个章节分别讲了绪论、基础知识、软件与硬件的实现以及最后的总结，各章节内容如下：

第一章：绪论，主要讲述了课题的背景以及涉及到的领域整体发展趋势，初

步介绍课题的内容，大致说明论文的整体结构，读者可从此了解本课题要解决的问题。

第二章：要完成拔河游戏机的实现，需要一些理论基础的支撑，本章主要介绍了涉及到的软硬件、硬件描述语言的基础知识。

第三章：本论文的重点，开始着手完成拔河游戏机功能的实现，主要介绍了系统的设计思路与模块的划分。之后每一节为一个模块，依次实现了分频器、按键处理、按键频率比较、亮点控制、分数输出、报警器，并且在每一章都单独作出电路管脚、程序代码、实现原理以及波形图的仿真，以保证整体的正确性。章节最后介绍了 **Modelsim** 的使用方法，有关拔河游戏机的 **Verilog** 代码都将在该软件中编写、编译、纠错、仿真、调试。同时在 **testbench** 中对所设计的程序进行测试，给出整体仿真波形。

第四章：完成最后的硬件工作，使用 **Quartus II**配置引脚并将程序烧进开发板，进行硬件调试完成设计要求。

第五章：最后的结束语和展望分别对本毕业设计进行总结和提出新的研究内容。

## 第二章 理论基础

### 2.1 FPGA 简介

基于 EDA 技术，可以很方便的在 FPGA（field programmable gate array）中搭建电路来实现设计需要的功能。FPGA 是指现场可编程门阵列，具有掩膜可编程门阵列的通用结构，由逻辑功能块排成阵列，并由可编程的互连资源连接这些逻辑功能块来实现不同的设计<sup>[2]</sup>。FPGA 作为一种特殊的 ASIC，可以大大缩短工程开发周期，减小产品开发难度，同时具有很好的稳定性，还免去了电路焊接和调试的工序，所以 FPGA 这一项技术有着良好的发展前景。程序的正确性可在软件仿真中得以保证，系统稳定性得以提升。

对于 FPGA 的设计，较为主流的均采用自顶向下的设计方法，设计人员只需将系统分为几个模块，每个模块只完成某单一功能，通过接口连接各个模块使系统能够工作，由于自顶向下的设计是通过行为描述来进行定义，因此设计人员不需要关心实现渠道就可以在综合库的帮助下完成对产品系统的设计，大大减轻了开发人员的工作量，缩短了开发周期。

随着 FPGA 技术的发展，基于 FPGA 的产品设计越来越多样化，基于 FPGA 的器件几乎能完成任何电子产品的功能，如电子时钟、拔河游戏机甚至实现各种算法。基于 FPGA 的电子产品设计实现逐渐成为 IC 设计的主流<sup>[3]</sup>。促使人们使用硬件编程语言设计各种各样的器件。

### 2.2 硬件描述语言 Verilog 简介

Verilog HDL 是一种常用的硬件描述语言，用于从算法级、门级到开关级的多种抽象设计层次的数字系统建模<sup>[4]</sup>。并且，所设计的电子系统的复杂性还可以介于简单的门和完整的电子系统之间。数字系统能够按层次描述，并可在相同描述中显式的进行时序建模<sup>[5]</sup>。

Verilog 从 C 语言里继承了多种操作符和结构，具体有三种描述方法，分别为：行为描述、结构描述、数据流描述。

### 2.3 FPGA 设计流程

为便于设计，将所设计的系统层次化，分为几个不同的模块，每个模块的功能再分别设计，通过定义接口连接这些模块，这就是典型的自顶向下的设计方法。目前该方法已大量运用在工程设计中，具体实现过程如图 1.1 所示。每一步的实现过程与要求如下：

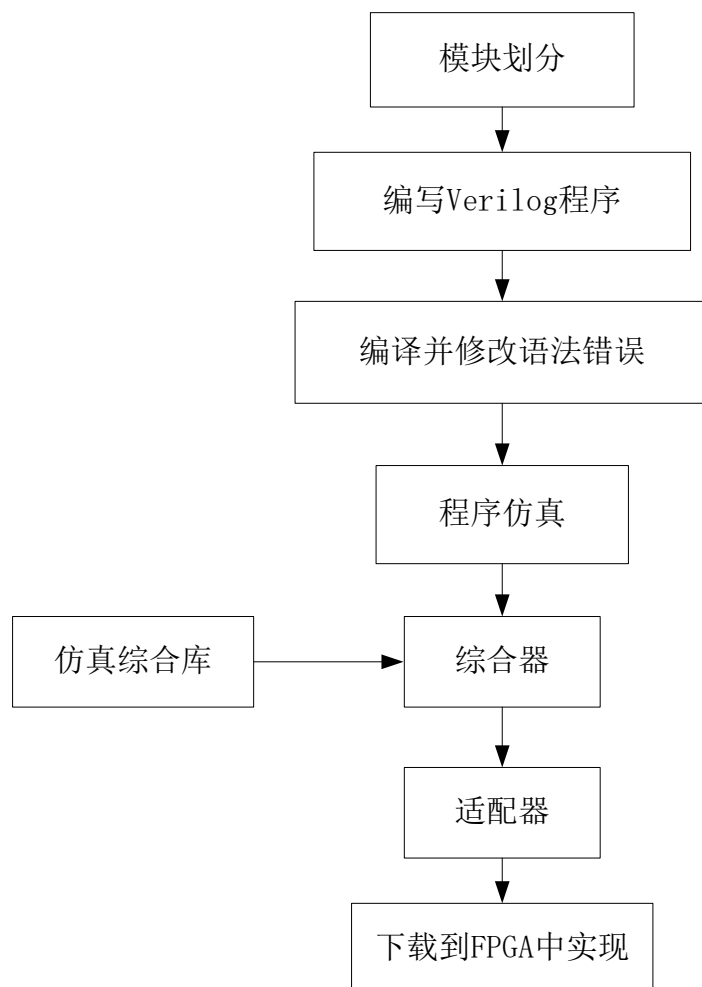


图 1.1 FPGA 设计流程图

（1）对要设计的系统进行整体划分，分成功能较容易实现的几个模块。

（2）采用 Verilog HDL 进行编码，所有的功能将在该语言中体现，采用这种输入方式，可以直观简便的看到要实现的功能且更易于修改纠错<sup>[6]</sup>。

（3）在计算机中将已编好的 Verilog HDL 程序进行编译，软件在编译的同时可对程序中的语法自动进行纠错。

（4）在 EDA 软件对程序进行仿真，通过波形图的数值与实时性来判断程序运行是否有误，对于完整的设计来说，仿真的正确性保证了后续步骤的正确运行<sup>[7]</sup>。

（5）对程序的综合，可将 Verilog HDL 程序综合成为门级描述的文件，这一步需要厂家几桶综合库，最后将把对电路的语言描述转化为实际的电路图。

（6）通过适配器完成引脚的配置、资源分配等任务，保证所完成的设计能运行在所给的器件上。

（7）最后用下载器把编程文件下载到开发板中。

## 第三章 系统设计与实现

### 3.1 设计要求

本毕业设计基于 FPGA 设计了一个拔河游戏机，该拔河游戏机使用七盏 LED 灯来代表拔河的“电子绳”，复位或开机后中间的灯亮，这是拔河的中心，A、B 两人各持一个按键，谁按得快，亮点就向谁移动，当亮点移到终端时决出胜负，此时 A、B 按键无效，蜂鸣器发出声响，同时数码管显示比分。

### 3.2 设计思路

按照 Verilog 自顶向下的语言风格<sup>[9]</sup>，可以把系统拆分成时钟分频、按键去抖、按键比较、亮点控制、比分显示、蜂鸣器六个模块，这样，在设计思路上不容易混乱，也保证了仿真的正确，各模块的划分与功能如图 3.1 所示。各模块各自的功能实现如下：

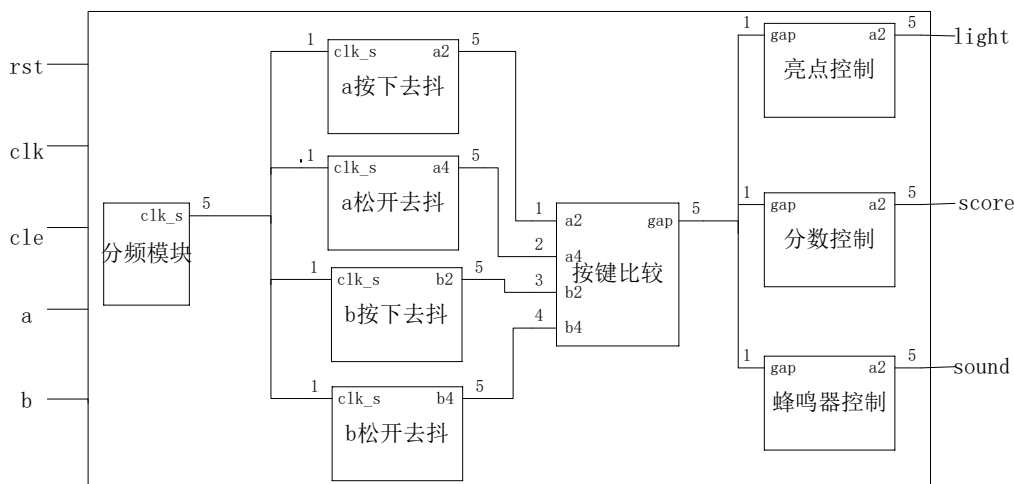


图 3.1 系统整体设计思路

（1）分频模块：主要用于按键去抖时过滤掉机械抖动部分，同时在蜂鸣器控制时还将用于蜂鸣器输入频率。

（2）去抖模块：分为四个部分，分别用于按键 a 的按下与释放的去抖、按键 b 的按下与释放去抖。配合 clk\_s 将抖动部分过滤掉后，输出分别为按键 a 已按下与已释放，按键 b 的已按下与已释放，四个输出分别用 a2、a4、b2、b4 表示。

（3）按键比较模块：输入为已去抖的按键信号，通过记录按键 a 与按键 b 有效点击次数并计较来实时记录比赛情况，经过处理后的输出为 gap，gap 的值表明了两按键点击频率的差距。

（4）亮点控制模块：每次只允许一盏灯被点亮，表明当时“拔河”的赛况。当亮点移到某一端时比赛结束。

（5）比分输出模块：控制两个数码管，分别记录两个玩家的比分。

（6）蜂鸣器控制模块：该模块作报警使用，蜂鸣器响起表明比赛已结束。系统整体设计思路如图 3.1 所示，以下为系统引脚及功能：

输入：input clk：给定时钟输入，用于分频输入和模块的触发信息。

input a,b：玩家按键，通过这两个按键的按下速度决定胜负

input rst：系统复位按键

input cle：亮点复位，该案件不对分数复位，表明一局结束

输出：output reg[3:0] score\_1,score\_2：分数输出，分别代表两个玩家的比分

output reg[6:0] light：两点控制，共六位，每一位对应一个 LED 灯，高电平控制灯点亮

output reg sound：蜂鸣器输出，在 0、1 之间变换发出声音

### 3.3 分模块实现系统功能

#### 3.3.1 分频模块

每次按下按键后都会有 20ms 的抖动部分，为了能较准确地计算出 20ms 的时间，需要对原有的 20MHz 时钟进行分频，得到周期为 20ms 的时钟<sup>[10]</sup>。

该系统的给定时钟频率为 20MHz，一个周期为  $1 \div (20 \times 10^6) = 5 \times 10^{-7} \text{s}$ ，编写一个计数器，当计数达到  $(10 \times 10^{-3}) \div (5 \times 10^{-7}) = 2 \times 10^4$  个周期时，分频后的时钟即 clk\_s 变换一次高低电平，则得到周期为 20ms 时钟 clk\_s，在去抖模块里，用 clk\_s 的边沿触发，只要从按下按键 A 到按键有效之间的间隔大于 clk\_s 的一个周期，就能实现去抖。

分频器，的管脚图如图 3.2。其中 rst 为复位；clk 是输入时钟，频率 20MHz；clk\_s 是分频后的时钟，频率 50Hz。

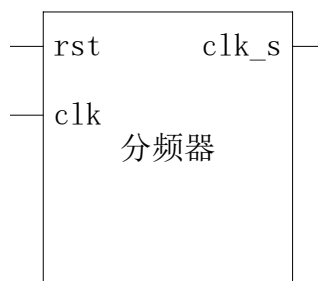


图 3.2 分频器引脚如

分频代码：

```
always@(posedge clk) begin
    if(!rst) begin//按下复位键后计数器清零
        count<=0;//计数器复位
        clk_s<=0;//时钟置初值
    end
    else if(count==20'H186A0) begin //设置计数值为十六进制的 186A0
        clk_s=~clk_s;//时钟输出
        count<=0;//计数器清零
    end
    else count<=count+20'H00001;//不够 clk_s 的一个波长时计数器加一
end
```

该模块采用 clk 上升沿触发，复位按键可对寄存器和分频时钟进行清零，当计数器计满  $2 \times 10^5$  即换成十六进制的 186A0 时，相当于分频后时钟的半周期。

对分频单独进行仿真，效果如图 3.3，结果显示 count 达到预设的数值时 clk\_s 跳转，并且 count 归零，说明分频满足设计要求。

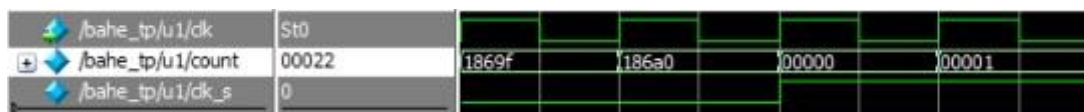


图 3.3 时钟分频波形仿真结果

### 3.3.2 按键处理模块

由于受按键本身限制，实际按下按键后会有约 20ms 的抖动时间，此期间，按键高低电平不确定，在计算按键按下次数是应该将这部分过滤掉，即只有当检测到超过 20ms 的时间里按键一直处于低电平输入状态，才算作有效按压，这样才能保证计数的准确性。抖动部分与实际有效部分如图 3.5。

因为系统要求比较两按键按压频率，所以即便可以准确检测到按键有效也还是远远不够的，解决方法是同时对按键释放也进行去抖并寄存，当系统同时检测到按键的有效按压和有效释放后，计数器加一或减一。

管脚信息如图 3.4。b 为按键输入，低电平有效，b1 为中间过渡变量，b2 低电平表示按键有效。这里只做出 b 的下降沿消抖，上升沿消抖类似，但输入 b 改为高电平有效。按键 a 原理相同。



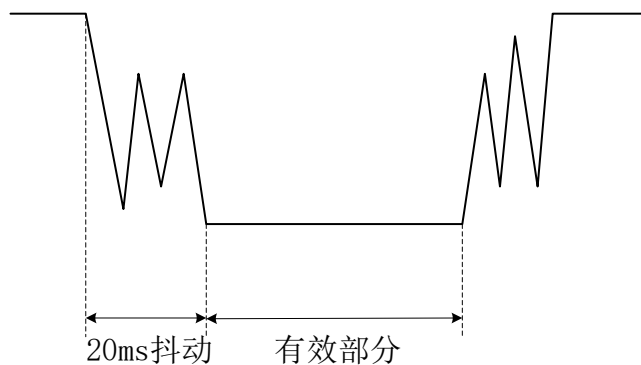


图 3.5 按键抖动原理图

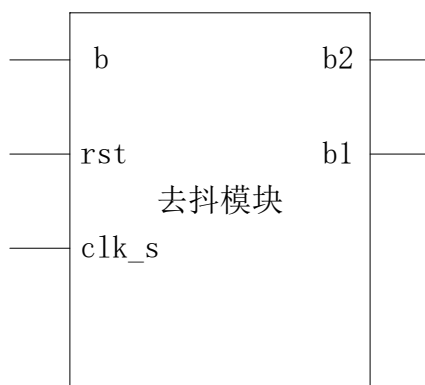


图 3.4 去抖模块管脚设置

去抖代码：

//b 下降沿消抖

always@(posedge clk\_s) begin//采用 clk\_s 触发，周期长为 20ms

if(!rst) begin//按键复位

b2<=0;

b1<=0;

end

else if(b==0) begin//检测到按键有动作时，启动

case(b1)//若 b1 未置位，则按键不输出；若 b1 已置位，按键输出

1:b1<=1;

0:b2<=1;

endcase

end

else begin//按键释放，寄存器自动复位

b2<=0;

```

        b1<=0;
    end
end

```

按键 b 按下后，在第一个 clk\_s 的上升沿时，该模块中 case 语句只对 b1 置位，在第二个 clk\_s 上升沿来临时，若 b 依然为低电平，case 语句才对 b2 置位，此时，表征 b 已按下。因为在两次 clk\_s 上升沿之间相差 20ms，所以 b2 的置位比 b 按下滞后 1~2 个周期，可以完成对按键的有效去抖。按键释放同理。

对按键 b 单独进行仿真，b2 和 b4 分别表征按键的按下有效和释放有效，可以看出均在 clk\_s 上升沿改变，且落后 b 至少一个周期，满足设计需求，实现去抖功能。去抖结果如图 3.6 所示。

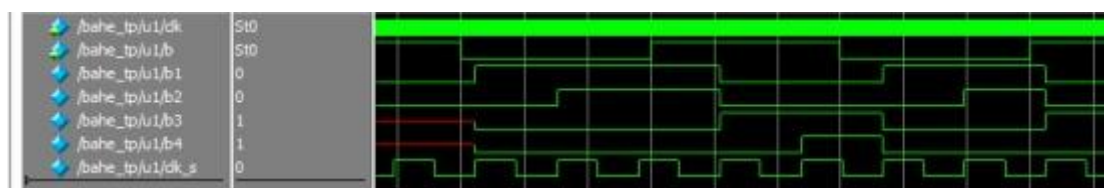


图 3.6 按键去抖波形图显示

### 3.3.3 按键频率比较模块

经去抖模块处理后，寄存 a2、a4、b2、b4 置位时分别表示按键 a 按下、按键 a 释放、按键 b 按下、按键 b 释放，在这一模块中，需要再加入两个寄存器 powera 与 powerb。

以按键 a 为例，当检测到 a2 高电平时置 powera 为一寄存按下信号，此时再检测到 a4 高电平时表明按键已松开，计按键 a 按压一次。powera 存在的意义是锁存按下信号，否则当按键松开时，按下信号将会清零。采用同时检测按键按下与释放是为了避免玩家按压按键后不松开，都是计数器重复计数，影响比赛公平性。

比较模块的引脚配置如图 3.7 所示。a2、a4、b2、b4 分别是两个按键的上下沿经去抖后的有效信号。三个输出，其中 powera 与 powerb 为标志位，gap 表征两个按键按下次数的差值，若 a 按键有效对 gap 加一，则 b 按键对 gap 减一。

按键比较代码：

//按键 a 与 b 按下频率的差值

```

always@(posedge clk) begin
    if(!(rst && cle)) begin gap<=4'b0101;end//按键差值复位
    else if(gap==4'b0000) begin gap<=4'b0000;end//差值到始端保持

```

```

else if(gap==4'b1011) begin gap<=4'b1011;end//差值到末端保持
else if(a2 ^^ b2) begin //一旦有按键按下，触发
    if(a2)
        powera<=1;//若是 a 按下，则置位 powera
    else
        powerb<=1;//若是 b 按下，则置位 powerb
end
else if(a4 && powera) begin //a 释放且标志位置位触发
    gap<=gap+4'b0001;//若 a 已释放，则按键差值加一；
    powera<=0;//同时标志位清零
end
else if(b4 && powerb) begin//b 释放且标志位置位触发
    gap<=gap-4'b0001;//若 b 已释放，则按键差值减一；
    powerb<=0;//同时标志位清零
end
end
end

```

以上是用 Verilog 编写的按键比较模块。由于语言风格限制，每一个变量只能在一个 always 块中被赋值，所以系统中只能在同一模块中先扫描按键 a，再扫描按键 b，然后判断对 gap 的加减。

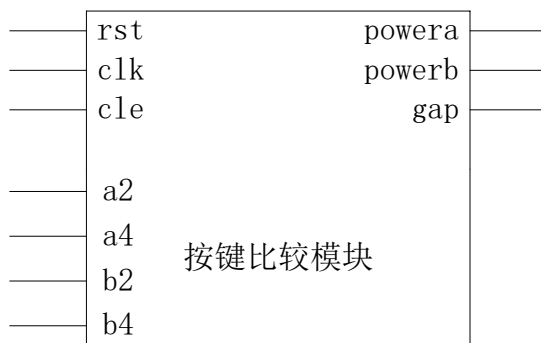


图 3.7 按键频率比较模块引脚设置

对于公平性，因为模块的触发方式为 20MHz 的时钟，远大于人手的按下速度，所以两次扫描之间的时差可以忽略不计。

为验证模块的正确性，对消抖模块与比较模块进行仿真，为了直观，并未演示寄存器的波形，但据图可以清楚地看出当 b 出现一次低电平时，gap 减一，当 a 出现一次低电平时，gap 加一，且 gap 值得变化较按键动作存在明显滞后。对比波形图如图 3.8 所示。



图 3.8 按键频率比较结果波形图展示

### 3.3.4 亮点控制模块

在对按键进行了处理后，按键比较情况用寄存器 `gap` 表示，为满足设计效果，以 `gap` 为输入，共分为三个输出部分：即亮点显示、分数显示、蜂鸣器报警。

亮点显示：设寄存变量为一个 7 位二进制数 `light`。当复位后，`light` 为 `B0001000`，高电平点亮 LED，即只有中间一个灯亮。每当 `gap` 增加或减少到一个阈值时，“1”的位置移动一位，即亮点也跟随变化。当移动到两端最后一位时，`light` 锁定，只有 `cle` 或 `rst` 低电平时才复位。

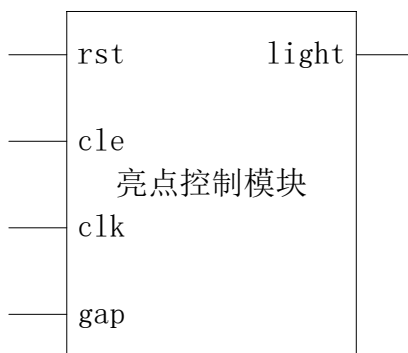


图 3.9 亮点控制模块引脚配置

亮点控制模块的管脚设置如图 3.9 所示。主要输入为 `gap`，代表了按键的及时信息；输出为 `light`，共 7 位，每一位控制一个 LED 灯，高电平点亮。

亮点移动控制模块代码：

//亮点移动的控制模块

```
always@(posedge clk,negedge rst,negedge cle) begin
    if(!rst) begin                //复位，中间灯点亮
        light<=7'b0001000;
    end
    else if(!cle) begin
        light<=7'b0001000;
    end
    else if(gap==4'b0000) begin    //玩家 b 胜出
```

```

    light<=7'b00000001; //移动到最左端时锁存
end

else if((4'b0000<gap) && (gap<=4'b0010))    //根据赛况亮点移位
    light<=7'b0000010; //第一盏灯亮
else if((4'b0010<gap) && (gap<=4'b0100))    //第二盏灯亮
    light<=7'b0000100;
else if((4'b0100<gap) && (gap<=4'b0110))    //中间灯亮，表明平局
    light<=7'b0001000;
else if((4'b0110<gap) && (gap<=4'b1000))    //第五盏灯亮
    light<=7'b0010000;
else if((4'b1000<gap) && (gap<=4'b1010))    //第六盏灯亮
    light<=7'b0100000;
else if(gap==4'b1011) begin    //玩家 a 胜出
    light<=7'b1000000; //移到最右端时锁存
end

end

end

```

以上是用 Verilog 编写的亮点控制模块，对于每一个 gap 值，都有相对应的灯点亮。

在 Modelsim 中对已编好模块进行仿真如图 3.10 所示，可以看出，随着 gap 值得变化，light 发生移位，最终“1”停止在最末端。这里只给出按键 b 的模拟，按键 a 同理。

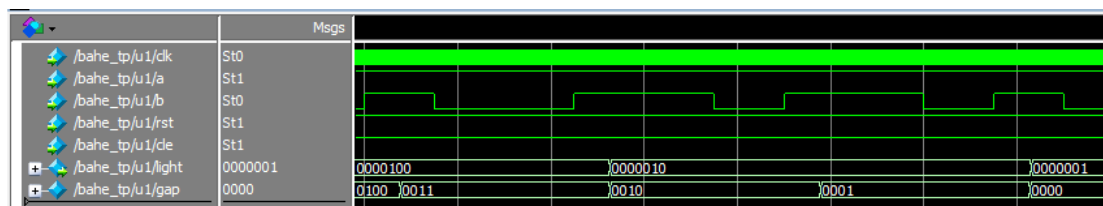


图 3.10 亮点移动波形图

### 3.3.5 蜂鸣器控制模块

当分出胜负后，启动蜂鸣器进行报警。当控制蜂鸣器的寄存器在 0 与 1 之间波动时，蜂鸣器发出响声，且波动频率越高，蜂鸣器音调也就越高<sup>[11]</sup>。为简便起见，该系统使用去抖时用到的时钟，即震动频率 50Hz。

蜂鸣器电路的管脚设置如图 3.11 所示。与亮点控制模块相比，特殊的有 clk\_s 输入，用于对振动频率的控制。输出为一位寄存器 sound，接蜂鸣器电路。

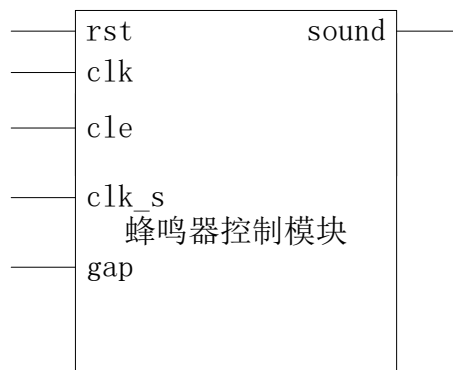


图 3.11 蜂鸣器控制模块引脚配置

蜂鸣器控制部分代码：

//蜂鸣器输出控制模块

```
always@(posedge clk) begin
    if((!rst)^(!cle)) sound<=0;
else if((gap==4'b0000)^(gap==4'b1011))//当决出胜负时触发
    sound<=clk_s;//将 50Hz 的时钟赋值给 sound 输出
end
```

该模块程序较为简单，当决出胜负时启动蜂鸣器。

在 Modelsim 中对蜂鸣器进行仿真结果如图 3.12 所示。可知，sound 初始值为 0，蜂鸣器无动作，如果一直按 b 键，最后决出胜负时，sound 与 clk\_s 波形保持一致，蜂鸣器此时有响声。

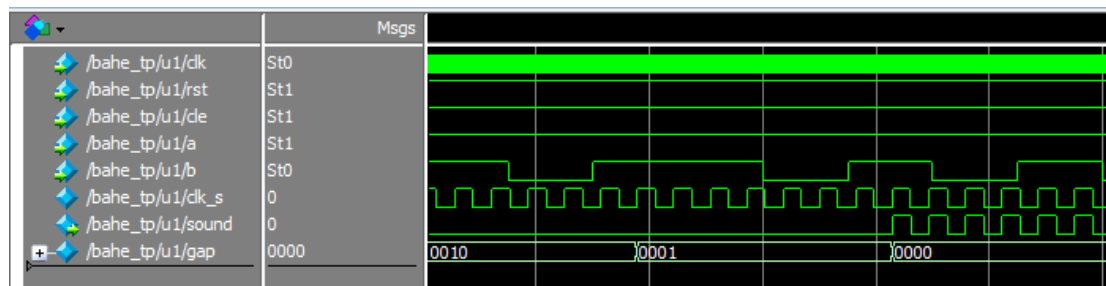


图 3.12 蜂鸣器控制波形图

### 3.3.6 分数显示模块

本系统采用两个数码管显示比分，由 score\_1 和 score\_2 控制，分别代表两个玩家。当亮点移动到末端时，根据玩家的胜负决定哪个数码管加一。为避免重复加分，在这里需要设置一个使能寄存器 score\_per，该寄存器 a、b 按键共用，在亮点移动到末端的前一个周期就先对该寄存器置 1，分数变化同时对寄存器清

零，这样就做到了只加一次分。

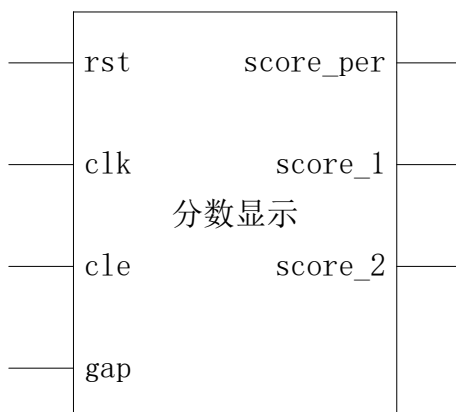


图 3.13 分数显示模块的引脚

做出分数显示模块的管脚设置如图 3.13 所示。主要通过 `gap` 的输入来控制输出；`rst`、`cle`、`clk` 分别为复位与时钟；`score_per` 为分数跳转的使能寄存器；`score_1`、`score_2` 均为四位二进制输出，分别代表 a、b 两个玩家的比分。

分数显示的代码：

//分数输出控制

```
always@(posedge clk,negedge rst,negedge cle) begin//clk、rst、cle均可触发
    if(!rst) begin //复位，分数清零
        score_1<=0;
        score_2<=0;
    end
    else if(!cle) //cle 复位，只清除寄存器不清比分
        score_per<=0;
    else if((gap==4'b0001)^(gap==4'b1010)) //在决出胜负之前，置位使能寄存
        score_per<=1;
    else if(gap==4'b0000 && score_per) begin//若 a 胜出且使能位为 1
        score_1<=score_1+4'b0001;//则 a 比分加一
        score_per<=0;//清除使能位
    end
    else if(gap==4'b1011 && score_per) begin //若 b 胜出且使能位 1
        score_2<=score_2+4'b0001;//则 b 比分加一
        score_per<=0;//清除使能位
    end
end
```

end

以上为控制数码管输出的代码，在复位部分，如果 `cle` 低电平则只对寄存器清零，表明一局结束；只有当 `rst` 低电平时才清零比分，表明系统复位；通过 `a`、`b` 按键的差值 `gap` 辨识是哪一方胜出，并给胜出的一方加分。

在对按键 `b` 一直按压后 `gap` 的值一直减小，在 `gap` 减为 `0001` 时，使能 `score_per`，在下次按下 `b` 按键，`score_1` 加一，即玩家 `b` 得一分，同时使能 `score_per` 复位。在 Modelsim 中分数显示的波形如图 3.14。

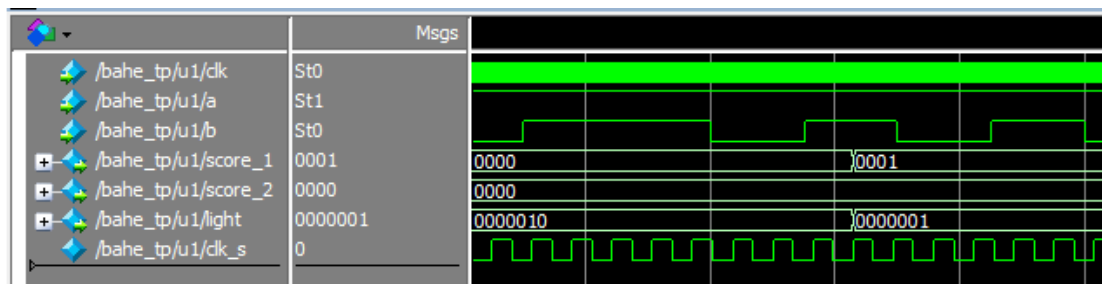


图 3.14 分数显示模块分数展示

## 3.4 系统仿真

### 3.4.1 Modelsim 软件介绍

Modelsim是一款针对于FPGA芯片的仿真软件，被广泛应用于硬件描述语言仿真。在课内也曾对该软件有所接触，功能的可靠与便捷的交互界面为开发人员提供了比较全面的开发环境，无论从功能还是速度上都是硬件编程领域优秀的仿真软件<sup>[12]</sup>。

在这次设计中用到的Verilog语言可以很好的运行在Modelsim中，该软件还提供了编译、纠错、波形图仿真、电路生成等一系列功能，是基于FPGA设计拔河游戏机最好的选择。

### 3.4.2 在 Modelsim 中完成软件仿真

在 Modelsim 中完成软件仿真，步骤为：

#### （1）编写程序并编译

在 Modelsim 中新建一个 library，并命名为 `work`，之后的程序都是在这里面进行建立与调试。依次点击 `file—new—project` 在建好的 library 中建立工程。弹出对话框，在 `project` 中输入拔河游戏机的工程名字 `bahe`，点击 `OK`。

在新建立的工程中，需要更改 `type` 为 `Verilog`，否则软件默认为 `VHDL`，编译时会出现错误。



按照同样的方法再建立一个 **testbench**，之后的仿真代码都将写入这个工程，以便于在 **simulate** 中进行波形图仿真。

双击所建立的工程，在新出现的界面上输入拔河游戏机的程序。保存之后进行编译，软件提示语法上的错误与警告，双击提示回到程序中修改。

为确保程序运行无误，需要通过相同的方法建立一个新工程，命名为 **bahe\_tp**。在该程序里，通过调用主程序并变换输入值来验证程序是否编写正确。

## （2）软件仿真

确保程序语法无误后，在 **library** 选项卡里右击 **testbench** 程序，这里的工程名是 **bahe\_tp**，点击 **Simlate** 进行仿真。运用 **Simulate** 仿真如图 3.15 所示。

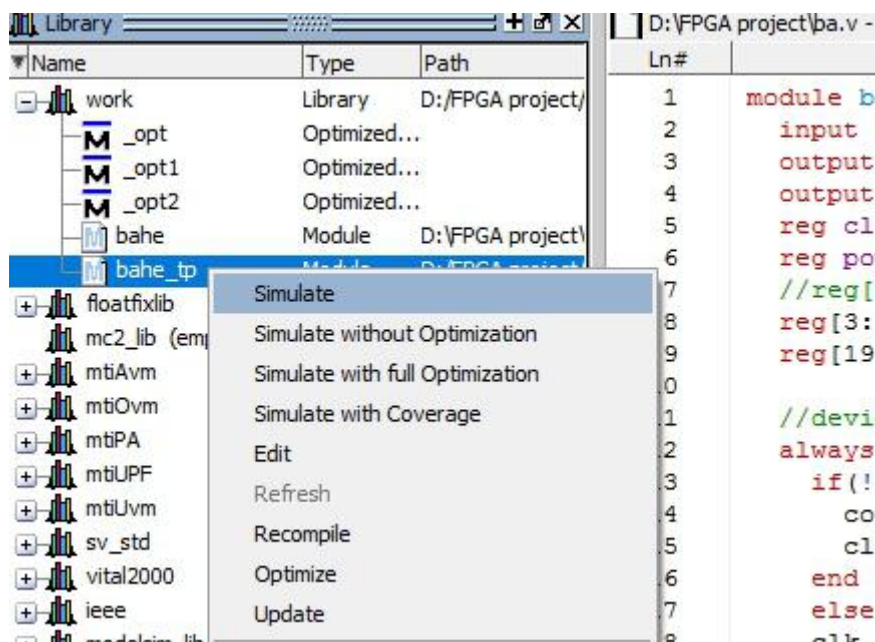



图 3.15 通过 **testbench** 对程序进行仿真

将所有的输入输出引脚与寄存器变量加入到 **wave** 界面，点击  运行仿真，可以直观的看到包括时钟、按键、寄存器在内的运行情况。仿真波形如图 3.16 所示。在波形图中，可以看到已给时钟频率太大，仿真时为简便可以换用相对较小频率的时钟，在进行程序烧制时再更改参数与实际时钟相频率匹配。

在 **Modelsim** 中显示包括所有输入输出、内部寄存器的波形变化，确保软件部分准确无误。



图 3.16 软件仿真结果

## 第四章 硬件实现

### 4.1 Quartus II 软件介绍

为了将程序下载到开发板完成硬件实现，本毕业设计需要使用开发软件 Quartus II，由 Altera 公司开发，主要用于 Altera 公司的各种 FPGA 和新器件开发，支持 Verilog、VHDL 等多种输入方式，整个 PLD 的设计过程都可以用 Quartus II 完成<sup>[13]</sup>。

与 Modelsim 类似，Quartus II 也可以实现 Verilog 程序的编译仿真、波形图分析、电路的布线等功能。除此之外，Quartus II 还可以实现管脚的配置，程序的下载。

### 4.2 程序下载与设计完成

将拔河游戏机的设计程序下载到 FPGA 开发板的流程如下：

#### （1）输入程序

建好工程后，弹出对话框，该拔河游戏机主要基于 Verilog 编程，所以这里选择 Verilog HDL File，为保证程序能被正确读出，需设置保存名和程序名一致为 bahe，下一行工程名将会自动补充。

#### （2）选择芯片

先在 family 中选择芯片类别为 CycloneII，再在 Available device 中选择具体用到的 EP2C8Q 型号。芯片选择方法如图 4.1 所示。

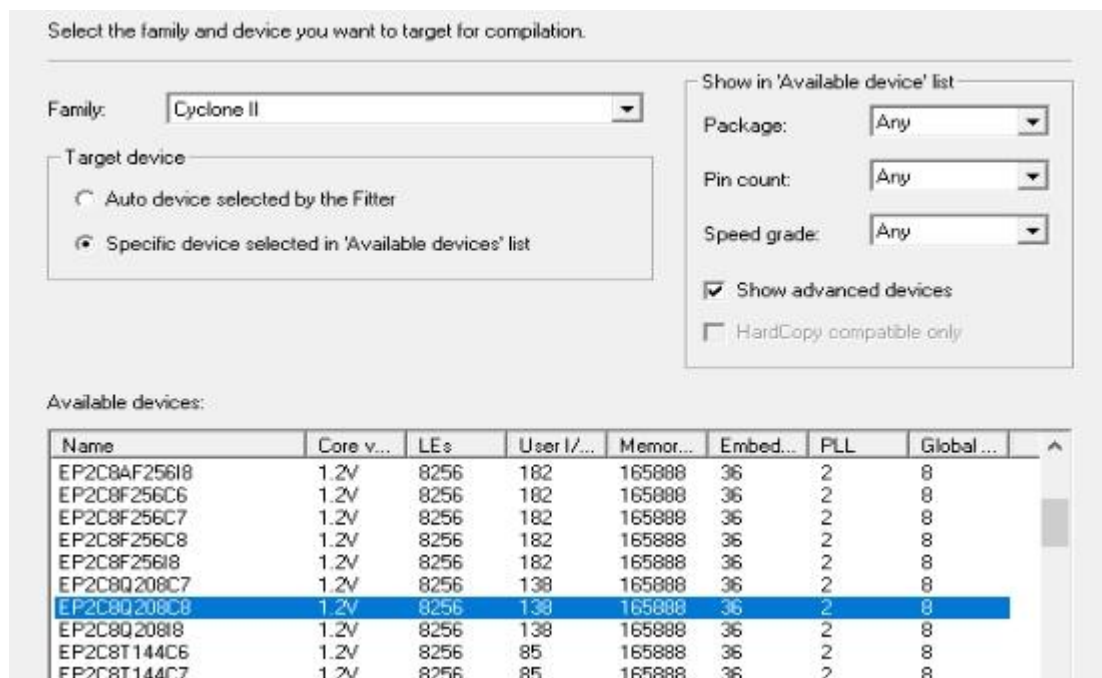




图 4.1 选择芯片类型

### （3）引脚配置

为了所设置的输入输出端口与开发板上的各个器件能一一对应，需要在 quartusII 中完成对管脚的配置，首先在菜单中选择 `processing--start--start analysis&synthesis` 或在工具栏中直接点击  生成管脚信息。之后选择 `assignment--pin planner` 或在工具栏中点快捷键弹出配置界面，界面如图 4.2 所示，在弹出的对话框中输入端口对应的管脚。

最后选择 `processing--start compilation` 或在工具栏点击  进行编译生成报告，由于之前已进行过仿真，这里没有错误显示。

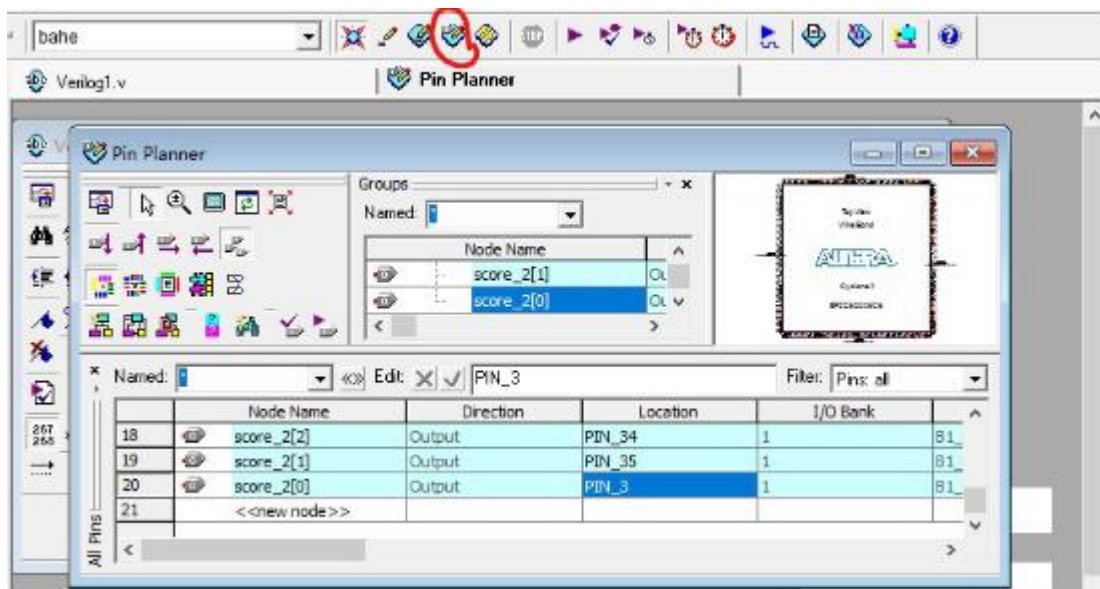


图 4.2 管脚信息生成与配置

（4）硬件实现，点击 `programmer` 快捷键，如图 4.3①处所示，弹出下载程序的对话框，给开发板插上电源和下载线，点击 `hardware setup`，如图 4.3②处所示，在弹出的对话框中选择 `USB-0` 作为下载程序的接口。

对于所添加的文件，如果当前使用的是 `JTAG` 接口，则选择 `sof` 文件，这种情况下程序被烤进 `RAM`，如果开发板断电，则程序消失；如果使用 `AS` 接口，则选择 `pof` 文件，这种情况下程序被烤进 `EPCS` 中，每次开发板上电，程序由 `EPCS` 烤进 `FPGA` 中，省去了每次都要重新配置管脚的麻烦。





图 4.3 添加硬件与烧制程序

(5) 点击 start 下载程序，待进程到达 100%后，表明程序烧好，此时可以再开发板上直接进行拔河游戏机的模拟。经检验，无误。硬件实物如图 4.4 所示。

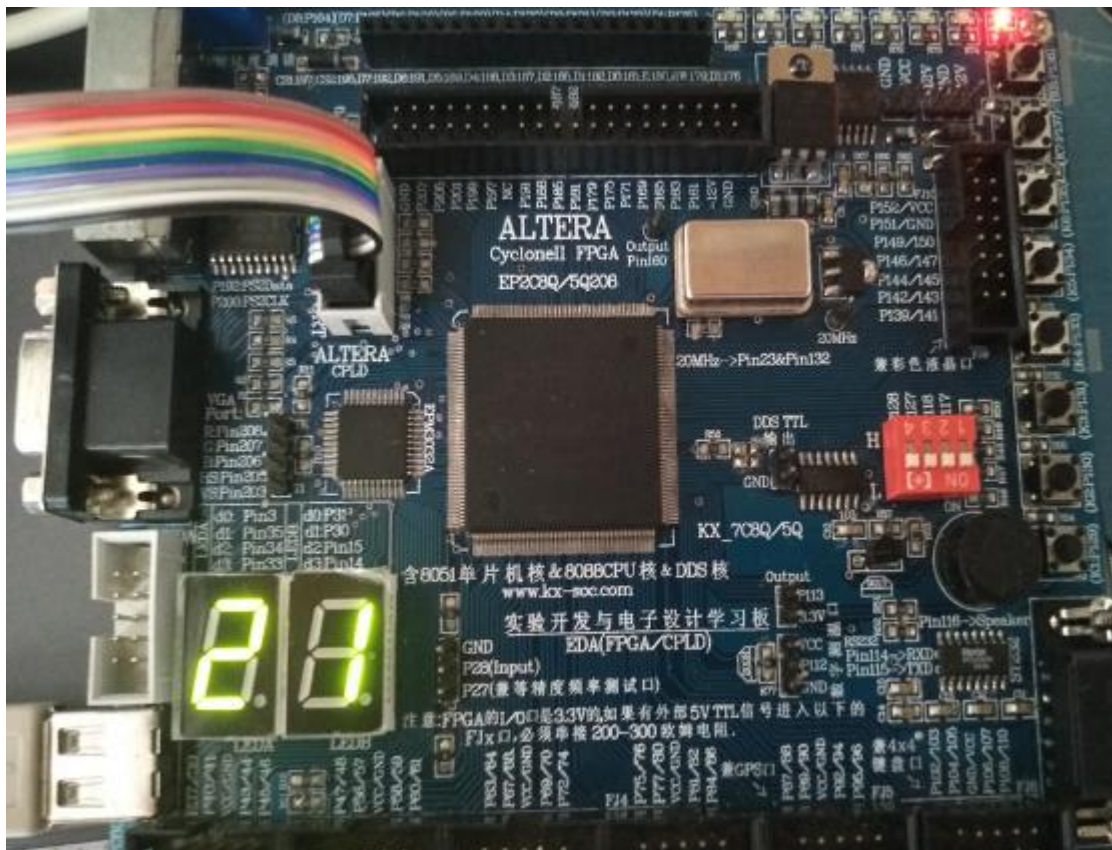


图 4.4 硬件实物图

## 第五章 结论

### 5.1 总结：

本毕业设计基于 FPGA 设计出一个拔河游戏机，设计过程中要注意比赛的公平性，避免因设计的不合理导致比赛结果偏差，才能将基于 FPGA 的拔河游戏机应用在实际中。对于该毕业设计，利用 FPGA 多线程处理模式克服了纯硬件数字系统方式控制不灵活的缺点，本设计采用 altera 公司的可编程逻辑器件，并配以开发软件在计算机上进行各种电路设计和修改，并对电路特性进行仿真模拟，最后将设计方案下载到开发板中，这样可实现高度集成和精确的电路设计和降低设计成本提高了设计效率和电路的可靠性。

在设计中所设计的拔河游戏机也存在许多不足，如比赛时间太短，显示内容不够丰富等等。

### 5.2 展望：

本课题成功完成了题目要求的内容，但还是有许多不足。在按键输入方面，可以外接按键，将 A、B 玩家的操纵按键距离拉大，这样更便于在实际中双人操作；在显示部分，可以外接 LED 灯，这样在外观上更能突出比赛的胜负。本设计的比赛时长也较短，可以通过改变计数器部分的参数并增加灯的数量来增加比赛时间；此外，还可以设置成多人拔河游戏，可以更好地体现基于电子系统拔河游戏机的优越性，编程难度也自然会更高。

## 致 谢

经过设计和仿真，到这次的毕业设计顺利完成，离不开导师的帮助与同学的鼓励，还有一只关心支持我的家人。我要在这里表达对每个人深深的谢意！

我要诚挚的感谢我的导师——姚霁老师，姚老师学识渊博，平易近人，本毕设的选题和撰写都是在姚老师的指导下完成的，从开始不会使用仿真的情况下，慢慢学习，通过一段时间的努力，开始在开发板上尝试操作，最终结果正确。老师对于解决问题的方法，都将成为我今后学习与工作的宝贵财富。在我走向工作岗位以后，老师给予我的帮助与教诲将促使我更积极的追求新的目标。

此外，我还将感谢与我一同毕业的同窗；感谢陪伴我四年的舍友；感谢虽然没提及姓名却也给予过我帮助的老师与同学！

## 参考文献

- [1]王金明.数字系统设计与 Verilog HDL—5 版[M].北京:电子工业出版社,2014.7
- [2]李洪国,沈明山.可编程器件 EDA 技术与实践[M].北京:机械工业出版社 2003
- [3]甘历. VHDL 应用与开发实践[M]. 北京: 科学出版社, 2003.
- [4]郑诗程, 曹小虎, 朱虹, 周谦之.基于 FPGA 的单相交交变频系统倍频同步的研究与实现[R].中国电源学会全国电源技术年会, 2009
- [5]孟志华.利用 FPGA 实现 UART 的设计[D].《信息化研究》,2007,33(4):14-16
- [6]徐志军. CPLD/FPGA 的开发与应用[M]. 北京: 电子工业出版社, 2002.
- [7]崔建明. 电工电子 EDA 仿真技术[M]. 北京: 高等教育出版社, 2004.
- [8]侯继红, 李向东. EDA 实用技术教程[M]. 北京: 中国电力出版社, 2004.
- [9]侯伯亨. VHDL 硬件描述语言与数字逻辑电路设计[M]. 西安: 西安电子科技大学出版社, 1997.
- [10]曾繁泰, 陈美金. VHDL 程序设计[M]. 北京: 清华大学出版社, 2001.
- [11]Mark Zwolinski. VHDL Digital Systems Design[M].BeijingPublishing House of Electronics Industry, 2002
- [12]Serigio Franco.Design with Operational Amplifiers and Analog Integrated Circuits[J]. Xi'an: Xi'an Jiaotong University Press,2004.8
- [13]FLEX 10K Embedded Programmable Logic Device Family Data Sheet[S].2003,104~109



## 附录 1

//testbench 测试程序

`timescale 1ns/1ns//设置仿真最小时间 1ns，精确度 1ns

module bahe\_tp;//工程名为bahe\_testbench的缩写

reg clk,a,b,rst,cle;//将运行程序里的输入定为reg型

wire[3:0] score\_1,score\_2;//输出定义为wire型

wire[6:0] light;

integer i,j;

bahe u1(clk,a,b,rst,cle,score\_1,score\_2,light);//调用运行程序，并取名u1

always #1 clk=~clk;//设置输入时钟周期为 2ns

initial begin

clk=0;a=1;b=1;rst=1;cle=1;//在initial中设置初始值

#2 rst=0;//用rst低电平复位

#2 rst=1;

end

always begin//每过 100ns按键a变一次电平

#100 a=0;//模拟不断按下按键a

#100 a=1;

#100 a=0;

#100 a=1;

#100 a=0;

#100 a=1;

#100 a=0;

#100 a=1;

#100 a=0;

#100 a=1;

end

endmodule