

# 北京邮电大学

## 数字电路逻辑设计课程（论文）



题目： 基于 ZyBo 的简易弹球以及 VGA 显示输出设计

姓 名 \_\_\_\_\_

学 院 \_\_\_\_\_ 理学院

专 业 \_\_\_\_\_ 应用物理

班 级 \_\_\_\_\_

学 号 \_\_\_\_\_

班内序号 \_\_\_\_\_

指导教师 \_\_\_\_\_

2014 年 6 月

# 基于 ZyBo 的简易弹球以及 VGA 显示输出设计

## 摘 要

现场可编程门阵列被广泛应用于定制芯片的设计、集成电路设计与学习等方面。提出了基于赛灵思的 ZyBo 开发板与若贝软件联合设计一个简易弹球与显示输出的实例的方案。研究了使用知识产权核、显示器扫场分频、硬件描述语言编写等方面的内容。实验流程为数字电路综合实验的开展提供了指导。

**关键词** 数字电路    若贝软件    现场可编程门阵列    硬件描述语言

# **A Concise Pinball Game with VGA Display Output Based on ZyBo**

## **ABSTRACT**

Field programmable gate array (FPGA) is widely used in the design of integrated circuit and the study of the customization of chips, etc. A practical example of a concise pinball game with VGA display output based on ZyBo is proposed. Intellectual property core, field sweeping method of display and the hardware description language (HDL) are been discussed. The process provides guidance for the digital circuit experiments.

**KEY WORDS**    Digital circuits    Robei software    FPGA    HDL

# 目 录

第一章 绪论.....	1
1.1 研究背景与意义.....	1
1.1.1 FPGA 原理简述 .....	1
1.1.2 FPGA 应用场景阐释 .....	1
1.1.3 相关名词解释.....	1
1.2 本文研究主要内容.....	2
1.3 本文组织结构.....	2
第二章 系统设计.....	3
2.1 功能及需求.....	3
2.2 体系结构设计.....	3
2.3 平台框架相关技术.....	3
2.3.1 利用 FPGA 进行 VGA 的驱动.....	3
2.3.2 利用计数器进行频率分配.....	5
2.3.3 调用 IP 核定制用户时钟.....	5
2.3.4 通过 Verilog HDL 实现简易弹球游戏.....	6
2.3.5 对于 Zynq Board 自带的 ARM Cortex™-A9 芯片的探究.....	6
2.4 详细设计.....	7
2.4.1 Robei 软件的使用综述 .....	7
2.4.2 利用 Robei 软件进行模块构建以及波形仿真.....	7
2.4.3 Vivado 工程的建立 .....	9
2.4.4 编写相关约束文件、综合以及比特流生成.....	9
第三章 功能实现.....	11
3.1 实例验证前的准备.....	11
3.2 实例验证具体操作过程.....	11
3.2.1 开发板连接与比特流下载.....	11
3.2.2 功能验证与测试.....	11
3.2.3 问题与可能的原因.....	12
第四章 总结与展望.....	13
4.1 总结与感想.....	13
4.2 对未来的展望.....	13
参考文献.....	14
致 谢.....	15

## 第一章 绪论

### 1.1 研究背景与意义

随着技术发展,传统的分立元件以及由晶体管构成的逻辑门电路(Transistor-Transistor Logic, TTL)正在逐渐淡出现代电子工业,取而代之的是大规模以及超大规模集成电路的设计。从上世纪八十年代以来,赛灵思®公司推出的现场可编程门阵列(Field-Programmable Gate Array, FPGA)得到广泛的应用,其灵活性以及可定制能力使其逐渐脱颖而出,让人们看到了未来更广泛的芯片定制的可能性。从技术上来说,FPGA<sup>[1]</sup>最初只是逻辑器件,而现在更强调平台概念,加入数字信号处理、嵌入式处理、高速串行和其他技术,从而被应用到更多的领域。

#### 1.1.1 FPGA 原理简述

FPGA<sup>[1]</sup>利用小型查找表( $16 \times 1$ RAM)来实现组合逻辑,每个查找表连接到一个 D 触发器的输入端,触发器再来驱动其他逻辑电路或驱动 I/O,由此构成了既可实现组合逻辑功能又可实现时序逻辑功能的基本逻辑单元模块,这些模块间利用金属连线互相连接或连接到 I/O 模块。FPGA 的逻辑是通过向内部静态存储单元加载编程数据来实现的,存储在存储器单元中的值决定了逻辑单元的逻辑功能以及各模块之间或模块与 I/O 间的联接方式,并最终决定了 FPGA 所能实现的功能,FPGA 允许无限次的编程。简单来说,FPGA 通过一些基本逻辑模块的组合实现真值表的逻辑,并将其组合实现大规模逻辑。

#### 1.1.2 FPGA 应用场景阐释

- 通信设备的高速接口电路设计:这类应用通常要求采用具备高速收发接口的 FPGA,同时要求设计者懂得高速接口电路设计和高速数字电路板级设计,具备 EMC/EMI 设计知识,以及较好的模拟电路基础,需要解决在高速收发过程中产生的信号完整性问题。
- 数字信号处理等产品设计:这类设计主要集中于将 FPGA 技术应用于实际场景中,诸如数字滤波器的设计、视频信号采样及处理,乃至更高级的数学计算如比特币算法挖掘等,都可找到其应用。
- 嵌入式系统与综合:类似于 ARM 但有所不同,FPGA 可为嵌入式软件开发提供一个底层硬件平台。在这个平台上开发者可以通过对端口驱动的重新定义实现更为定制化的功能。同时,FPGA 与 ARM 结合也是一种潮流,通过软硬结合的开发方式使资源最大化利用。

#### 1.1.3 相关名词解释

FPGA: 现场可编程门阵列, Field-Programmable Gate Array

Verilog HDL: Verilog Hardware Description Language, 以文本形式来描述数字系统硬件的结构和行为的语言。

**Robei<sup>®</sup>**: Robei<sup>[3]</sup>是一款可视化的 EDA 设计工具。是一款为芯片设计工程师量身定做的专用工具。

**Xilinx<sup>®</sup>**: Xilinx 赛灵思公司是全球领先的可编程逻辑完整解决方案的供应商。

**ARM<sup>®</sup>**: ARM(Advanced RISC Machines)是微处理器行业的一家知名企业, 设计了大量高性能、廉价、耗能低的 RISC 处理器、相关技术及软件。

**IP core**: 知识产权核(Intellectual Property Core), 是指某一方提供的、形式为逻辑单元、芯片设计的可重用模块。

**ZyBo**: Zynq Board, 是赛灵思公司(Xilinx)推出的行业第一个可扩展处理平台。

**VGA**: Video Graphics Array, 是 IBM 在 1987 年随 PS/2 机一起推出的一种视频传输标准。不支持热插拔。

**GUI**: 图形用户界面(Graphical User Interface)。

## 1.2 本文研究主要内容

本文将通过一个基于 Zynq Board 的简易 FPGA 弹球与 VGA 输出实例来介绍通过 Robei 软件与 Vivado<sup>®</sup>设计套件进行联合开发的流程。通过对本实例的学习, 开发者将能更快地熟悉联合开发的流程。

## 1.3 本文组织结构

在前面的绪论中, 本文主要介绍 FPGA 等研究背景, 以及本文的主要研究内容。在随后的章节中, 将详细介绍以下内容:

第二章, 系统设计及架构。本章将介绍本文的主要技术以及详细设计。

第三章, 功能实现。本章将介绍程序在 ZyBo 上测试的流程。

第四章, 总结与展望。本章将说明本实例的不足与改进方向。

最后, 简短地对本文所做的工作做了总结, 并指明了未来的研究重点。

## 第二章 系统设计

本章将介绍基于 ZyBo 的简易弹球以及 VGA 显示输出设计实例的设计流程，并将依据功能及需求分析、体系结构设计、相关技术与详细设计等四个方面来进行介绍。其中，在详细设计中，将会对通过 Robei 若贝 EDA 软件进行模块规划、硬件描述语言编写、波形仿真，通过 Vivado 套件应用 IP 核、进行综合与实施，通过 ZyBo 实现等方面进行详细介绍。

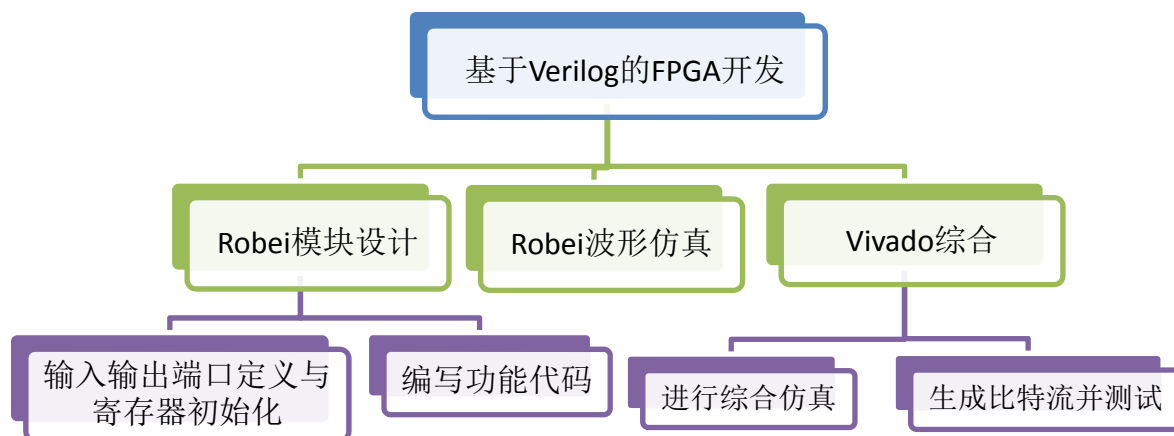
### 2.1 功能及需求

本实例的主要功能为：输出视频信号并构成简易弹球游戏，通过拨码开关和按键改变图像样式与位置等。

本实例需要满足的需求为：开发者通过对本实例的学习及模仿，能够初步了解 Robei 与 Vivado 联合设计的基本流程，加深对 FPGA 与数字电路课程的理解。

### 2.2 体系结构设计

本实例体系结构如下所示。



### 2.3 平台框架相关技术

本设计主要基于下列几项技术，本节将会详细介绍实验的框架设计。

#### 2.3.1 利用 FPGA 进行 VGA 的驱动

在 FPGA 设计中，可以通过开发者自定义时钟的方式产生 VGA 视频输出信号。通过查阅视频电子标准协会(VESA)所给出的相关标准，本实例采用标准的 VGA 输出，即  $640 \times 480 @ 60\text{Hz}$  的信号。实际操作中，为了兼顾资源占用与显示效果，本实例进行了一些修改。

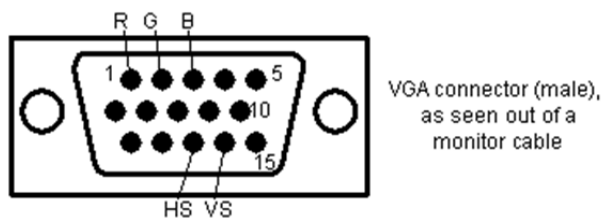


图 2-1 VGA 接口引脚定义

查阅资料后我们发现，VGA 的十五针引脚中，本实例中只需用五个引脚，其分别为 R、G、B 三种颜色的引脚以及水平同步 Hsync 以及垂直同步 Vsync 引脚。具体引脚排列与 ZyBo 板上端口定义见图 2-1、图 2-2。

本实例中，由于并不需要复杂及高精度的 16 位色彩深度来进行图形显示，因而略去对于 RGB 色彩显示原理的探讨。下面主要介绍对于扫场同步信号所需时钟的产生。

显示器采用光栅扫描方式，即轰击荧光屏的电子束在 CRT 屏幕上从左到右（受水平同步信号 Hsync 控制）、从上到下（受垂直同步信号 Vsync 控制）做有规律的移动。光栅扫描又分逐行扫描和隔行扫描。电子束采用光栅扫描方式，从屏幕左上角一点开始，向右逐点进行扫描，形成一条水平线；到达最右端后，又回到下一条水平线的左端，重复上面的过程；当电子束完成右下角一点的扫描后，形成一帧。此后，电子束又回到左上方起点，开始下一帧的扫描。这种方法也就是常说的逐行扫描显示。<sup>[5]</sup> 完成一行扫描所需时间称为水平扫描时间，其倒数称为行频率；完成一帧（整屏）扫描所需的时间称为垂直扫描时间，其倒数为垂直扫描频率，又称刷新频率，即刷新一屏的频率。

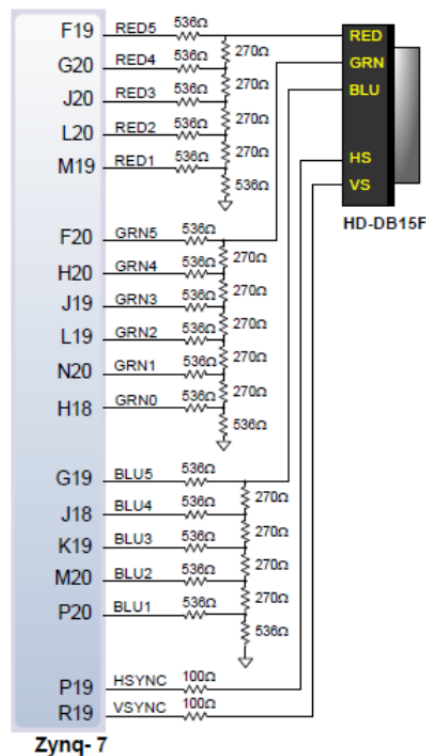


图 2-2 VGA 接口引脚定义

中文名称	Parameter	Vertical Sync			Horiz. Sync	
		Time	Clocks	Lines	Time	Clks
帧长	Sync pulse	16.7ms	416,800	521	32 us	800
有效时间	Display time	15.36ms	384,000	480	25.6 us	640
同步脉冲	Pulse width	64 us	1,600	2	3.84 us	96
前沿	Front porch	320 us	8,000	10	640 ns	16
后沿	Back porch	928 us	23,200	29	1.92 us	48

表 2-1 像素时钟脉冲为 25MHz 的 VGA 视频信号 640×480@60Hz 相关参数<sup>[4]</sup>

通过查阅资料，我们可以设计对应的时钟信号。由于 ZyBo 不具有原生的 25MHz 时钟频率，因而需要进行用户定制，这些内容将在 2.3.3 节进行介绍。



为了节省开发板资源，本实例中将水平与垂直同步帧长分别调整为 768 与 512 帧，这时水平同步频率为 32.8KHz 而垂直同步频率为 64Hz。由于该频率不符合标准的 VGA 视频信号，在少数情况下有可能产生兼容性问题。然而，经过多次测试，所有显示器均可以正常显示，故采用此参数。

该部分的 Verilog 具体实现如下：

```
wire CounterXmaxed = (CounterX==767);

always @(posedge clk)

if(CounterXmaxed)

    CounterX <= 0;    //置零

else

    CounterX <= CounterX + 1;

always @(posedge clk)

if(CounterXmaxed) CounterY <= CounterY + 1;
```

### 2.3.2 利用计数器进行频率分配

由于本例中仍需要一频率较低的时钟边沿作为按键的触发边沿，而在实际体验中该信号的占空比并不会影响按键输入信号的触发，因而考虑采用计数器方式产生此低频信号。根据计算，当时钟信号为 25MHz 时，需采用一模值为  $4 \times 10^5$  的计数器产生周期为 0.16s 的周期上升沿。

该部分的 Verilog 具体实现如下：

```
reg [18:0] count;

wire countmax = (count==400000); //产生时延防止按键按下变换过快
```

### 2.3.3 调用 IP 核定制用户时钟

查阅资料<sup>[4]</sup>得知，ZyBo 开发板提供一频率为 125MHz 的外部参考时钟。由于本实例中需要时钟频率为 25MHz，显然奇数分频通过编程实现将占用较多资源，因此采用板上自带的 IP 工具 Clocking Wizard 进行频率生成。该界面可在 Vivado 左侧导航窗格的 IP Catalog 选项中找到。将输入设为 PLL、并选择所需的引脚类型以及数目、设置好输入输出频率，系统将自动生成一个用于分频的 IP 文件。

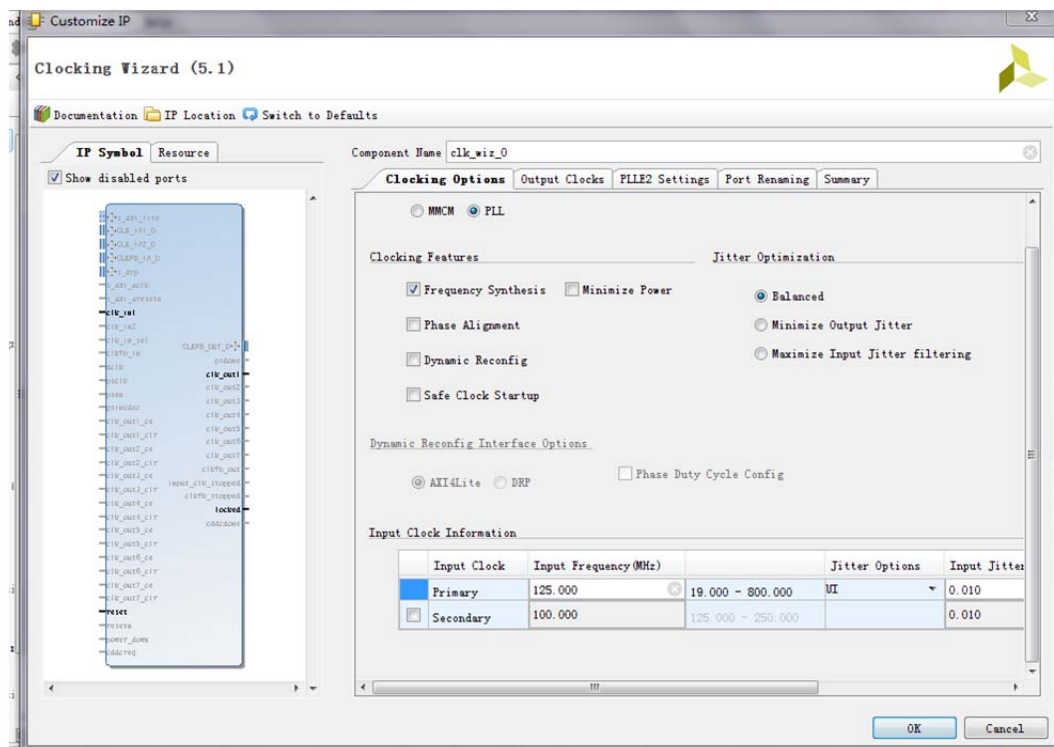


图 2-3 Clocking Wizard 生成 IP 文件界面

生成对应代码后，开发者便可以调用对应的变量，从而获得定制后的时钟频率。

### 2.3.4 通过 Verilog HDL 实现简易弹球游戏

本实例仅提供该弹球游戏的模式简述，具体实现请参阅附件中的代码。

弹球游戏主要涉及以下几方面需求：

- 固定物体的生成（墙、砖块等），自动移动物体的生成（弹球），用户可控移动物体的生成（挡板）。
- 球与墙、球与板、球与砖块碰撞的判断以及碰撞后方向的改变。
- 相应修饰等。

对应各个模块，附件中的源代码都进行了简要注释，方便开发者理解代码用途。

### 2.3.5 对于 Zynq Board 自带的 ARM Cortex™-A9 芯片的探究

ZyBo 的另一大特性便是其对 FPGA 与 ARM 的结合。开发者不仅可以通过硬件描述语言进行纯硬件的开发，亦可以用具有丰富资源与强大性能的 ARM 芯片进行常规的 C 语言、Java 语言等软件开发。二者结合将会创造出更多新的功能，期待开发者的探索。



图 2-4 Linaro Demo 图形界面下的 Ubuntu 14.04

ZyBo 出货前，在板载 QSPI Flash 中预装了 Open Linux(无 GUI 显示)。另外，在德致伦(Digilent)公司的官网上提供了 Linaro Demo(有 GUI 显示)的下载，初级开发者将获得更为直观的开发体验。Linaro Demo 的界面如图 2-4 所示。本文附件中提供了最新版的 Linaro Demo 供参考。

## 2.4 详细设计

本节将着重介绍实验从设计转化为实例操作的详细步骤，以及在实验进行中所遇到的问题以及其解决方案。可据此完成对 Robei 与 Vivado 联合设计的初步了解并在实际操作中熟悉整个流程。

### 2.4.1 Robei 软件的使用综述

Robei 软件是世界上最小的芯片设计仿真工具，也是唯一一个移动平台上的 EDA 演示工具。它不依赖于任何芯片，在仿真后自动生成 Verilog 代码，可以与其他 EDA 工具无缝衔接<sup>[3]</sup>。本实例主要利用了 Robei 的模块化设计方法以及波形仿真的功能。其生成的结构化代码具有高度可读性，对应的图形化模块模型为他人理解系统结构提供了很大帮助。

### 2.4.2 利用 Robei 软件进行模块构建以及波形仿真

本实例中，存在多个变量可以进行仿真。由于大部分变量的时序逻辑较为简单，故仅以 2.3.2 中“Countmax”变量作为示例进行仿真。

首先，新建一个模块，并分别定义位宽为 1bit 的一个输入引脚 clk，一个输出引脚 Countmax。接着，在 Robei 软件的 Code 视图中，键入计数器 Verilog 代码，代码如下：

```
reg [18:0] count;

assign countmax = (count == 200000);

assign count = 0;

always @(posedge clk)

begin

    if(countmax)
```

```
count <= 0;  
  
else  
  
count <= count + 1;  
  
end
```

代码运行通过后，新建一个 testbench 测试台。设定同样的输入输出引脚后连接测试台与模块同名引脚。在测试台的 Code 视图中键入检测波形代码：

```
initial  
  
begin  
  
clk=1'b0;  
  
#800000 $finish;    //检测时钟为 800000 次，波形应恰有两次翻转  
  
end  
  
always #1 clk=~clk;
```

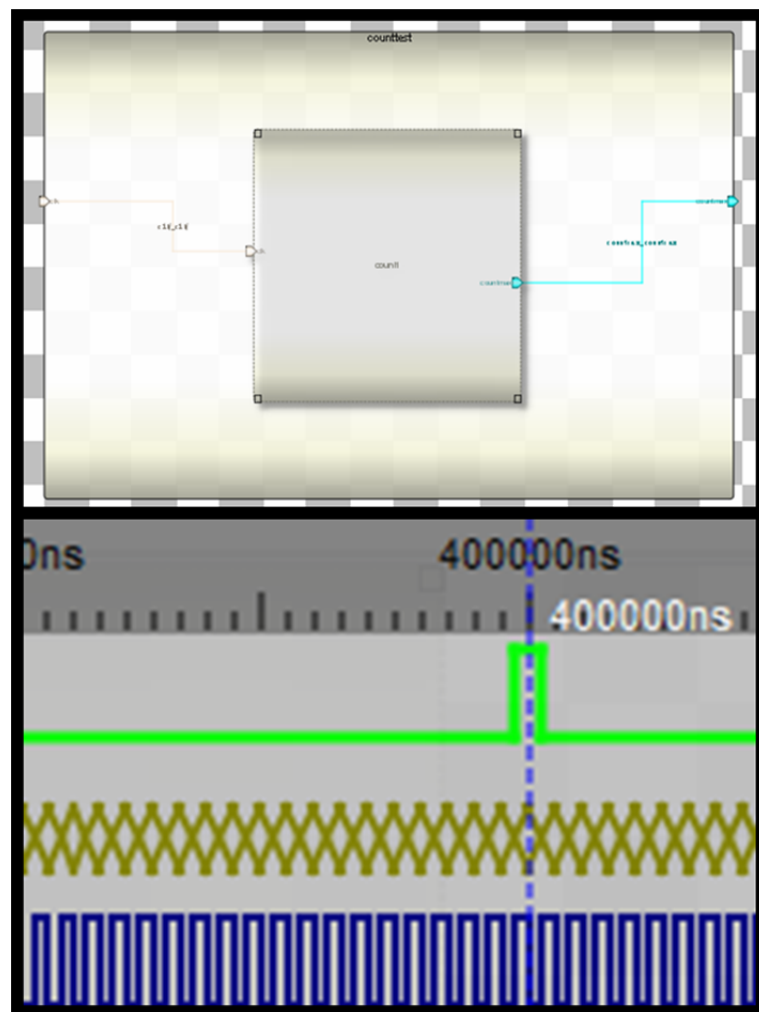


图 2-5 测试台界面以及波形仿真结果

测试台连接与波形仿真界面见图 2-5。经过测试台测试，本实例中的计数器波形正确，可以进行下一步设计。

本实例最初设计为一个主模块嵌套三个子模块的模式，后为化简设计结构改为一个主模块嵌套两个子模块的模式。经检验，后者可正常工作，故最终版采用后一种设计方法。模块设计见附件中 Robei 软件生成的模块文件。

### 2.4.3 Vivado 工程的建立

打开 Vivado 套件，创建 RTL 项目，将编写好的 Verilog 代码添加进去。IP 文件缺省，约束文件缺省，开发板核心选择 xc7z010clg400-1 核心。

建立工程后，首先根据程序设定创建时钟的 IP 文件。设定需要的频率。

### 2.4.4 编写相关约束文件、综合以及比特流生成

Vivado 套件进行综合前最重要的步骤就是编写约束文件，指定引脚。这一步需要细致地检查设计是否存在问题，否则将浪费较多时间在约束文件的错误清除工作上。

本实例中共用到 23 个引脚，下面仅以时钟信号为例介绍约束文件编写方法。

```
#clk of external reference clock

set_property PACKAGE_PIN L16 [get_ports clk116]

set_property IOSTANDARD LVCMOS33 [get_ports clk116]
```

在上面的语句中，我们可以看到，对于约束文件，“#”代表注释。主要工作为在原理图文件中找到所需引脚的编号，并将其与 Verilog 代码中对应的变量进行链接。本例中将 L16 引脚输出的外部时钟作为 clk116 变量的输入，从而为本实例提供时钟信号。

约束文件编写完成后，首先从左侧导航中选择“运行综合”选项。Vivado 将首先进行程序的可行性验证。如果程序存在语法错误、引脚定义有误或存在变量声明的二义性，系统将自动报错。

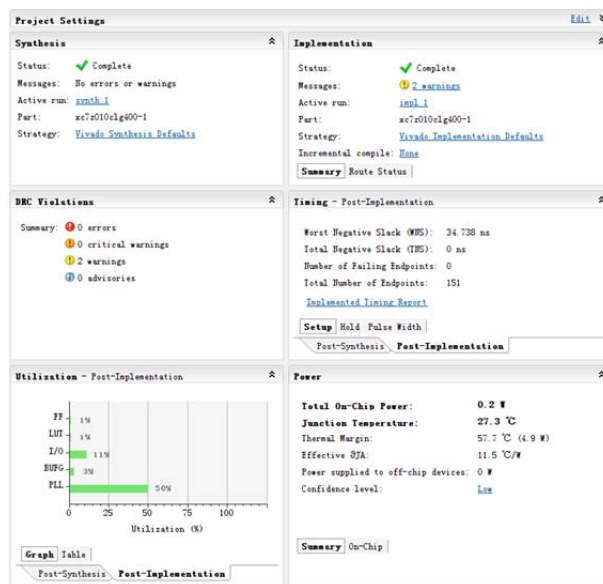


图 2-6 运行实施后项目概况与资源统计

当综合完成且没有错误与警告时，选择“运行实施”选项。Vivado 将模拟资源使用状况，逻辑布线等。实施完成后便可连接 ZyBo 开发板并生成比特流。如果一切顺利，比特流将成功生成，这时便可以在屏幕上验证本实例的功能了。

## 第三章 功能实现

本章将展示本实例在 Zynq Board 开发板上功能实现的成果、具体操作过程以及可能的问题与对策。

### 3.1 实例验证前的准备

在将程序下载到 Zynq Board 开发板上之前，根据功能需要，我们需要准备如下设备：

Mini USB 数据线；支持宽范围屏幕刷新率的显示器；VGA 数据线。

### 3.2 实例验证具体操作过程

#### 3.2.1 开发板连接与比特流下载

首先，将开发板的 VGA 端口连接到显示器上。将开发板连接到电脑，电脑将自动识别为 USB-UART 类型设备。从左侧导航打开 Vivado 的“硬件管理器”界面，点选“自动检测”按钮，软件将会自动识别当前连接的开发板。选择本实例中的 XC7Z010 核心，并右键点选“编程”，比特流会下载到开发板上。此时开发板上绿灯亮起表明成功下载。

#### 3.2.2 功能验证与测试

不要断开开发板与电脑的连接，打开显示器，此时显示器上应出现纯黑的显示区域，下方有白色挡板，并有白色小球在区域中运动。屏幕中央有一“T”行字符砖块，代表作者姓名简写。

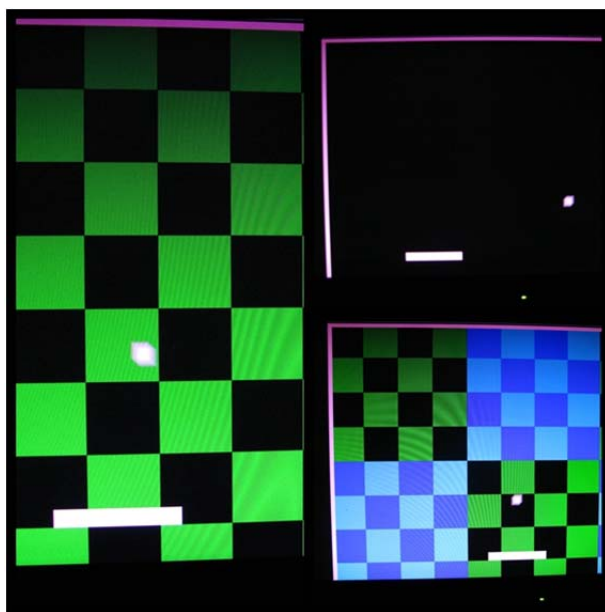


图 3-1 显示器显示本实例

按下开发板上的 btn0, btn3 按钮，挡板应向右或向左移动，并可接住小球实现反弹。若挡板未能接住小球，则一段时间后小球将自动从屏幕上方再次出现在区域内。另外，通过改变左侧拨码开关 sw0、sw1 和 sw2 可以实现区域背景的切换。用户可自行选择适合观感的背景图案。

### 3.2.3 问题与可能的原因

在测试中，可能会遇到以下问题：

- 屏幕没有显示——此时可尝试调节屏幕亮度或调节屏幕的显示位置，直到图像出现。
- 挡板移动与操作不符——可尝试将按下时间延长，直到挡板动作。
- 屏幕显示不正常——请更换宽范围屏幕刷新率的显示器以确保该程序正常运行。同时，请避免热插拔 VGA 电缆，否则可能损坏显示器。
- 图像偏离屏幕中心——如果是显像管显示器，可以通过显示器菜单里的相关设置进行调节；若为液晶显示器，则可以通过显示器自带的“自动适配”按钮来进行调节。



## 第四章 总结与展望

### 4.1 总结与感想

本学期数字电路与逻辑设计课程中，孙老师曾为我们讲述了 FPGA 在当今电子工业中的广泛应用。本人有幸参与到了北京邮电大学的数字电路实验课程新开设的 FPGA 设计：Robei 与 Vivado 联合开发课程中来。在此课程中，青岛若贝科技有限公司的总裁吴国盛先生为我们快速地讲解了 FPGA 设计初步，为我们进一步深入学习 Verilog 以及 FPGA 打开了一扇大门。

从零起步是艰难的，亦是充满挑战与快乐的。无数次程序综合失败、无数次显示不正常。一次又一次的 Debug 后，我终于积累了一定的经验。随着最终程序的完成，我切实感受到自己的 FPGA 开发水平有了质的飞跃。希望电路中心与学校更多地开设此类课程，让同学们有机会动手，在实践中习得真知。

### 4.2 对未来的展望

本实例经历了从模值为 60 的计数器，到主动降噪音频反相器，再到 VGA 输出的思维过程。原先为了尽快完成任务，我打算依照示例设计一个简单的计数器来完成任务。后来发现计数器的实现有难度且没有新意，遂转为设计一个主动降噪器——即音频反向器。本以为这个设计的逻辑将会很简单，但去做时发现真正的难点在于音频采样芯片的调用上。遂再次转为做一个 VGA 输出视频信号的简单游戏。本实例最大的优点在于既利用了 Robei 软件的模块化设计之便，又考验了 Verilog 代码编写能力，还强化了应用陌生软件 Vivado 时的应变排错能力。非常适合其它开发者进行模仿学习。

本实例后续值得改进的有：

- 灯标记命：可通过开发板自带的四个 LED 进行生命值标记，每当漏球便熄灭一个 LED。
- 分数显示：可引入数字字符库，类似本实例中定义字符“T”的方法，实现分数实时显示的功能。
- 打砖块游戏：真正实现一打砖块游戏：即砖块击打后消失，可计分，挡板不同位置改变球运动方向等。基于本实例给出的基本元素，可通过一定设计实现上述功能。期待后续开发者能完善本实例。
- 乒乓球游戏：将本实例改为一双人乒乓球游戏。双方可分别控制两个挡板，漏球可计分等。

## 参考文献

- [1]. 百度百科: FPGA: <http://baike.baidu.com/view/51371.htm>
- [2]. 电子工程专辑: 看了这个你还觉得 FPGA 难学吗? : <http://www.wtoutiao.com/p/gcag3l.html>
- [3]. 青岛若贝科技有限公司官网: <http://robei.com/?lang=zh>
- [4]. ZYBO Reference Manual: [http://www.digilentinc.com/data/products/zybo/zybo\\_rm\\_b\\_v6.pdf](http://www.digilentinc.com/data/products/zybo/zybo_rm_b_v6.pdf)
- [5]. 瑞芯科技设计示例: 使用 FPGA 控制 VGA 显示, July 23, 2004
- [6]. FPGA4FUN-乒乓球游戏设计: <http://www.fpga4fun.com/PongGame.html>

## 致 谢

感谢电路实验中心对本实验的器材及场地支持。

感谢孙辰同学对本实验的器材支持。