



**汇编与接口**

**课 程 设 计**

团队实验报告

|  |  |
| --- | --- |
| 学 院 | 计算机学院 |
| 专 业 | 计算机科学与技术 |
| 指导老师 | 王殿欣 |
| 组 长 | 杨昊 |
| 组 员 | 王欣宇、李昊阳 |
| 组长联系方式 | 13818794782 |

二O二一年八月

目录

[第一章 项目概述 3](#_Toc145412175)

[第二章 设计目标 3](#_Toc145412176)

[第三章 设计环境 3](#_Toc145412177)

[第四章 设计思路 4](#_Toc145412178)

[4.1 基本原理 4](#_Toc145412179)

[4.2 接口关参数确定 4](#_Toc145412180)

[4.3 数据流 5](#_Toc145412181)

[第五章 设计过程 6](#_Toc145412182)

[5.1 Top模块 6](#_Toc145412183)

[5.2 VGA接口模块 7](#_Toc145412184)

[第六章 测试与上板 10](#_Toc145412185)

[第七章 心得体会 11](#_Toc145412186)

## 第一章 项目概述

本项目是一个基于 Verilog 的 VGA 接口设计，旨在将 0-0x400 的内存映射到屏幕上一块 16×16 的区域。通过 VGA 接口，用户可以在屏幕上显示文本、图像和其他内容。该项目使用 FPGA 实现，可以与任何具有 VGA 接口的计算机或显示器兼容。

项目的核心是 VGA 控制器，它负责将内存中的数据转换为 VGA 信号，并在屏幕上显示。VGA 控制器使用 Verilog 代码编写，并实现了一个简单的 16×16 像素的显示缓冲区，用于存储要显示的图像数据。该缓冲区可以通过用户程序进行写入，以更新屏幕上的内容。

同时，我们建立了顶层控制模块将接口和在上一阶段实现的CPU进行了连接，使VGA接口可以在需要的时候从CPU内存中读取数据，刷新自己的缓存，由此便实现了CPU对VGA的控制。

项目还包括一些辅助模块，如地址生成器、时钟管理器和状态机等。地址生成器用于生成内存地址，以便将数据写入显示缓冲区。时钟管理器用于控制 VGA 接口的时序，以确保数据在正确的时间被写入屏幕。状态机则用于管理 VGA 接口的状态，包括显示模式、颜色和分辨率等。

## 第二章 设计目标

实现一个VGA控制接口，与CPU内存中的一块区域进行映射，最终可以通过CPU控制VGA显示的图案，生成动态图像。

## 第三章 设计环境

|  |  |
| --- | --- |
| 操作系统 | Windows x64 10 Pro 22H2 |
| 编程语言 | Verilog |
| EDA工具 | Vivado v2019.2 (64-bit) |
| 汇编语言 | MIPS32 |
| 汇编语言与Verilog代码编辑器 | Visual Studio Code 1.81.1 |
| 开发板 | xc7a35tcsg324-1 |

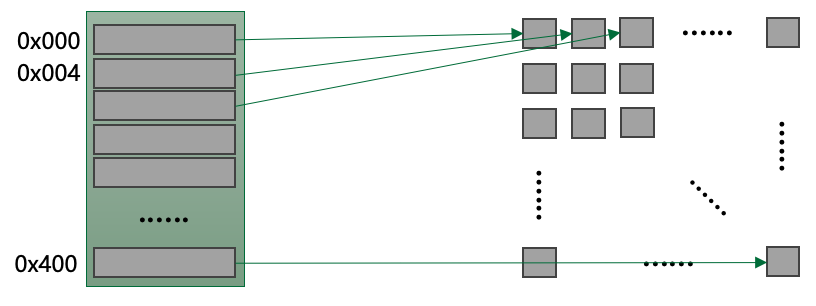
## 第四章 设计思路

## 4.1 基本原理

我们希望呈现CPU对VGA图像的控制以及动态展示内存中数据的效果，因此需要确立VGA与缓存的映射关系。

本项目为VGA配置了单独的缓存空间，当接口收到刷新指令时从指定位置读取数据并刷新缓存。映射关系上，项目选择将地址0x000-0x400的内存空间（共256个双字）作为VGA对应的缓存地址，每个双字对应屏幕上一块40x40大小的像素块。所有的数据合并起来就建立了由内存中256个双字到屏幕上16x16像素块的映射。

数据结构上，因为VGA接收的RGB信号一共占12位，选择32位双字数据的低12位每4位依次对应RGB信号。



## 4.2 接口关参数确定

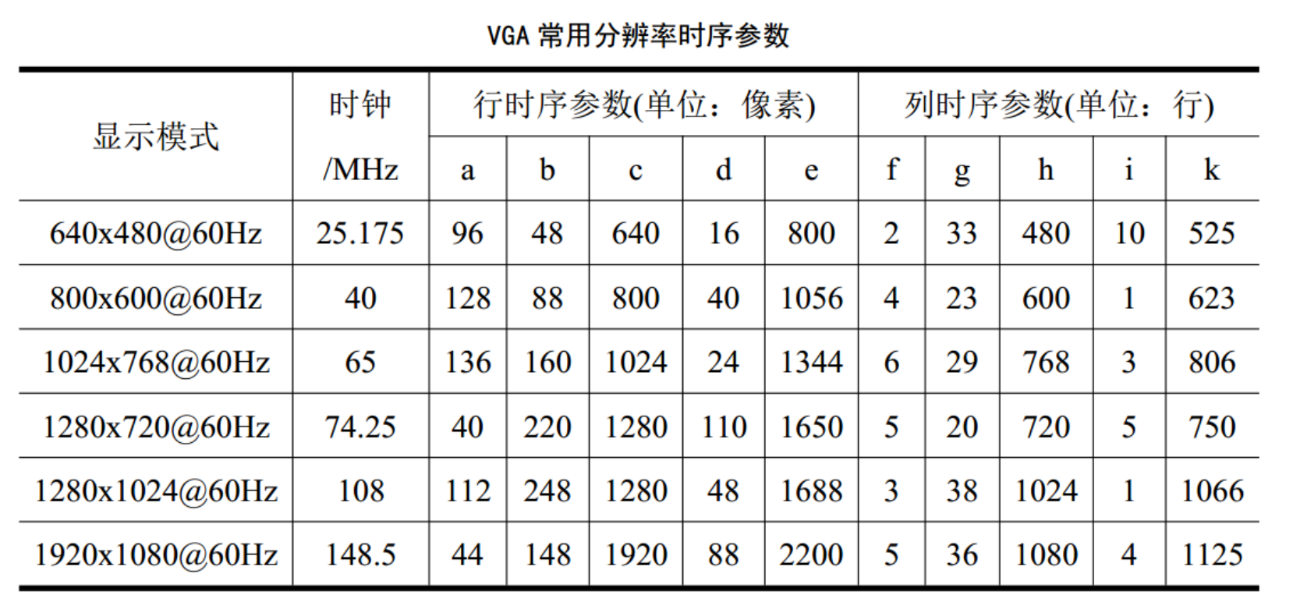
VGA接口的控制信号有两个，分别为行同步信号HS和列同步信号VS，不同分辨率刷新率的屏幕对应不同的HS和VS情况。显示设备通过扫描电子束的方式将图像显示在屏幕上。扫描电子束首先从左到右扫描一行，然后再从上到下扫描下一行，直到扫描完整个屏幕。在扫描的过程中，需要同步电子束的位置和时间，以确保扫描的准确性和稳定性。行同步和列同步信号就是用来同步扫描电子束的位置和时间的信号。行同步信号用于控制扫描电子束的行位置，而列同步信号用于控制扫描电子束的列位置。通过这两个信号的同步，可以确保扫描电子束在正确的位置和时间扫描每一个像素点，从而显示出正确的图像。在选择行同步和列同步信号时需要考虑以下几个方面：

1显示设备的分辨率和刷新率：显示设备的分辨率和刷新率决定了扫描电子束需要扫描的像素点数量和扫描的速度。因此，需要选择合适的行同步和列同步信号频率，以确保扫描电子束能够在正确的位置和时间扫描每一个像素点，并保证图像的流畅度和稳定性。

2显示设备的类型和性能要求：不同类型的显示设备，例如 CRT 显示器、LCD 显示器、投影仪等，对行同步和列同步信号的要求也不同。同时，对于不同的应用场景和性能要求，也需要选择合适的行同步和列同步信号。

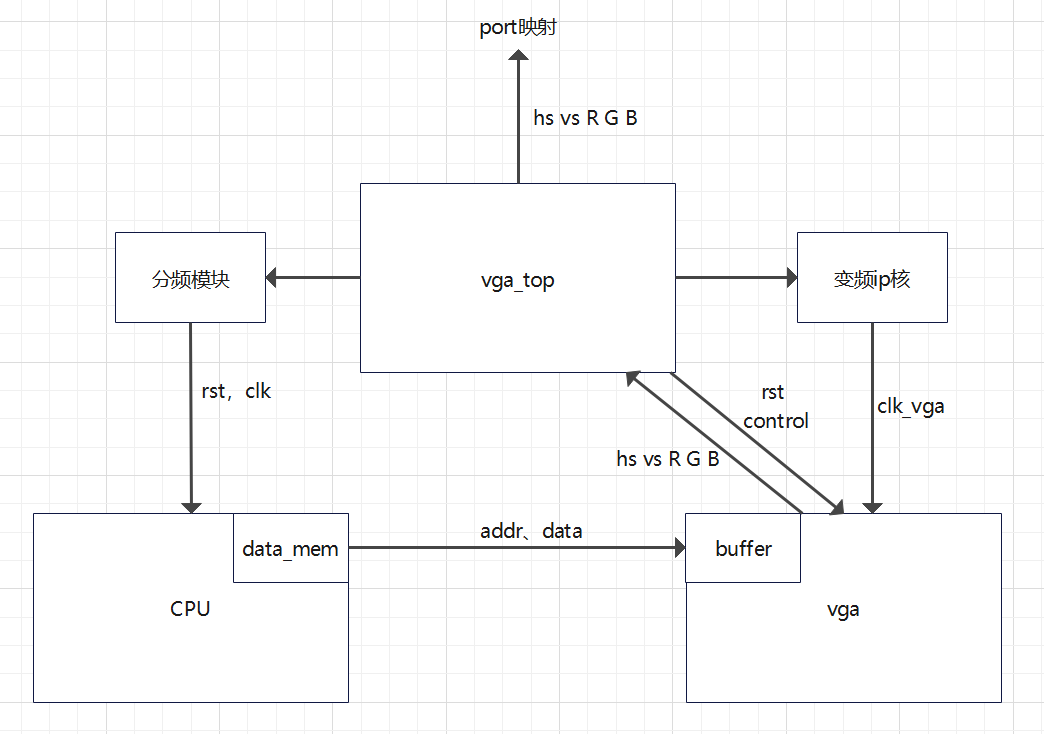
3行同步和列同步信号的相位差：行同步和列同步信号的相位差决定了扫描电子束的扫描顺序和方向。在选择行同步和列同步信号的依据和原因时，需要考虑相位差的影响，以确保扫描电子束的扫描顺序和方向的正确性。

在确定了机房显示器的各项性能参数后，我们查阅资料按照以下表格确定了参数值，其中同类中字母序号由小到大依次对应DISPLAY , FRONT\_PORCH, SYNC\_PULSE , BACK\_PORCH , TOTAL。



## 4.3 数据流

为了更好的控制VGA接口和CPU间的交互，我们设立了一个顶层模块，它负责协调CPU和VGA之间的时钟信号以及控制信号，并最终与管脚相连接。此外还有两个频率变换模块，连接CPU的是为了大幅降低时钟频率使CPU对内存的改变可以在VGA输出上被观测到，连接变频IP核是为了将时钟信号转变为输入需要的65MHz。图中的control信号控制VGA是否要将CPU的缓存区刷新到自己的缓存中。



## 第五章 设计过程

## 5.1 Top模块

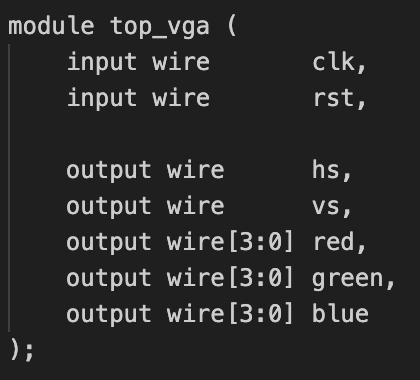
Top模块的主要功能是生成 VGA 图像并综合利用其他所有模块，达成CPU和VGA接口的联动，其输入信号包括时钟信号和复位信号，输出信号包括水平同步信号、垂直同步信号以及红、绿、蓝三原色的输出信号 red、green、blue。

在模块设计中，我们首先定义了一个名为 cpu\_clk 的信号，该信号由外部时钟信号 clk 经过一个名为 changer 的模块处理后得到，处理过程中会受到复位信号 rst 的影响。然后定义了一个名为 BIT\_cpu 的模块，该模块的输入信号包括 clk 和 rst，其输出信号未在代码中明确定义。

为了控制 VGA 图像的生成，我们定义了一个名为 control 的信号，该信号的值为 2 位的 01。同时定义了一个名为 data\_in 的 32 位寄存器，其初始值为 0。为了VGA能正确生成行同步和列同步信号，还需要转变时钟频率，项目中使用了vivvado提供的clk\_wiz频率变换模块，生成65MHz的clk\_vga 时钟信号。

在主逻辑部分，我们首先检查复位信号 rst 是否为有效复位，如果是，则将 addr 信号清零。否则，将 BIT\_cpu 模块的输出数据赋值给 data\_in，并将 addr 信号加 1。在每次时钟上升沿，都会执行这个操作。

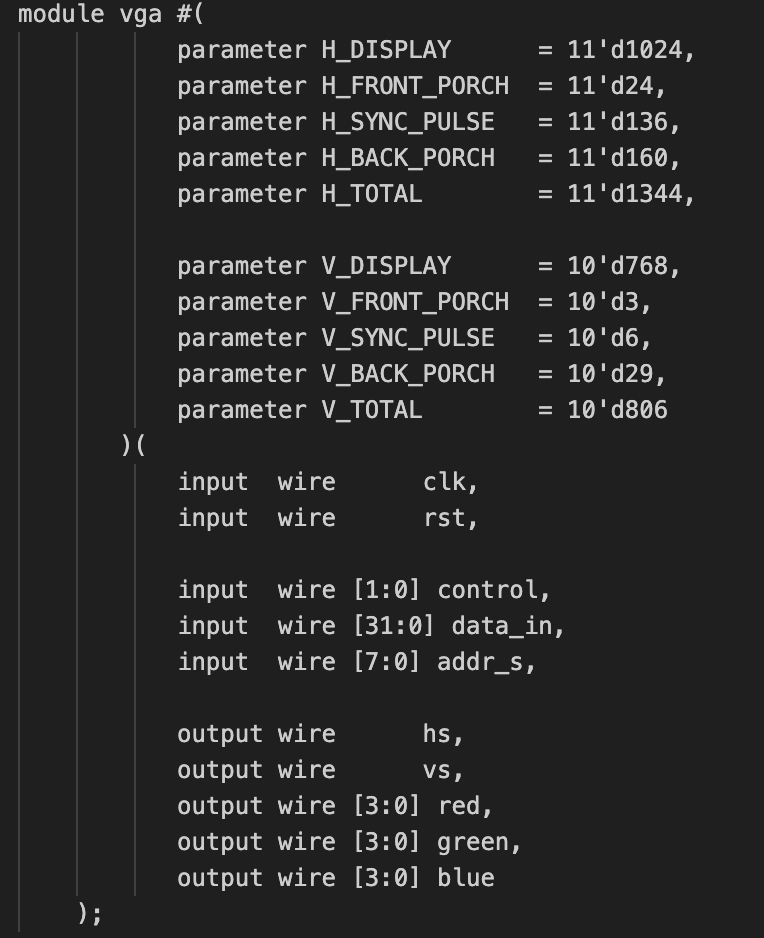
最后，我们定义vga 接口模块，该模块的输入信号包括 clk、rst、control、data\_in、addr\_s，输出信号包括 hs、vs、red、green、blue。vga 模块会根据输入信号生成 VGA 图像，并将图像的同步信号 hs、vs 和颜色信号 red、green、blue 输出。

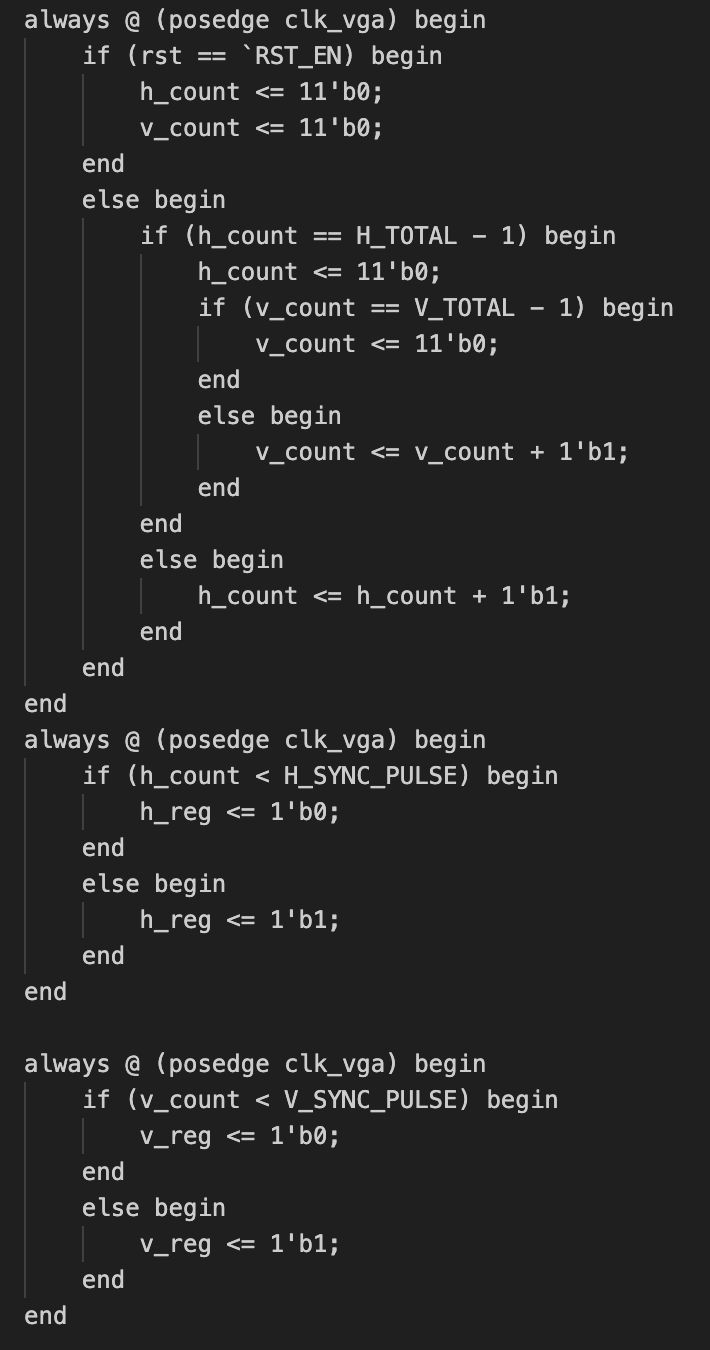


## 5.2 VGA接口模块

VGA接口模块中首先定义了生成我们需要的行同步和列同步信号所必需的参数，具体的值通过查阅第四部分提到的表格得到。例如Total参数代表刷新的最大像素，Sync是同步hs和vs信号所必须的，在代码中使用v\_count和h\_count记录当前显示的像素点位置，由于采用的是行刷新，当到达Total参数所限制的边界时切换所在行，当count越过Sync界限时改变hs或vs的值。

VGA接口接收由Top模块传来的时钟、复位信号，以及对应内存数据的data和addr当control信号最低位是1的时候，将VGA缓存中addr地址的值刷新为data。每次输出RGB信号时都根据当前像素位置判断其对应的缓存地址，从缓存中取出对应RGB信号。



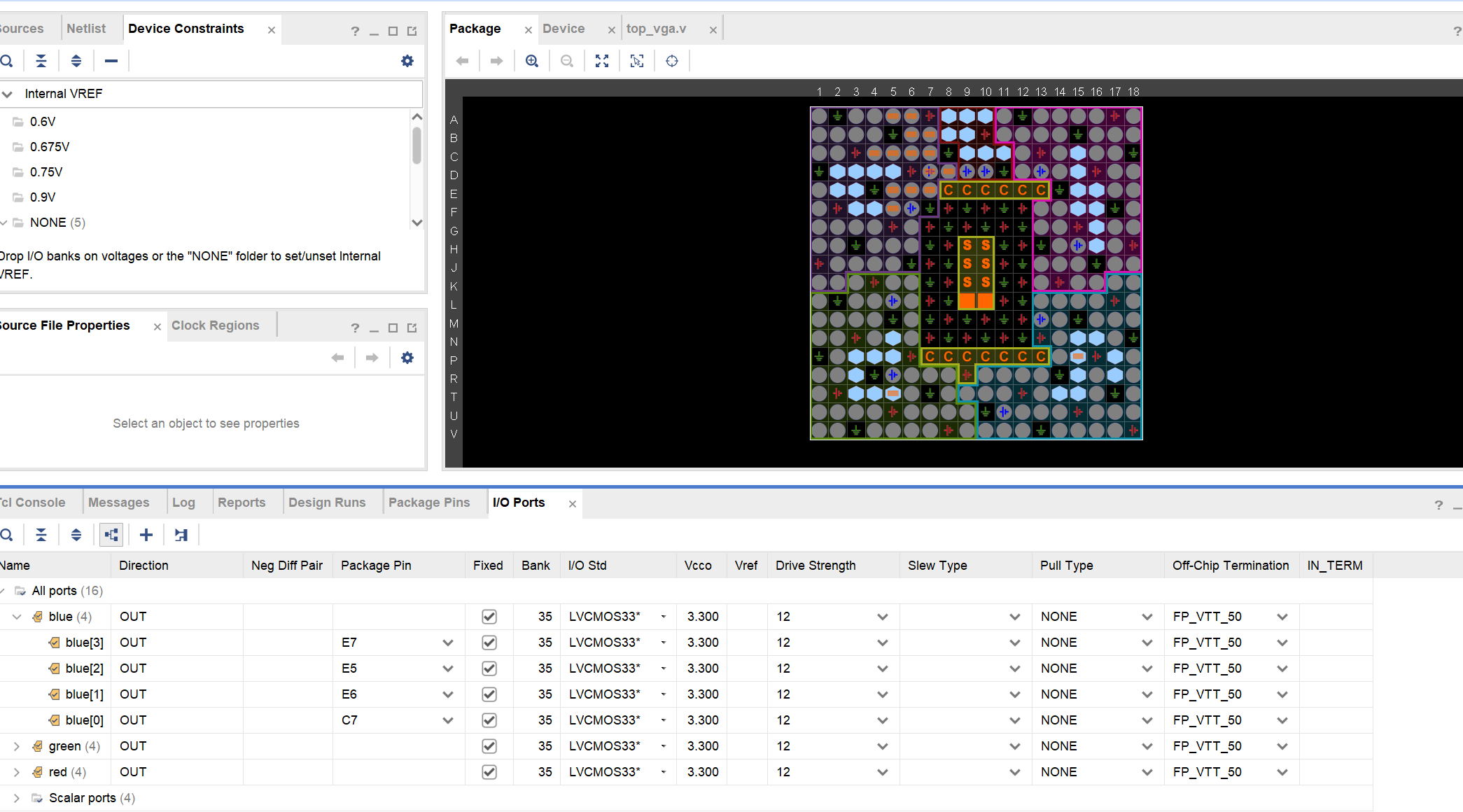


## 第六章 测试与上板

**代码结构**

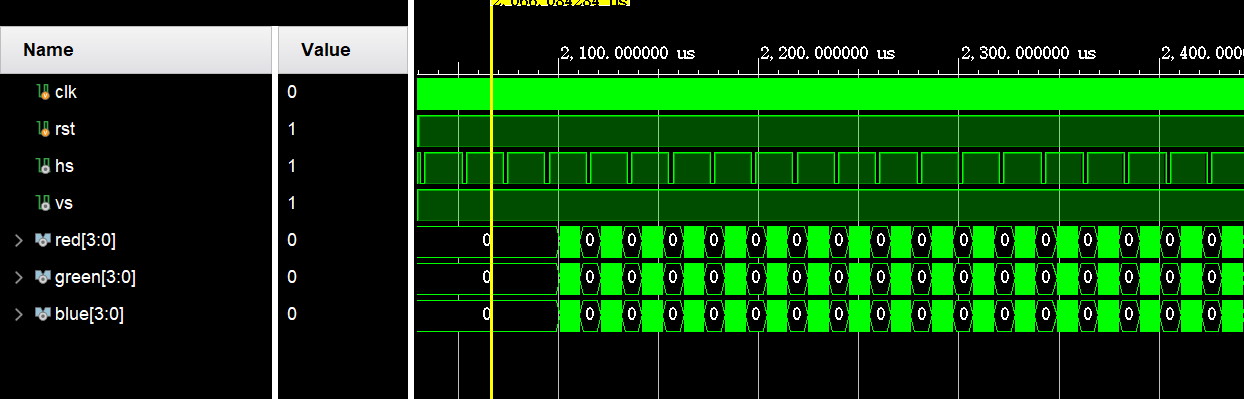


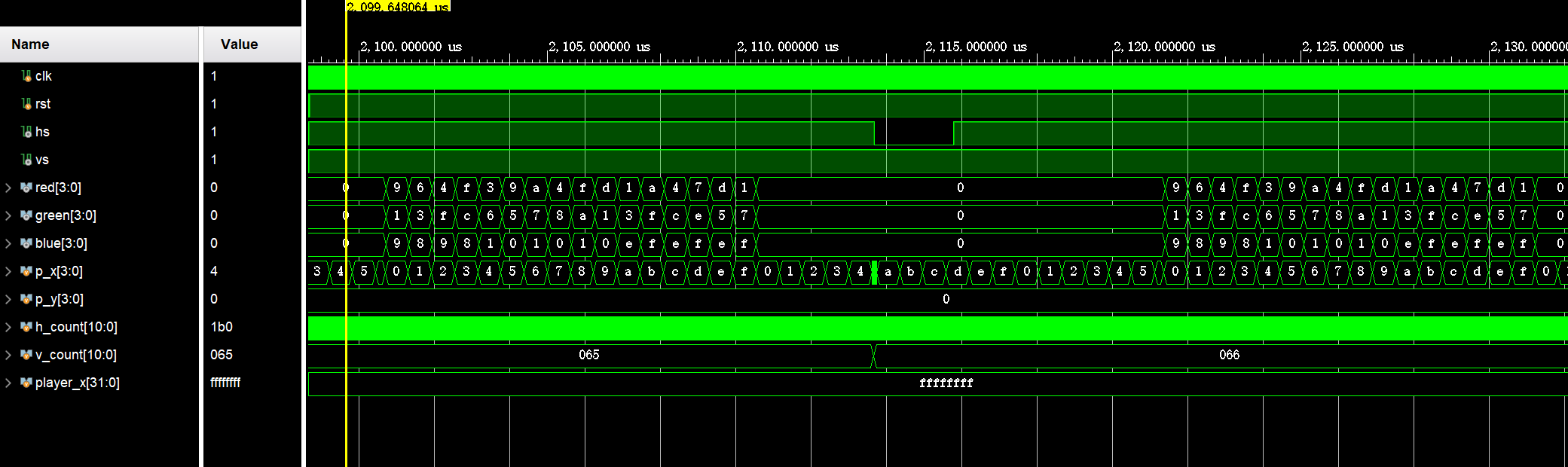
**管脚配置**



**仿真测试**

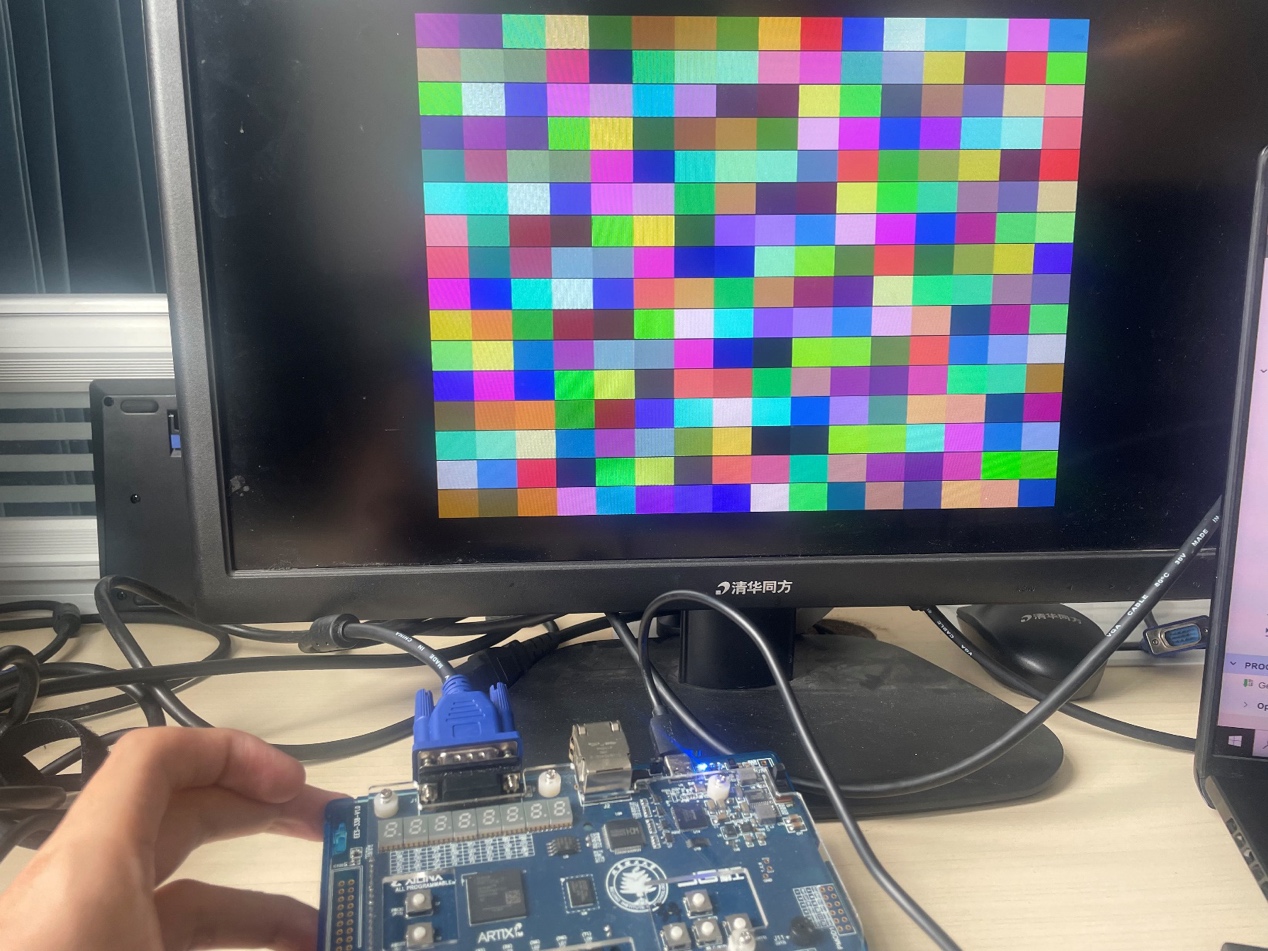
仿真测试在CPU内存中随机初始化了256个数，观察仿真图像发现行同步和列同步信号符合预期，RGB信号周期性变化，每当像素点位于中心方阵中的时候产生波纹颜色输出，区域外则一直为0。





上板测试

与仿真的结果一致，能通过VGA接口进行正确显示。



## 第七章 心得体会

设计一个 VGA 接口并将其连接到流水线 CPU 上，让色块能够在屏幕上显示，是一个非常有趣和有挑战性的项目。在这个项目中，我学到了很多关于数字电路、计算机体系结构、和图形显示的知识。

设计 VGA 接口的过程需要深入了解 VGA 协议和标准。VGA 是一种模拟接口，包含 15 个信号线，用于传输红、绿、蓝三个基色的亮度和色度信号。在设计过程中，我需要考虑如何将这些信号线连接到 CPU 的输出，以及如何在屏幕上显示图像。为了解决这个问题，我使用了一些硬件描述语言来设计和验证我的接口电路。这个过程让我更深入了解了数字电路和计算机体系结构。

在将 VGA 接口连接到流水线 CPU 上时，我需要考虑如何在流水线中添加一个新的操作，用于将图像数据传输到 VGA 接口。在这个过程中，我学习了如何在流水线中插入新的操作，以及如何优化流水线的性能。我还需要考虑如何将 CPU 中的图像数据转换为 VGA 接口所需的格式，以及如何控制 VGA 显示硬件。

最后，在实际看到色块在屏幕上显示的那一刻，我感到非常兴奋和满足。这个项目让我深入了解了计算机图形学的基本原理和实现方法，也让我更加熟练掌握了硬件描述语言和计算机体系结构的知识。我感到非常自豪，能够设计和实现一个完整的 VGA 接口，并将其连接到流水线 CPU 上，实现图形显示。