

课程设计报告

**中文题目 ：** **基于片上Cpu系统的俄罗斯方块**

**英文题目：** **Tetris On SoC**

姓名/学号： 魏世嘉/3130000026

指导教师： 施青松

参加成员： 魏世嘉

专业类别： 计算机科学与技术

所在学院： 计算机学院

**论文提交日期 2014 年 9 月 27日**

摘要

**关键词：**

目录

[摘要 ii](#_Toc399768733)

[1. 绪论 4](#_Toc399768734)

[1.1 设计背景 4](#_Toc399768735)

[1.2 国内外现况分析 4](#_Toc399768736)

[1.3 主要内容和难点 4](#_Toc399768737)

[2. 设计原理 5](#_Toc399768738)

[2.1 设计相关内容 5](#_Toc399768739)

[2.2 设计方案 5](#_Toc399768740)

[2.3 硬件设计 5](#_Toc399768741)

[2.4 系统软件设计（若有） 5](#_Toc399768742)

[3. 设计实现 7](#_Toc399768743)

[3.1 实现方法 7](#_Toc399768744)

[3.2 实现过程 7](#_Toc399768745)

[3.3 仿真与调试 7](#_Toc399768746)

[4. 系统测试验证与结果分析 8](#_Toc399768747)

[4.1 功能测试 8](#_Toc399768748)

[4.2 结果分析 8](#_Toc399768749)

[4.3 系统演示与操作说明 8](#_Toc399768750)

[5. 结论与展望 10](#_Toc399768751)

1. 绪论

### 设计背景

MIPS(Microprocessor without interlocked piped stages architecture)架构，是一种采取精简指令集(RISC)的处理器架构。常见的MIPS构架为32位指令集体系，最新的版本也已经支持64位。MIPS构架在计算机演进历史上担任了重要角色，又因为其RICS的特性，是学生学习底层计算机组成以及一些简单嵌入式开发的重要平台，应用广泛。基于Diligent Nexsy3 FPGA平台可搭建MIPS32处理器架构，在实现CPU的基础上，添加裸机外设，并运行简单的应用程序。

### 国内外现况分析

MIPS体系架构相对成熟，由于采用RISC体系，32位定长的指令使得MIPS不仅在业界有着广泛应用，在高校计算机教育中也有重要地位。国内有类似设计的院校主要有清华、浙大等211,985院校，同时台湾及香港有不少院校的计算机专业或电子工程专业的毕业生选择做MIPS架构的毕业设计。在国际上，许多大学，比如美国的犹他大学也有相类似的课程与实践项目。项目相关的代码工程在github托管网站上均能找到。

另一方面，由于CPU设计分有单周期，多周期与流水线的设计方式，大多数设计采用的是五级流水线的设计方式，与本设计中实现的多周期不同，五级流水线的设计使得CPU硬件利用率更高，在相同稳定时钟平率的条件下，流水线设计的CPU比多周期的设计更高效。

### 主要内容和难点

主要内容分为硬件设计和软件设计两块。

硬件部分设计包括：完成在原MIPS32多周期CPU的基础上完成CPU与外设VGA,PS/2键盘的连接，设计VRAM实现VGA显示的文本模式等；

软件部分设计包括：完成系统简单平台的设计（包括精简的shell系统），CPU轮询策略，PS/2键盘相应，硬件计数器响应以及VRAM的写入等。

设计实现的过程中，主要的难点在于软件的设计，由于硬件系统一定程度上的不完善性（支持的指令只有二十余条），软件设计存在较大瓶颈，同时在硬件系统上并没有实现中断机制，CPU只能采取轮询的策略来替代中断。

1. 设计原理

### 设计相关内容

课程设计中，主要涉及的是Verilog行为描述设计PS/2接口程序，VGA控制和接口读写程序，以及MIPS汇编语言实现shell命令控制以及俄罗斯方块游戏的设计。

硬件设计方面需要对行为描述语言以及时序电路时延等相关设计因素有周密考虑。在PS/2接口模块中，尤其需要考虑时钟频率对PS/2采样信号的影响。

另一方面MIPS汇编语言的软件设计上，需要实现对PS/2键盘扫描码的译码工作，并将相应的字符打印到屏幕上，在译码过程中，原先有采用硬件译码的方式，但处于规范的考虑改用软件译码。采用打表的方式对每一个键盘按键均可快速响应并获得对应的ASCII码或控制码。

### 设计方案

硬件设计主要是在原先完成的MIPS多周期CPU的基础上，添加VGA与PS/2外设，并设计VRAM，同时联通CPU对VGA与PS/2键盘的读写通道。

在完成硬件设计之后，整个运行系统的设计就进入了MIPS软件设计体系，进入利用汇编程序写软件的模块。软件主要分为系统软件和游戏软件。主要系统软件包括键盘按键响应程序，命令显示程序和命令检查程序，详细信息见3.1节。而游戏软件则是主要的俄罗斯方块设计。

### 硬件设计

硬件设计主要参考了课程设计老师提供的开源VGA控制器代码以及PS/2键盘接口代码。在原先设计多周期CPU工程的基础上，添加了模块，使得新增模块与原工程几乎完美兼容。具体模块图见下。

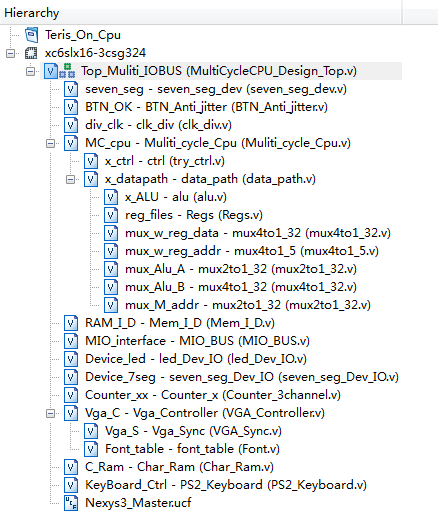


Figure 1 Hardware Hierarchy

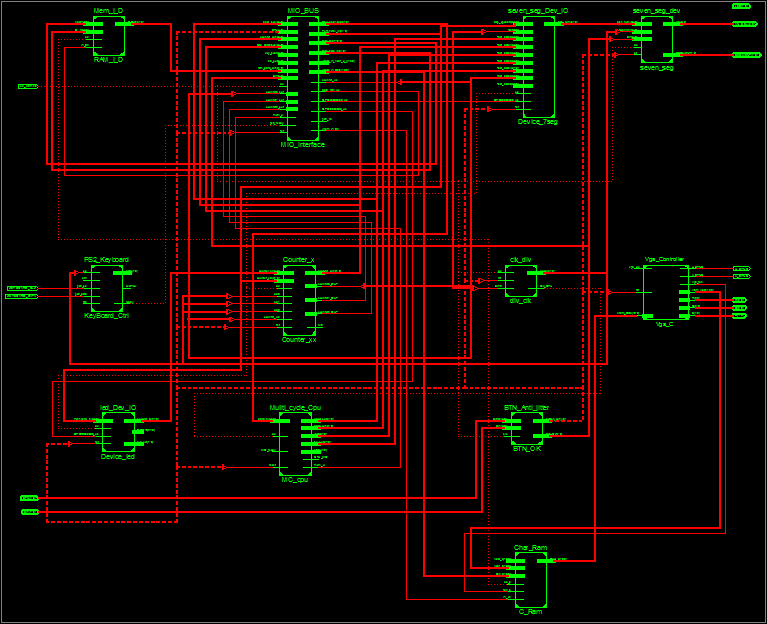


Figure 2 Hardware Schematic Graph

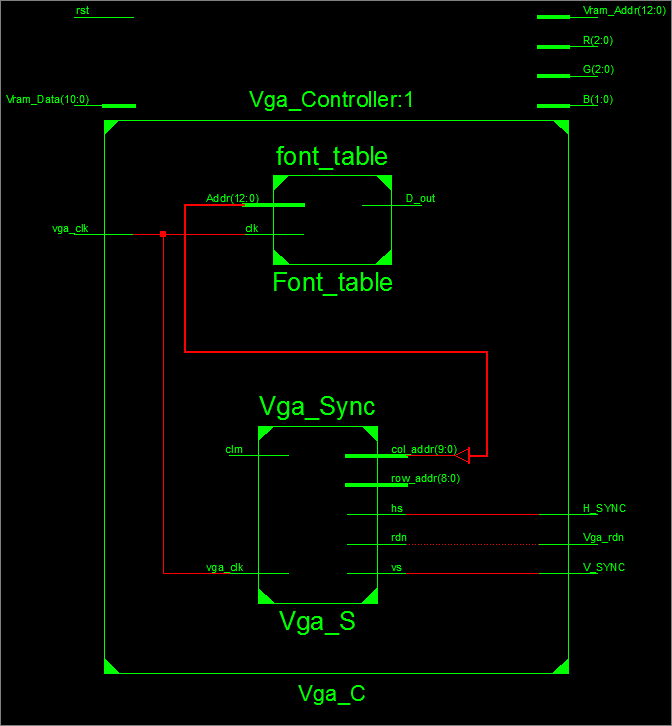


Figure 3 Schematic Graph of VGA Controller

### 系统软件设计（若有）

除游戏程序外，其余MIPS汇编程序均可在支持MIPS基本指令集的CPU上运行，并获得相同的效果。分别为对应的键盘按键响应程序，命令屏幕显示程序和命令检查程序。

系统运行流程图见下页。

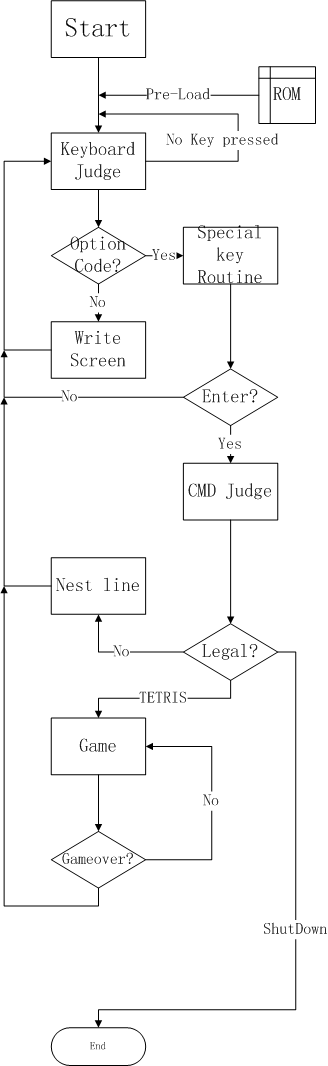


Figure Flow Chart

其中键盘响应程序利用了打表的方法，在内存中预存扫描码对应的ASCII和控制码的信息，可快速响应与获得。另外关于命令屏幕显示程序，其中涉及的主要内容包括读写VRAM,换行与清屏功能的实现，屏幕显示地址的计算等，合理运用beq与bne指令即可完成。

而命令检查功能，可用于判断当前行所输入的指令，目前所添加的指令仅有TETRIS一条游戏指令，但可判断检查所有命令行无参指令。目前尚不支持命令行参数功能。

1. 设计实现

### 实现方法

硬件设计实现均较为简单，原CPU代码也在早先的报告中详细讲述，不在赘述。对于VGA与PS/2键盘与CPU联通的硬件描述，采用行为描述的方式实现，方便快捷。

通过对行场计数器的控制，实现了生成了VGA行场同步信号，只要是的信号保持在60HZ/Screen左右即可是的VGA自动同步显示。

PS/2键盘模块，通过对Ps2\_clk信号下降沿的监测，读取Ps2\_data信号便可得到键盘扫描码，此处需要注意的是PS/2扫描信号频率相对CPU时钟频率不可过高，否则易导致PS/2数据的误读。

接下来介绍系统软件中的功能的实现方法。

对键盘的相应程序中，主要采取打表方法。即固定ASCII编码数据在内存中存储的起始地址。编写程序生成扫描码对应的ASCII码，并按扫描码从小到大排序，空码补20（空格的ASCII），将生成的数据采用打表的方式嵌入内存空间中，处理扫描码时，将其作为ASCII表起始地址的偏移地址（需要一定的以为操作）便可获得对应的扫描码。

对于命令显示功能，程序利用MIPS寄存器设置了显存地址寄存器和显示地址行列计数寄存器等程序内专用寄存器来完成字符的显示功能，最后连续显示，完成命令显示功能。

命令检查功能相对简单，运用存储长度为二十的字符数组存储当前行的命令，对已实现命令进行逐一比较即可。暂不支持命令参数功能。

### 实现过程

实现过程主要分为硬件实现以及软件实现。硬件在原先CPU设计的基础上实现起来较为简单。同时在软件方面，编写MIPS汇编程序，将各个功能模块化，使得各部分功能模块得以调试通过。

### 仿真与调试

仿真与调试主要为软件调试。多次测试后使得系统软件和游戏应用均调试通过。

1. 系统测试验证与结果分析

### 功能测试

系统功能相对显见，只需在VGA上通过相应的操作显示即可。而对于俄罗斯方块的游戏应用。我们可以认为当游戏可以顺畅运行时，整个游戏的应用在功能上已达到要求。

### 结果分析

该系统可在设计的多周期CPU上运行俄罗斯方块的应用程序，并已完成了俄罗斯方块该游戏的基本功能。

### 系统演示与操作说明

主要演示结果详见视频，以下图文仅供参考。



Figure 5 Init Interface



Figure 6 CommandLine in System

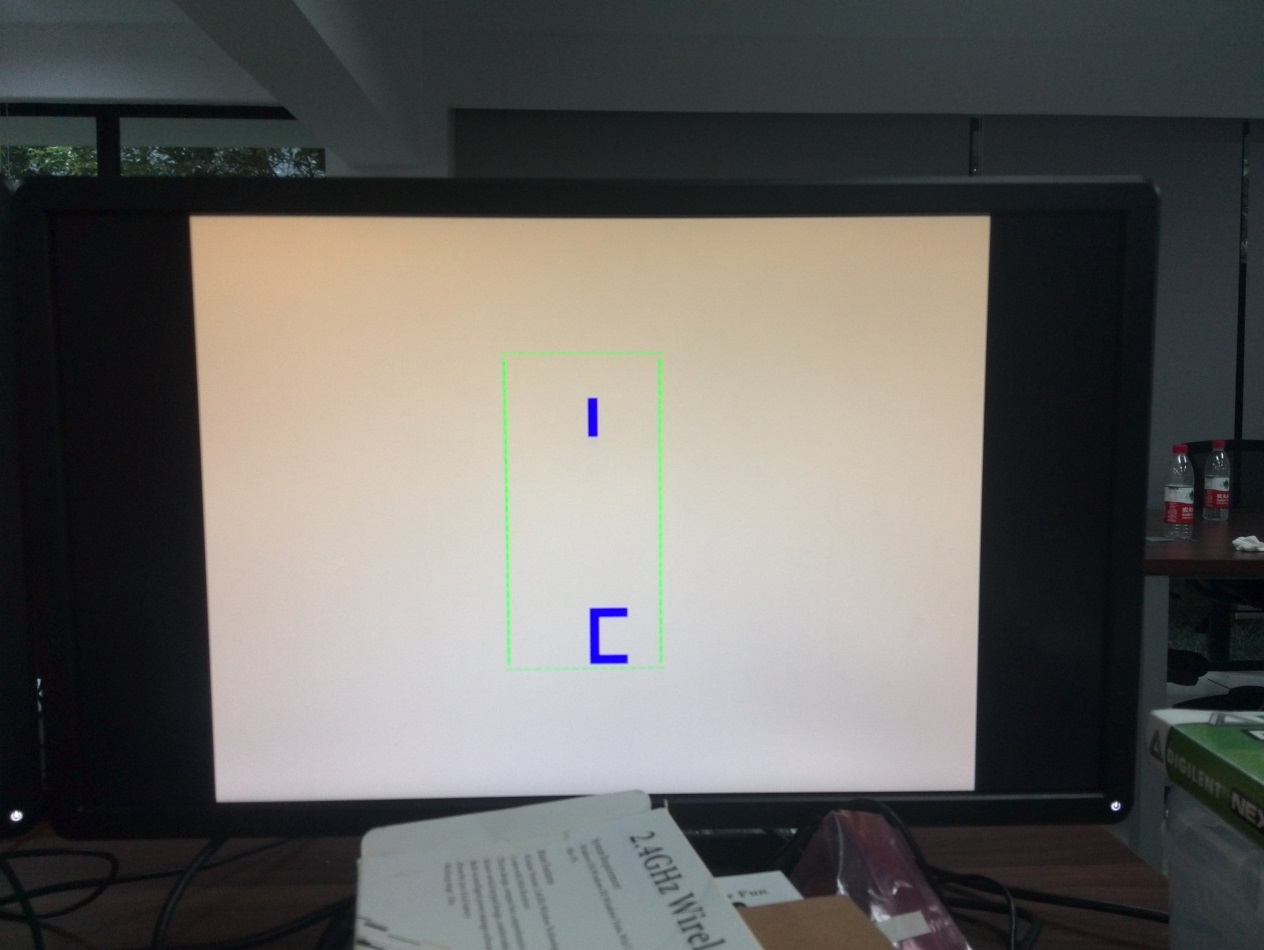


Figure 7 Tetris Game

在进入系统后，可以在shell界面输入任意字符，并进行命令的判断。

进入游戏后，可用A和D键控制俄罗斯方块的左右移动，SPACE键控制俄罗斯方块方向。

1. 结论与展望

在系统设计的过程中，系统开发的主要精力集中在后期的系统软件和游戏应用的开发上，由于硬件结构相对完整，只需稍稍注意时序问题即可。而汇编程序编写系统软件以及游戏应用的过程相对比较漫长。尤其在Debug的过程中，需要尤其注意是硬件问题导致的bug还是软件代码的漏洞。由于缺少与系统相符合的模拟器，调试过程相对费时。在接下来的过程中，希望能将游戏和系统功能完善后，完成一个适合该系统的模拟器。

另一方面，在系统设计的过程中没有设计到多周期中断的实现，所以在后期的设计过程中，希望能进一步完善中断机制和一定程度上的SD卡文件系统。