THCO MIPS 大作业

设计报告

计32

陈文潇 何轩 李奕昕

目录

[一、 实验概述 2](#_Toc437511872)

[1. 实验目的 2](#_Toc437511873)

[2. 实验环境 2](#_Toc437511874)

[序号、内存储器访存 3](#_Toc437511875)

[1. 基本架构 3](#_Toc437511876)

[2. 存储器控制单元 AddrMem 与 DataMem 3](#_Toc437511877)

[3. 存储器单元 Ram1Ctrl, Ram2Ctrl 3](#_Toc437511878)

[4. 访存冲突处理 4](#_Toc437511879)

# 实验概述

## 实验目的

1. 加深对计算机系统知识的理解
2. 进一步理解和掌握流水线结构计算机各个部件组成及内部工作原理
3. 掌握计算机外部输入输出的设计
4. 培养硬件设计和调试的能力

## 实验环境

1. 硬件环境： 安装有Windows 10操作系统的微机，THINPAD教学计算机。其中，THINPAD教学计算机包含：FPGA CPU, RAM Memory, FLASH Memory, UART/USB/PS2/VGA
2. 软件环境： FPGA开发工具软件Xilinx ISE 14.7, THCO MIPS 指令系统，THINPAD教学计算机软件包，包括监控程序、终端程序、数据通信程序、汇编语言等。

# 序号、内存储器访存

## 基本架构

在数据通路的设计中,我们采用了将指令存储器（AddrMem）与数据存储器(DataMem)分离共两个单元的设计。基于THINKPAD教学计算机的基本设计，AddrMem与DataMem功能相分离，AddrMem单元负责对Ram2的读操作，主要用于实现读出Ram2中存储的指令。DataMem单元负责对Ram1,Ram2的读与写操作。可以发现，这二者在对Ram2的操作中存在一定冲突，二者可能会同时对Ram2进行操作，因此在此处访存的冲突处理应当被考虑在内。

## 存储器控制单元 AddrMem 与 DataMem

* 1. AddrMem指令存储器控制单元

AddrMem单元的主要功能是实现对Ram2存储器中指令的读取，即其只涉及对Ram2存储器的读操作。

由于AddrMem功能相对简单，只负责对Ram2存储器的读操作，因而其作为访问Ram2的入口，内部为组合逻辑，直接将其输入进行判断操作后与其输出相接。

* 1. DataMem数据存储器控制单元

DataMem单元的主要功能是与其他单元协作，实现系统及用户数据的存取，以及在需要时对Ram2指令存储器的读取与修改。功能提取可以了解到，相关功能涉及对Ram1与Ram2的读与写。与此同时，由于串口以及VGA与键盘相关的读写基本等同于对Ram1的读写，DataMem同时需支持对串口与VGA+键盘的控制。

DataMem相较AddrMem而言较为复杂，涉及对Ram1Ctrl, Ram2Ctrl 两个单元进行读写控制，其内部依然如同AddrMem单元，采用组合逻辑，效率相比时序逻辑更高。由于需要对输入数据进行判断，并根据访存地址的不同将输入转接到不同存储单元的对应接口，我们这里采用效率较高的when else 语句，如下：

DataMem\_Ram1\_EN <= '1' when DataMem\_Addr >= "1011111100000000" and DataMem\_EN = '1' else '0';

## 存储器单元 Ram1Ctrl, Ram2Ctrl

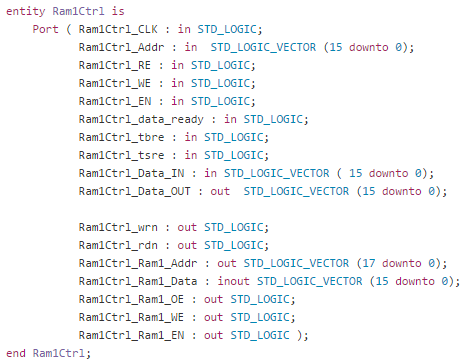
1）内存访问时序及信号

内存使用实验板上的Ram1与Ram2两个存储单元，其中地址Addr为18位，Data为16位。本次CPU设计中暂不使用Addr的高2位，默认置为0。

2）Ram1Ctrl 单元设计

Ram1与Ram2的地址段划分不同，Ram1的地址范围为： 0xBF00~0xFFFF。其中0xBF00~0xBF03看似为Ram1中的地址，事实上为串口的访问地址。

Ram1Ctrl单元的相关接口设计如下：



其中，Ram1Ctrl\_EN, Ram1Ctrl\_RE, Ram1Ctrl\_WE分别为总使能，读使能，写使能。Ram1Ctrl\_Data\_IN为写入时需写入的数据，Ram1Ctrl\_Data\_OUT为读取时读到的数据。Ram1Ctrl\_data\_ready, Ram1Ctrl\_tbre,Ram1\_Ctrl\_tsre,Ram1\_wrn,Ram1\_rdn都为与串口交互相关的接口。至于下方5个接口则是直接通过总线与Ram1单元交互的接口。之所以单独设计Ram1Ctrl单元而不是将其功能综合进DataMem单元，是由于AddrMem与DataMem可能会同时读写Ram2单元，这其中，Data是inout类型的接口，对其作选择器较为困难，故将Ram的读写单元单独设计，以实现对inout接口的分离。

关于内存部分的读写时序如下图：

读时序：



写时序：



考虑到流水线运行的效率，我们摒弃了在之前做的内存储器实验中采用的状态机转移方法来控制读写逻辑，而是采用了高低电平判断的方式来使得一个周期内可以包含两种状态。每个时钟周期默认所有使能为关。

数据读写部分： Addr > 0xBF03

读状态：读使能开启写使能关闭即Ram1Ctrl\_RE = ‘1’ and Ram1Ctrl\_WE = ‘0’。此时若为时钟高电平则读入地址Addr并开启Ram1的总使能Ram1\_EN，若为低电平则开启Ram1的读使能Ram1\_OE。

写状态：读使能关闭写使能开启即Ram1Ctrl\_RE = ‘0’ and Ram1Ctrl\_WE = ‘1’。此时若为时钟高电平则读入地址Addr并开启Ram1的总使能Ram1\_EN，若为低电平则开启Ram1的写使能Ram1\_WE。

串口读写部分：Addr<=0xBF03

读状态：读使能开启写使能关闭即Ram1Ctrl\_RE = ‘1’ and Ram1Ctrl\_WE = ‘0’。此时若为时钟高电平则读开启Ram1的总使能Ram1\_EN，若为低电平则开启串口的读使能Ram1Ctrl\_rdn。

写状态：读使能关闭写使能开启即Ram1Ctrl\_RE = ‘0’ and Ram1Ctrl\_WE = ‘1’。此时若为时钟高电平则开启Ram1的总使能Ram1\_EN，若为低电平则开启串口的写使能Ram1\_Ctrl\_wrn。

VGA与键盘读写部分：

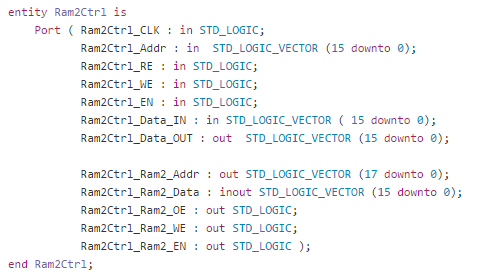
VGA与键盘方面完全组合成了Term单元，仿照串口工作，因此将所有的与串口的通信复制一份与Term单元即可。

数据写入方面，为了避免可能的冲突，统一在时钟下降沿写入。读状态时将Ram1\_Data 设置为高阻态，写状态时将Ram1\_Data设置为输入的Data\_IN。

3）Ram2Ctrl 单元设计

Ram1与Ram2的地址段划分不同，Ram2的地址范围为： 0x0000~0xBEFF

Ram2Ctrl单元的相关接口设计如下：



其中，Ram2Ctrl\_EN, Ram2Ctrl\_RE, Ram2Ctrl\_WE分别为总使能，读使能，写使能。Ram2Ctrl\_Data\_IN为写入时需写入的数据，Ram2Ctrl\_Data\_OUT为读取时读到的数据。至于下方5个接口则是直接通过总线与Ram2单元交互的接口。之所以单独设计Ram2Ctrl单元而不是将其功能综合进AddrMem单元，是由于AddrMem与DataMem可能会同时读写Ram2单元，这其中，Data是inout类型的接口，对其作选择器较为困难，故将Ram的读写单元单独设计。

关于内存部分的读写时序如下图：

读时序：



写时序：



考虑到流水线运行的效率，我们摒弃了在之前做的内存储器实验中采用的状态机转移方法来控制读写逻辑，而是采用了高低电平判断的方式来使得一个周期内可以包含两种状态。每个时钟周期默认所有使能为关。

读状态：读使能开启写使能关闭即Ram2Ctrl\_RE = ‘1’ and Ram2Ctrl\_WE = ‘0’。此时若为时钟高电平则读入地址Addr并开启Ram2的总使能Ram2\_EN，若为低电平则开启Ram2的读使能Ram2\_OE。

写状态：读使能关闭写使能开启即Ram2Ctrl\_RE = ‘0’ and Ram2Ctrl\_WE = ‘1’。此时若为时钟高电平则读入地址Addr并开启Ram2的总使能Ram2\_EN，若为低电平则开启Ram2的写使能Ram2\_WE。

数据写入方面，为了避免可能的冲突，统一在时钟下降沿写入。读状态时将Ram2\_Data 设置为高阻态，写状态时将Ram2\_Data设置为输入的Data\_IN。

## 访存冲突处理

在设计中，读取指令的指令存储器控制单元AddrMem与数据存储器控制单元DataMem同时操作Ram2单元，于是这里便存在了冲突。我们采用了插气泡让其暂停一个周期的方法来解决该问题。

DataMem中存在有一输出接口DataMem\_Stop\_EN，使用组合逻辑来判断，每当DataMem的输入地址Addr经过判断为Ram2的Addr时，DataMem\_Stop\_EN输出为’1’，以表示此时为DataMem对Ram2访问优先，需要暂停AddrMem对Ram2的操作，同时暂停一个周期。否则DataMem\_Stop\_EN = ‘0’。

冲突的控制选择，我们采用了二选一选择器来进行选择，控制信号为DataMem\_Stop\_EN，当DataMem\_Stop\_EN = ‘1’时，选择来自DataMem的数据与控制信号连接到Ram2Ctrl的对应接口上，当DataMem\_Stop\_EN = ‘0’时，选择来自AddrMem的数据与控制信号连接到Ram2Ctrl的对应接口上。

# 序号、外接设备

## VGA端口模块

VGA

## PS2键盘模块

## Term组合模块