

Tsinghua University  
Department of Computer Science and Technology

## Mip 需求文档

瞿凡	孙维孝	钟闰鑫
2017010636	2017010650	2017010306

First version

October 2019

# Abstract

这是我们 IntoMips 项目的需求文档的初步版本，我们的工作将针对需求展开。

# 目录

Abstract	i
<b>1 前言</b>	<b>1</b>
1.1 背景	1
1.2 编写目的	1
1.3 项目目标	1
<b>2 基本部件</b>	<b>2</b>
2.1 ALU	2
2.2 乘法器	2
2.3 寄存器组	2
2.4 协处理器	2
2.5 MMU	3
2.6 TLB	3
2.7 异常与中断处理	4
<b>3 流水线</b>	<b>5</b>
3.1 流水线概述	5
3.1.1 流水线冒险	5
3.1.2 流水线数据通路及控制	6
3.1.3 功能需求	6
<b>4 外存</b>	<b>8</b>
4.1 SRAM	8
4.2 Flash	8
<b>5 外围设备</b>	<b>9</b>
5.1 串行接口	9
5.2 DVI 图像输出接口	9
5.3 以太网口	10

5.4	USB 接口 . . . . .	10
<b>6</b>	<b>工具链</b>	<b>11</b>
6.1	硬件环境 . . . . .	11
6.2	软件环境 . . . . .	11
6.3	调试工具 . . . . .	11
<b>7</b>	<b>附录</b>	<b>12</b>
	<b>Bibliography</b>	<b>13</b>

# 1. 前言

## 1.1 背景

本项目旨在实现一个 MIPS32 的 CPU 及其周边硬件，能够支持标准 MIPS32 指令集的一个子集，最后在那个能够运行 ucore 操作系统。

## 1.2 编写目的

本文档旨在明确项目需求，对项目进行总览，把握方向。

## 1.3 项目目标

1. 实现 CPU、MMU、通信等功能。
2. 顺利运行 CPU，接受用户输入，执行对应功能。
3. 逐步提高主频。

## 2. 基本部件

### 2.1 ALU

ALU 全称为算术逻辑单元 (Arithmetic Logic Unit)，是实现多种算数运算和逻辑运算的组合逻辑电路，是 CPU 中的核心组成部分。具体实现中，ALU 以一个控制信号和两个 32 位整数为输入，一个 32 位整数为输出，ALU 根据控制信号执行相应的操作。从 ucore 操作系统的实际需求出发，ALU 的运算需求可以整理为下表。

操作码	功能	描述	操作码	功能	描述
ADD	$A + B$	加法	SLL	$A \ll B$	逻辑左移 B 位
SUB	$A - B$	减法	SRL	$A \gg B$	逻辑右移 B 位
AND	$A \text{ and } B$	按位与	SRA	$A \ggg B$	算术右移 B 位
OR	$A \text{ or } B$	按位或	EQU	$A = B$	A 是否等于 B
XOR	$A \text{ xor } B$	按位异或	SLT	$A < B$	A 是否小于 B
NOT	not A	按位非			

Table 2.1: ALU 功能表

### 2.2 乘法器

乘法器 (Multiplier) 用于执行乘法指令，以两个 32 位整数为输入，以 64 位整数为输出。在 MIPS32 指令集中，运算结果保存在 HiLo 寄存器中。其中高 32 位保存在 Hi 寄存器中，低 32 位保存在 Lo 寄存器中。从 ucore 操作系统的实际需求出发，我们只需要实现 MULT 和 MULTU 两种指令。常见的乘法器模型基于布斯乘法。具体算法不在此赘述。

### 2.3 寄存器组

寄存器 (Register) 是容量有限的高速储存部件，用于暂存指令、数据、地址等。基本寄存器由 D 触发器构成，在 CP 时钟信号的上升沿写入数据，其余保持不变。寄存器读写速度极快，相对于内存和硬盘有着显著的性能优势。MIPS32 中有 32 个通用寄存器，被命名为 \$0 至 \$31。

### 2.4 协处理器

协处理器 (Coprocessor) 表示处理器的一个可选部件，通过拓展指令集或提供配置寄存器的方式来拓展内核处理功能。MIPS32 架构最多提供 4 个协处理器，分别命名为 CP0-CP3，功能如下表所示。

由于运行 ucore 操作系统不需要浮点运算，因此我们只需要实现 CP0。CP0 的详细功能如下。

1. **配置 CPU 工作状态：**改变 CPU 特性，比如大端表示和小端表示的切换。
2. **高速缓存控制：**用来控制读写缓存。

协处理器	作用
CP0	系统控制
CP1	浮点处理单元
CP1	特定实现
CP2	浮点处理单元

**Table 2.2:** MIPS32 协处理器功能表

3. **异常控制:** 异常发生时的检测和处理。
4. **存储管理单元控制:** 对系统的储存区域进行合理的控制、管理和分配, 主要是对 MMU、TLB 的一些配置、管理、访问。
5. **其他:** 不方便当作外设访问的额外的功能, 如时钟、时间计数器、奇偶校验错误检测。

CP0 包含 32 个 32 位寄存器, ucore 操作系统涉及其中 11 个, 其名称和功能列在下表。

寄存器号	寄存器名	功能
0	Index	TLB 阵列的入口索引
2	EntryLo0	偶数虚拟页入口地址的低 32 位部分
3	EntryLo1	奇数虚拟页入口地址的低 32 位部分
8	BadVAddr	记录最近一次存储发生异常时的虚拟地址
9	Count	与 Compare 寄存器组成片内计时器, 两者相等所示发出时钟中断信号
10	EntryHi	TLB 入口地址的高 32 位部分
11	Compare	与 Count 寄存器组成片内计时器, 两者相等所示发出时钟中断信号
12	Status	处理器状态和控制寄存器, 决定 CPU 特权等级和中断使能等
13	Cause	保存最近一次异常原因
14	EPC	保存最近一次异常的程序计数器
15	Ebase	保存异常处理程序的入口地址

**Table 2.3:** CP0 寄存器功能表

## 2.5 MMU

MMU(Memory Management Unit), 即内存管理单元, 是管理虚拟存储器、物理存储器的控制模块, 同时负责虚拟地址映射为物理地址, 以及提供硬件机制的内存访问授权, 多用户多进程操作系统。MIPS32 架构的 MMU 主要实现虚拟内存映射功能, 以读写控制信号和虚拟地址位输入, 实现对应物理地址数据的储存和访问。

## 2.6 TLB

旁路快表缓冲 (Translation Lookaside Buffer), 简称快表, 是一种地址变换高速缓存。虚拟地址和物理地址之间的对应关系使用页表储存, 故程序至少两次访问内存: 第一次获取物理地址, 第二

次获取物理地址中的数据。TLB 通过对最近使用的页表进行高速缓存以实现性能优化。CPU 访问虚拟地址时，会首先在 TLB 中查找，如果 TLB 中没有相应页表，则会出发页表缺失异常，并通过访问内存中的页表计算出相应的物理地址。与此同时，这个页表会存放在 TLB 中，之后对该地址的访问可以直接从 TLB 中获取物理地址，称为页表命中。

## 2.7 异常与中断处理

MIPS32 中的异常包括中断 (Interrupt)、陷阱 (Trap)、系统调用 (System Call) 以及其他任何可以打断正常执行流程的操作。下表列出了 MIPS 需要处理的异常类型及其优先级。

优先级	异常	描述
1	Reset	硬件复位
2	Soft Reset	发生致命错误后对系统进行软复位
5	NMI	不可屏蔽的中断
7	Interrupt	检测到 8 个中断之一
11	AdEL	取值地址对齐异常
12	TLB Refill	指令 TLB 缺失
13	TLB Invalid	指令 TLB 无效
16	Sys	执行系统调用指令 SYSCALL
16	RI	无效指令
16	Ov	算数操作指令 ADD、ADDI、SUB 运算溢出
19	AdEL	加载数据的地址未对齐
19	AdES	存储数据的地址未对齐
20	TLB Refill	数据 TLB 缺失
21	TLB Invalid	数据 TLB 无效
22	TLB Mod	对不可写的 TLB 进行了写操作

**Table 2.4:** MIPS32 异常类型和优先级表

检测到异常发生后，MIPS32 对异常处理的过程如下：

1. 检测 CP0 中 Status 寄存器的 EXL 字段。如果 EXL 为 0，将异常原因保存到 CP0 协处理器 Cause 寄存器的 ExcCode 字段。
2. 检查发生异常的指令是否在延迟槽中，如果在则设置 EPC 寄存器的值为指令地址减 4，Cause 寄存器 BD 字段为 1；否则设置 EPC 寄存器的值为指令地址，Cause 寄存器 BD 字段为 0。
3. 设置 Status 寄存器的 EXL 字段为 1，表示进入内核态处理异常，禁止中断。
4. 处理器根据异常种类，转移到相应异常处理例程的让入口地址，运行异常处理程序。
5. 处理结束，调用异常返回指令 ERET 转到异常发生前的状态。ERET 指令会清除 Status 寄存器的 EXL 字段，并且将 EPC 寄存器的值复制回 PC 中。



## 3. 流水线

### 3.1 流水线概述

流水线, 是一种实现多条指令重叠执行的技术。在 MIPS 的流水线实现中, 通常情况下, 一条 MIPS 指令被分为 5 个处理步骤

1. 从指令存储器中读取指令 (IF)
2. 指令译码并读取寄存器 (RD)
3. 算数运算 (ALU)
4. 从数据存储器中读取操作数 (MEM)
5. 将结果写回寄存器文件 (WB)

这样的流水线设计, 使得处理器在同一时刻最多能够处理 5 条 MIPS 指令, 在不缩短单条指令的处理时间的情况下, 提高了指令的吞吐率 (即单位时间内完成指令数目)。在理想情况下, 流水线上的指令执行时间变成非流水线指令执行时间的  $\frac{1}{5}$ 。

#### 3.1.1 流水线冒险

由于流水线中不同指令执行会互相重叠, 因此若两条指令之间存在依赖关系, 就必须单独考虑, 这即是“流水线冒险”。流水线冒险被分为了结构冒险, 数据冒险和控制冒险。

##### 结构冒险

由于硬件不支持多条指令在同一时钟周期执行导致的冒险。例如在单一寄存器多条指令同时访问该寄存器则会发生结构冒险。

\* 结构冒险可以直接采用增加硬件资源解决

##### 数据冒险

由于一条指令必须在另一条指令完成后才能开始而造成的冒险。例如下一条指令需要上一条指令写回寄存器文件的结果, 即下一条指令必须等到上条指令的 WB 阶段结束后才能开始自己的 RD 阶段, 这就浪费了时钟周期。

数据冒险解决一般采用如下两种方式

1. 旁路  
具体做法即是从内部寄存器中提前取出数据交给下一步指令使用
2. 流水线阻塞 (或称气泡)  
即在两条指令之间插入空指令 (nop), 或者重新安排指令执行顺序来消除相邻指令间的依赖

## 控制冒险

又称分支冒险，由于取到的指令不一定是所需要的指令（分支跳转）从而指令无法在预定时间周期内执行导致的冒险。常在分支跳转处出现，在一条跳转指令执行时，其下一条指令已经进入流水线，但跳转指令的结果可能不是代码中的下一条指令，因此下一条指令进入流水线的结果可能需要被清除。

\* 控制冒险一般使用预测技术来解决。最简单的预测方法即使一直认为分支总是按照某一种规律跳转。更成熟的预测方式则是动态分支预测，需要缓存的使用，通过以前的分支行为来预测现在的分支行为

### 3.1.2 流水线数据通路及控制

指令不断地顺序通过 5 级流水线，流水线各部分被流水线寄存器分开，流水线寄存器储存每个时钟周期中一条指令从前一级流水线到下一级流水线的全部数据。

流水线控制中，控制信号根据 5 级流水线分为 5 组，每一级流水线中设置相应的控制信号

#### 1. 取指令

读指令存储器信号，写 PC 信号（不同指令在这一步的控制信号是相同的）

#### 2. 指令译码/读寄存器文件

（不同指令在这一步控制信号相同）

#### 3. 指令执行/地址计算

由于用到 ALU，有操作类型的控制信号，结果寄存器的选择信号等

#### 4. 访问存储器

内存读写信号等

#### 5. 写回

写入寄存器/存储器的控制信号等

### 3.1.3 功能需求

流水线有着如下功能方面的需求

1. 将硬件按照 5 级流水线进行划分，建立数据通路
2. 各级流水线之间采用寄存器来保存流通的数据，每一个时钟周期传递一次数据
3. 实现立即数符号扩展单元，多路选择器等子部件，并整合到流水线数据通路中
4. 实现指令寄存器对于指令的解析

5. 由控制单元根据不同的指令，发出相应的控制信号，控制数据转发/选择，寄存器/存储器读写等
6. 实现数据冒险的检测，以及在检测出数据冒险时产生流水线阻塞或者使用数据旁路
7. 对于异常指令，实现在流水线中异常处理程序的调用，识别在延迟槽内的异常指令
8. 整合 TLB 等其他功能

## 4. 外存

### 4.1 SRAM

SRAM 即静态随机访问存储器，是一种广泛使用的存储器类型。由于它访问延迟低、速度快，并且容易与其它数字电路集成进一块芯片中，故常被用于实现片上的少量数据存储。现代计算机的 CPU 中，高速缓存 (Cache) 本质上就是 SRAM。其名称中的“静态”是指，写入数据后，不需要定期进行刷新操作就可以保持，这使得它相比于动态存储器更加易用，不需要复杂的内存控制器。然而 SRAM 也存在一些缺点，其存储原理决定了它存储密度较低，因而单块芯片的容量不能做得非常大。

THINKPAD-Cloud 采用了 SRAM 作为主存储器。板上共有 4 片高速异步 SRAM 芯片。每片 SRAM 存储容量为  $1024\text{K} \times 16\text{bit}$ ，即 16 位宽，1M 深度，容量 2MB。板上总共有 8MB 的 SRAM 存储空间。SRAM 芯片对外主要有地址线、数据线和读写使能等控制信号线。除了数据线是输入、输出双向信号外，其余都是输入信号。实验平台设计时，将 4 片 SRAM 分为两组，一组内有两片。组内的两片 SRAM 地址和读写控制信号连接到一起，数据线分开，这样相当于把两片 SRAM 拼接成一个 32 位宽的 SRAM。

### 4.2 Flash

Flash 存储器常常被称之为闪存，是目前主要的非易失性半导体存储器件。实验平台上提供一片 8MB 容量的 NOR Flash 作为非易失存储，可用于存储实验用的程序，比如监控程序等。在完成 CPU 设计实验过程中，需要反复修改硬件逻辑进行测试，这个过程中软件程序是固定不变的。此时就可以把程序写到 Flash 里面，这样每次把 CPU 设计写入 FPGA 后，CPU 就可以直接从 Flash 加载并运行程序，节省了调试的时间。Flash 的读取时序与 SRAM 相似，只是相比之下速度略慢；但 Flash 写入过程就较为复杂了，需要按照特定步骤发送命令和数据，还需要查询、等待 Flash 的状态，一般是软件、硬件配合完成。

## 5. 外围设备

### 5.1 串行接口

异步串口是一种低速 (每秒数百至数千字节) 的通信接口, 目前在个人计算机上已逐渐被淘汰。然而由于串口通信方式简单, 容易硬件实现, 目前大部分的嵌入式设备上仍然保留了串口。要实现基本的串口通信功能, 硬件上只需要两根信号线, 分别用于接收和发送数据。THINPAD 上有两种可选的串口使用方式, 一种直连串口, 另一种是外部串口控制器。使用直连串口时, 串口的接收、发送信号直接连到 FPGA 引脚上, 需要在 FPGA 中实现串口控制器。

当使用外部串口控制器时, 串行化和解串行化逻辑在 FPGA 外部芯片中实现, FPGA 使用 8 位并行数据接口直接收发数据。串口控制器的控制信号独立, 但数据信号与 BaseRAM 的数据线共享, 需要处理总线冲突问题。

在调试阶段, 我们将使用串口与外部计算机进行通信, 向计算机发送调试输出并接受计算机传回的调试命令。

### 5.2 DVI 图像输出接口

DVI 即数字视频接口, 是一种以数字信号传输图像数据的接口标准。单个 DVI 通道支持的最大图像格式为 1920×1200@60Hz, 红绿蓝 3 种颜色各 8 位时, 可以计算通道上的数据速率为 3.7Gbit/s, 这是相当高的信号速率。为了保证信号完整性, DVI 标准规定线缆上采用 TMDS(最小转换差分信号) 方式传输信号。该技术要求并行的图像数据通过移位寄存器转换成高速的串行信号, 再通过一对差分信号线传输。最终在接口上呈现为 4 对信号线, 分别是红绿蓝三种颜色的数据线和一对时钟线。

THINPAD 选用了专用的并行信号转 TMDS 芯片 TFP410, 放置于 DVI 插座和 FPGA 之间完成这一转换工作。我们需要在 FPGA 上编程输出并行的图像数据、同步信号和时钟信号, 经过芯片转换后即可输出标准的 TMDS, 从而驱动显示器显示图像。

由于 DVI 连接器的机械尺寸较大, 出于节省空间的目的, 实验板采用 HDMI 连接器替代了 DVI 连接器。HDMI 规范向下兼容 DVI 格式的信号, 因此将实验板直接连接至 HDMI 显示器, 或者通过转接线连接到 DVI 显示器, 都是可以正常工作的。

DVI 模块需要实现以下功能:

- 固定 DVI 显示器的分辨率和屏幕刷新率
- 支持显示所有可显示的共 95 个 ASCII 字符
- 支持输入光标的显示于移动
- 支持屏幕滚动和翻页控制 (Page Up/Down)

## 5.3 以太网口

THINKPAD 使用的网口是当今广泛使用的 100M 快速以太网接口，同时向下兼容 10M 以太网。与网口相连的网卡型号为 DM9000A，这是由 DAVICOM 公司生产的通用以太网卡芯片，常见于嵌入式设备当中。该芯片集成了以太网 Phy 与 MAC，处理器可通过总线接口读写片上缓冲区，从而收发以太网帧。网卡芯片与 FPGA 为并行接口，包括 16 位数据信号和若干控制信号，接口时序与 Flash 类似。可以考虑移植 TCP/IP 协议栈。

## 5.4 USB 接口

USB 即通用串行总线，是一种计算机上的外围通信接口标准。THINKPAD 平台上有一个标准的 Type-A 的 USB 接口，版本为 1.1，最高速率 12Mbit/s，可以满足键盘、鼠标等低速外设的需要。由于较高版本的 USB 外设可以向下兼容，市面上的 U 盘、读卡器等设备同样可以兼容 THINPAD。目标是使 THINPAD 支持 USB 键盘、U 盘读取和 USB 扩展串口。

实验平台上使用的是赛普拉斯公司生产的 SL811HS 型 USB 控制器，该控制器符合 USB1.1 规范，支持主、从两种工作模式。它和处理器之间采用 8 位的并行接口，时序与 Flash 类似，可以非常方便地由 FPGA 逻辑控制。控制器还内置了 256 字节的缓冲区，以避免大量数据传输时，由于处理器读取数据不及时导致的数据丢失。

实现的 USB 接口至少需要支持 PS2 键盘。

## 6. 工具链

### 6.1 硬件环境

1. 开发板:THINKPAD-Cloud
2. FPGA: Xilinx XC7A100T
3. SRAM: IS61WV102416 32 位 8M
4. Flash: 16bit 8M NOR

### 6.2 软件环境

1. EDA 软件: Xilinx Vivado HL WebPACk 2018.3
2. 编译时 OS: Windows10 x64
3. 支持运行 OS: ucore-thumips
4. 操作系统模拟器: qemu

### 6.3 调试工具

调试工具可以帮助测试硬件与程序是否正常运行，能够减少开发时间。调试工具至少需要完成以下功能：

- 通信: 通过串口进行开发板与 PC 进行通信
- 断电: 能够设置、查看、删除断点
- 能够启动程序、操作系统
- 能够查看寄存器、内存单元的内容

## 7. 附录

Text



# Bibliography