# 计算机组成原理实验报告

19373106 裴宝琦

## 一、CPU设计方案综述

本CPU为Verilog实现的流水线MIPS – CPU。

第一阶段：先构造出来不考虑转发暂停的流水线主控制器。

先后问题 发现jal和转发分开再处理会极度麻烦 所以直接用AT法。。

冒险表格：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Tnew | Tusers | Tusert |  |
| add | 1 | 1 | 1 |  |
| sub | 1 | 1 | 1 |  |
| ori | 1 | 1 | 3 |  |
| lw | 2 | 1 | 3 |  |
| sw |  | 1 | 2 |  |
| beq |  | 0 | 0 |  |
| lui | 1 | 1 | 3 |  |
| j |  |  |  |  |
| jal |  |  |  |  |
| jr |  |  |  |  |
| nop |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

这里把i型指令都调到3了 防止误判暂停

最后的项目总结：基本的模块都是对应相应的功能 主要是根据5个转发位点冒险来建立对应的MUX，我这里有的地方用了五级流水线的12345，而不是FDEMW，其实无区别，针对jal/jalr的值、R类指令ALUOUT值的转发，解决了先后顺序的问题，然后根据Tnew和Tuse来触发暂停，暂停操作包括封锁PC的变化、IF/ID(1级)的变化、清空ID/EX(2级)。

然后延迟槽的行为？有点迷惑

暂停最苛刻的情况 lw接beq 暂停两次

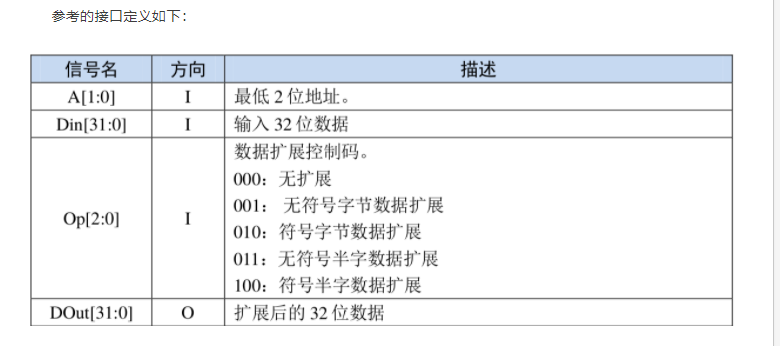
MULT、 MULTU、 DIV、 DIVU、MFHI、MFLO、MTHI、MTLO}

P6接P5 今天做简单的计算指令 ADD SUB SLL SLLV SRL SRA SRLV SRAV AND OR XOR NOR ADDI ADDIU ANDI XORI SLT SLTI SLTIU SLTU

1.增加BED BEDop LED 扩展DM

BEDop: 0:SW 1:SH 2:SB LB LH LBU





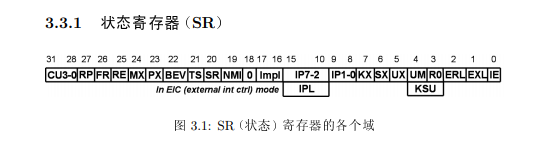
BE模块和LE模块

增加判断溢出的操作（等待完善）

P7：

第一个部分：

CP0



必须实现的寄存器：SR,CAUSE,EPC，prid（12 13 14 15）

SR：IM7-2：由CPU核外的信号产生，1-0为cause上可写的中断位

EXL：异常发生时置位 强行进入核心态禁止中断 决定CPU

IE：中断使能位 EXL禁止所有中断

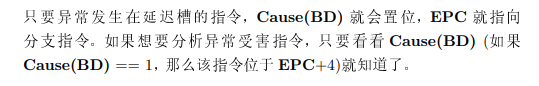
CAUSE：

IP7-2：等待决定的中断 照抄输入信号 1-0读写最近的值 告诉你现在发生了什么

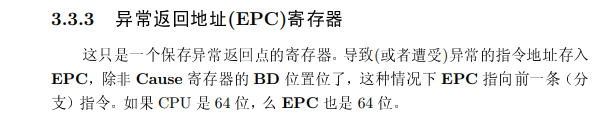
其他位0

EXCCODE:发生了哪种异常

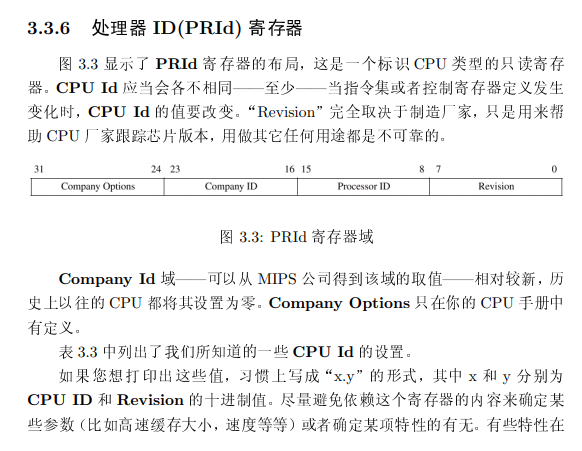
BD:分支延迟

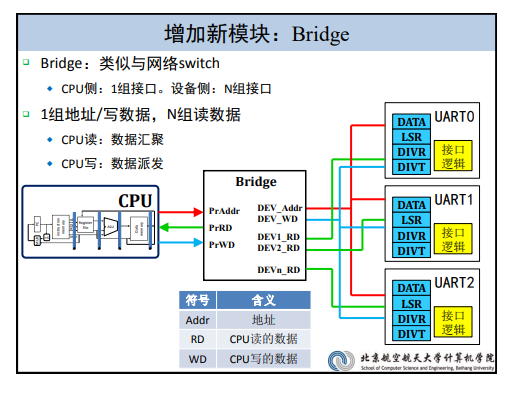


EPC：写入字对齐



PRID：





建模CP0

  我们计组课程一本参考书目标题中有“硬件/软件接口”接口字样，那么到底什么是“硬件/软件接口”？（Tips：什么是接口？和我们到现在为止所学的有什么联系？）

硬件接口：外设和CPU连接的接口，用来让两者产生联系

软件接口：用来接入软件来控制程序，让软件来控制设备

1. 在我们设计的流水线中，DM 处于 CPU 内部，请你考虑现代计算机中它的位置应该在何处。

在现在的计算机中DM应该都是通过CACHE和CPU进行数据交换

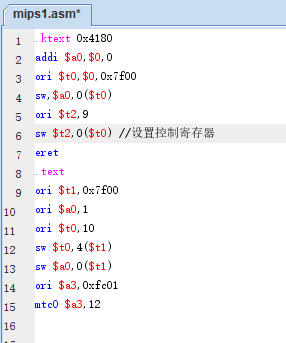
1. BE 部件对所有的外设都是必要的吗？

不是 比如这一次的TIMER寄存器，是读写整个寄存器，显然就不需要BE。

1. 请阅读官方提供的定时器源代码，阐述两种中断模式的异同，并分别针对每一种模式绘制状态转移图

见计时器设计文档

1. 请开发一个主程序以及定时器的exception handler。整个系统完成如下功能：
   * 1. 定时器在主程序中被初始化为模式0；
     2. 定时器倒计数至0产生中断；
     3. handler设置使能Enable为1从而再次启动定时器的计数器。2及3被无限重复。
     4. 主程序在初始化时将定时器初始化为模式0，设定初值寄存器的初值为某个值，如100或1000。（注意，主程序可能需要涉及对CP0.SR的编程，推荐阅读过后文后再进行。）



请查阅相关资料，说明鼠标和键盘的输入信号是如何被 CPU 知晓的？

设备包括接口控制器和设备主体，CPU并不是直接和外设联系，而是通过接口连接，当鼠标键盘输入时相当于中断，然后中断程序会把数据读到寄存器，从而获得信息。

处理过的一些bug：

1.暂停应该屏蔽异常中断否则会进入不了handler

2.MFC0考虑寄存器为0的情况

3.DM级不能乱写

4.宏观PC问题，延迟槽判断对应错了

5.没有考虑MTC0接ERET暂停。

6.lw计时器