**控制单元说明文档（CU – Control Unit）**

1. **功能说明**

1)、底层功能说明

控制单元用于控制流水段寄存器的阻塞（包括pc）、冲刷以及进行中断异常的处理。

定义说明：

阻塞：使得需要被阻塞的流水段寄存器的使能信号(enable)为0（无效）。

冲刷：使得需要被冲刷的流水段寄存器的复位信号(rst)为1（有效）。

中断异常：进行由CPU外部部件输入的中断，以及CPU内部的流水线中产生的异常的处理。

2)、实际实现功能说明

检测处理控制冒险、检测处理load-use冒险、进行异常及中断处理。

1. **Control unit单元检测说明**
2. 冒险：
   1. 控制冒险(转移指令branch引起)

由于转移指令branch 指令在MEM阶段才对PC进行写入，因此如果其转移的地址不是本身branch 指令的下一条指令地址，那么该转移指令之后的三个指令都是错误的，如下图所示。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| branch/jr | IF | ID | EX | MEM | WR |  |  |  |  |
| 延迟槽 |  | IF | ID | EX | MEM | WR |  |  |  |
| Error2 |  |  | IF | ID | EX | MEM | WR |  |  |
| Error3 |  |  |  | IF | ID | EX | MEM | WR |  |
| Ture |  |  |  | PC写入 | IF | ID | EX | MEM | WR |

处理方式：（仅进行静态分支预测，总判断是不跳转，错误时刻进行处理）

错误判断方式：在EX/MEM流水段寄存器中，EX/MEM.branch\_state == true || EX/MEM.j == true (分支条件满足 或 为直接跳转指令)，则进行处理。

处理方式：PC选择子选择EX/MEM.target\_address（实际图中是BRANCH\_ADDR的输出final\_target）作为输入地址，IF/ID冲刷、ID/EX冲刷（Reset = 1）（EX/MEM为延迟槽指令不进行处理）。

*注：jr指令由于可能存在数据冒险，如果是在ID段进行跳转，那么需要将EX/MEM/WR阶段的数据转发到ID段，这样和一般定义的转发逻辑不匹配，同时也过于麻烦，因此jr指令使用branch指令的数据通路（传输跳转地址），即jr也是在MEM阶段进行跳转，之前的指令会进行相应冲刷。*

* 1. load-use冒险（数据冒险）

由于load指令在MEM阶段才将数据从内存中加载出来，置入MEM/WR流水段寄存器，并在WR写入寄存器中。因此如果下一条指令使用需要load的寄存器，那么要使用的数据还未写入导致出错。同时MEM阶段刚读出的数据也无法进行转发。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Load | IF | ID | EX | MEM | WR |  |
|  |  | IF | ID | EX | MEM | WR |

处理方式：（在load与后面指令之间插入气泡）

Load-use冒险存在判断方式：ID/EX.MemRead == 1 and ((ID/EX.Rt == IF/ID.Rs) or (ID/EX.Rt == IF/ID.Rt))，(要读内存 且 下一指令一源寄存器是load指令的目的寄存器)

处理方式：PC使能端置0（无效），IF/ID阻塞（enable = 0），ID/EX冲刷（Reset = 1）。

1. 异常及中断处理

在Control Unit中，如果出现中断或者异常，那么输出中断异常处理程序地址，并传入PC选择数据输入端，并选择该异常中断处理程序地址。PC跳转到异常中断处理程序地址，开始进行异常中断的处理。

异常及中断检测处理过程详见“异常中断检测处理说明”部分。

1. 处理优先级说明

如果Control Unit出现了所有检测的问题，那么问题处理优先级需要进行定义，其优先级如下：**中断 > 异常 > 控制冒险（branch/jr） > load-use冒险**

优先级说明：

a)、异常中断处理优先级最高，因为如果该指令出现异常或者中断，那么该指令产生了错误，因此后面的指令不能执行，应当先进行错误的处理。

b)、对于转移指令引起的控制冒险与load-use冒险可能会产生冲突（同时发生），分为两种情况：

* 转移指令在load-use冒险之前，因为已经发生跳转显然load-use不应该执行，因此应该执行转移指令。
* load-use在转移指令之前，load-use在ID/EX检测，而转移在EX/MEM检测，因此此种情况不可能发生。

因此总结来看，两者同时出现时，执行转移指令，即转移指令优先级高于load-use冒险。

**三、接口模块说明**

Input:

一般接口：reset

load-use冒险处理：idex\_mem\_r、ifid\_rs\_addr、ifid\_real\_rt\_addr（来自ID阶段）、idex\_real\_rd\_addr

分支处理：id\_jmp、mem\_branch\_state（需要注意两者均具有延迟槽，因此jmp不冲刷，仅进行pc\_src选择；branch冲刷前2条指令）

中断异常：mem\_stall（存储器问题）、，mem\_exceptype（来自mem阶段中断检测部分）

|  |  |  |
| --- | --- | --- |
| reset(1) | CU | (1)cu\_pc\_stall |
|  | (1)cu\_ifid\_stall |
| id\_jmp(1) | (1)cu\_idex\_stall |
| mem\_jr(1) | (1)cu\_exmem\_stall |
| mem\_branch\_state(1) | (1)cu\_memwb\_stall |
|  |  |
| mem\_stall(1) | (1)cu\_ifid\_flush |
| mem\_excepttype（32） | (1)cu\_idex\_flush |
|  | (1)cu\_exmem\_flush |
|  |  |
| idex\_mem\_r（1） | (3)cu\_pc\_src |
| ifid\_rs\_addr（5） |  |
| ifid\_real\_rt\_addr（5） | (32)cu\_vector |

CP0框架图接口说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 接口名 | 宽度(bit) | 输入/输出 | 作 用 |
| 1 | reset | 1 | 输入 | 复位 |
| 2 | id\_jmp | 1 | 输入 | ID阶段，判断是否是j/jal指令。是则选择下一个pc为该地址。由于延迟槽的存在，无需冲刷if阶段刚取出的指令 |
| 3 | mem\_jr | 1 | 输入 | MEM阶段，判断是否是jr指令，是则选择从regfile中取出的指令地址 |
| 4 | mem\_branch\_state | 1 | 输入 | MEM阶段跳转部件，比较得出branch分支是否跳转，1为跳转，0为不跳转 |
| 5 | mem\_stall | 1 | 输入 | 由存储器送入的阻塞信号，cu对流水线进行阻塞等待存储器访问数据完毕 |
| 6 | mem\_excepttype | 32 | 输入 | 由MEM阶段的中断异常检测部件发送来的异常编码，用于判断流水段寄存器的处理，以及中断向量的输出 |
| 7 | idex\_mem\_r | 1 | 输入 | 用于判断是否是load指令 |
| 8 | ifid\_rs\_addr | 5 | 输入 | 在ID阶段指令Rs寄存器选择信号 |
| 9 | ifid\_real\_rt\_addr | 5 | 输入 | 在ID阶段指令真正的Rt寄存器选择值（有的指令指令Rt域是作为目标寄存器，而非源寄存器） |
| 10 | idex\_real\_rd\_addr | 5 | 输入 | 在EX阶段指令真正的Rd寄存器选择值，用于判断Load的目的寄存器是否与之前指令源寄存器相同 |
| 11 | cu\_pc\_stall | 1 | 输出 | pc的阻塞信号，原始意义为enable信号 |
| 12 | cu\_ifid\_stall | 1 | 输出 | IF/ID的阻塞信号，原始意义为enable信号 |
| 13 | cu\_idex\_stall | 1 | 输出 | ID/EX的阻塞信号，原始意义为enable信号 |
| 14 | cu\_exmem\_stall | 1 | 输出 | EX/MEM的阻塞信号，原始意义为enable信号 |
| 15 | cu\_memwb\_stall | 1 | 输出 | MEM/WR的阻塞信号，原始意义为enable信号 |
| 16 | cu\_ifid\_flush | 1 | 输出 | IF/ID的冲刷信号，原始意义为reset信号 |
| 17 | cu\_idex\_flush | 1 | 输出 | ID/EX的冲刷信号，原始意义为reset信号 |
| 18 | cu\_exmem\_flush | 1 | 输出 | EX/MEM的冲刷信号，原始意义为reset信号 |
| 19 | cu\_pc\_src | 3 | 输出 | 选择下一个pc值，  0 – 绝对跳转地址，j/jal指令  1 – 中断向量，中断异常跳转地址  2 – epc，eret指令  3 – 正确跳转地址，出现控制冒险(branch/jr)  4 – 正常pc+4 |
| 20 | cu\_vector | 32 | 输出 | 中断异常跳转地址 |