# MIPS设计文档

## cpu.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| PrAddr[31:2] | O | 32位地址总线 |
| PrRD[31:0] | I | 从Bridge读入的数据 |
| PrWD[31:0] | O | 输出至Bridge的数据 |
| PrWe | O | cpu写使能 |
| HWInt[7:2] | I | 6个硬件中断请求 |

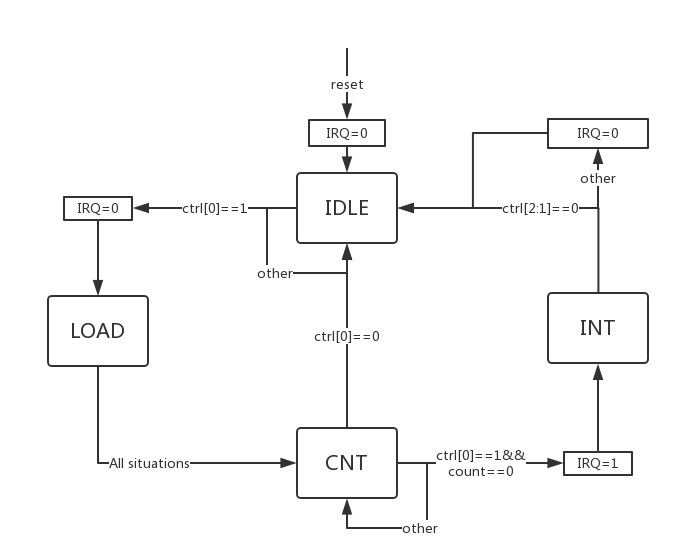
## timer.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| Addr[31:2] | O | 32位地址总线 |
| IRQ | O | 中断请求 |
| Rd[31:0] | I | 从Bridge读入的数据 |
| Wd[31:0] | O | 输出至Bridge的数据 |
| We | O | cpu写使能 |

表 1‑2 寄存器功能定义

|  |  |  |
| --- | --- | --- |
| 寄存器 | 偏移 | 描述 |
| ctrl | 0 | {28’b0, IM, Mode[2:1], Enable} |
| preset | 4 | 初值 |
| count | 8 | 计数器 |



## bridge.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| PrAddr[31:2] | I | cpu端地址 |
| PrRd[31:0] | O | 设备读出数据 |
| PrWD[31:0] | I | cpu写入数据 |
| PrWe | I | cpu写使能 |
| timer\_addr[1:0] | O | 设备地址 |
| timer1\_RD | I | 定时器1读出数据 |
| timer2\_RD | I | 定时器2读出数据 |
| timer\_WD[31:0] | O | 定时器写入数据 |
| timer1\_We | O | 定时器1写使能 |
| timer2\_We | O | 定时器2写使能 |

# CPU设计文档

## 数据通路设计

### 模块规格datapath.v

#### 端口说明

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| StallF | I |  |
| StallD | I |  |
| FlushE | I |  |
| Forward\_RS\_F[2:0] | I |  |
| Forward\_RS\_D[2:0] | I |  |
| Forward\_RT\_D[2:0] | I |  |
| Forward\_RS\_E[2:0] | I |  |
| Forward\_RT\_E[2:0] | I |  |
| Forward\_RT\_M[2:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@D[31:0] | O |  |
| HWInt[7:2] | I | 设备中断请求 |
| PrRD[31:0] | I | 设备读出数据 |
| PrAddr[31:2] | O | 设备地址 |
| PrWD[31:0] | O | 设备写入数据 |
| PrWe | O | 设备写使能 |

#### 功能定义

表 1‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| grf写入 | 若grf需要写入数据，则输出写入的位置及写入的值。 |
| dm写入 | 若dm需要写入数据，则输出写入的位置及写入的值。 |

### 取指

#### 模块规格F.v

表 2‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| NPC[31:0] | I |  |
| NPC\_Sel[1:0] | I | 下一指令地址控制信号 |
| Branch | I | 是否跳转 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| StallF | I | 暂停更新PC |
| StallD | I | 暂停更新D级流水线寄存器 |
| IR@D[31:0] | O | 当前指令 |
| PC4@D[31:0] | O |  |
| exp\_in | I | 进入异常 |
| iseretD | I | D级是eret指令 |
| EPC[31:0] | I |  |
| ExcCodeF[6:2] | O | F级异常类型 |
| bdD | O | F级指令是否是延迟槽指令 |

#### 程序计数器pc.v

表 2‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| NPC[31:0] | I | 下一指令地址 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| En | I | 使能信号 |
| PC[31:0] | O | 当前指令地址 |

表 2‑3 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 存储指令地址 | 输出当前指令地址。时钟上升沿到来，且使能端有效时保存下一指令地址 |
| 复位 | 复位信号有效时，PC同步复位于0x0000\_3000 |

#### 指令存储器im.v

表 2‑4 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[11:2] | I | 指令地址 |
| Instr[31:0] | O | 指令 |

表 2‑5 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 存储指令 | 初始化时，加载指令码。容量为32bit\*4096 |
| 输出指令 | 输出地址信号对应的指令内容 |

### 译码

#### 模块规格D.v

表 3‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| bdD | I | D级指令是否是延迟槽指令 |
| IR@D[31:0] | I | 当前指令 |
| PC4@D[31:0] | I | 下一指令地址 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| FlushE | I | 清空流水级寄存器 |
| Forward\_RS\_D\_Sel[2:0] | I | 转发MUX控制信号 |
| Forward\_RT\_D\_Sel[2:0] | I | 转发MUX控制信号 |
| PC4fromE[31:0] | I |  |
| AO[31:0] | I |  |
| PC4fromM[31:0] | I |  |
| MUX\_RF\_WD\_OUT[31:0] | I |  |
| Branch | O | 是否跳转 |
| NPC\_Sel[1:0] | O |  |
| NPC[31:0] | O | 跳转至地址 |
| MF\_RS\_D\_OUT[31:0] | O | D级读寄存器值 |
| IR@E[31:0] | O |  |
| PC4@E[31:0] | O |  |
| RS@E[31:0] | O |  |
| RT@E[31:0] | O |  |
| EXT@E[31:0] | O |  |
| MUX\_RF\_A3\_OUT[4:0] | I |  |
| We | I |  |
| exp\_in | I |  |
| ExcCodeF[6:2] | I |  |
| iseretD | O |  |
| isjrD | O |  |
| ExcCodeE[6:2] | O |  |
| bdE | O | E级指令是否是延迟槽指令 |

#### 寄存器文件grf.v

表 3‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A1[4:0] | I | 第一操作数地址 |
| A2[4:0] | I | 第二操作数地址 |
| A3[4:0] | I | 第三操作数地址 |
| Wd[31:0] | I | 待写入数据 |
| We | I | 写使能信号 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| Rd1[31:0] | O | 第一操作数 |
| Rd2[31:0] | O | 第二操作数 |

表 3‑3 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 复位 | 复位信号有效时，全部寄存器同步复位为0 |

#### 扩展单元ext.v

表 3‑4 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ExtOp[1:0] | I | 扩展功能 |
| Imm[15:0] | I | 待扩展立即数 |
| Ext[31:0] | O | 扩展结果 |

表 3‑5 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | ExtOp[1:0] | 描述 |
| {16’b0, Imm} | 00 | 无符号扩展 |
| {{16{Imm[15]}}, Imm} | 01 | 有符号扩展 |
| {Imm, 16’b0} | 10 | 扩展至低位 |
| 32’b0 | 11 | 复位 |

#### 指令自增单元npc.v

表 3‑6 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| PC4[31:0] | I |  |
| I26[25:0] | I | 指令的低26位 |
| NPCOp | I | 控制信号 |
| NPC[31:0] | O |  |

表 3‑7 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | NPCOp | 描述 |
| Branch | 0 | NPC = PC+4+{{14{I26[15]}}, I26[15:0], 2’b00} |
| Jump | 1 | NPC = {PC[31:28], I26[25:0], 2’b00} |

#### 比较单元cmp.v

表 3‑8 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Rd1[31:0] | I | 第一个操作数 |
| Rd2[31:0] | I | 第二个操作数 |
| CmpOp[2:0] | I | 指令类型 |
| Branch | O | 是否跳转 |

### 执行

#### 模块规格E.v

表 4‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@E[31:0] | I |  |
| bdE | I | E级指令是延迟槽指令 |
| PC4@E[31:0] | I |  |
| RS@E[31:0] | I |  |
| RT@E[31:0] | I |  |
| EXT@E[31:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@M[31:0] | O |  |
| PC4@M[31:0] | O |  |
| AO@M[31:0] | O |  |
| RT@M[31:0] | O |  |
| Forward\_RS\_E\_Sel[2:0] | I | 转发MUX控制信号 |
| Forward\_RT\_E\_Sel[2:0] | I | 转发MUX控制信号 |
| AO@M[31:0] | I |  |
| PC4fromM[31:0] | I |  |
| MUX\_RF\_WD\_OUT[31:0] | I |  |
| exp\_in | I |  |
| ExcCodeE[6:2] | I |  |
| ExcCodeM[6:2] | O |  |
| bdM | O | M级指令是流水线寄存器 |

#### 乘除单元mudi.v

表 4‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A[31:0] | I | A |
| B[31:0] | I | B |
| MuDiOp[2:0] | I | 运算类型 |
| start | I | 乘除运算开始信号 |
| We | I | 乘除单元写使能 |
| Din[31:0] | I | 待写入数据 |
| Clk | I |  |
| Reset | I |  |
| Dout[31:0] | O |  |
| buzy | O | 计算正在进行 |
| exp\_in | I | 进入中断 |

#### 计算单元alu.v

表 4‑3 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A[31:0] | I | A |
| B[31:0] | I | B |
| ALUOp[2:0] | I | 运算类型 |
| C[31:0] | O | C |
| N | O | (C < 0) |
| Z | O | (C == 0) |
| V | O | 运算溢出 |
| Carry | O | 运算进位 |

表 4‑4 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | ALUOp[2:0] | 描述 |
| Nop | 000 | 0 |
| Add | 001 | A+B |
| Sub | 010 | A-B |
| And | 011 | A&B |
| Or | 100 | A|B |
| Xor | 101 | A^B |
| Le | 110 | A<B |
| B | 111 | B |

### 访存

#### 模块规格M.v

表 5‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@M[31:0] | I |  |
| PC4@M[31:0] | I |  |
| AO@M[31:0] | I |  |
| RT@M[31:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@W[31:0] | O |  |
| PC4@W[31:0] | O |  |
| AO@W[31:0] | O |  |
| DR@W[31:0] | O |  |
| Forward\_RT\_M\_Sel[2:0] | I | 转发MUX控制信号 |
| MUX\_RF\_WD\_OUT[31:0] | I |  |
| D2write[31:0] | O |  |
| Dread\_exp[31:0] | I |  |
| exp\_in | I |  |
| ExcCodeM[6:2] | I |  |
| ExcCode[6:2] | O |  |
| PrRD[31:0] | I |  |
| PrAddr[31:2] | O |  |
| PrWD[31:0] | O |  |
| PrWe | O |  |

#### 字节使能be.v

表 5‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[31:0] | I | 数据地址 |
| BEOp[2:0] | I | 指令类型 |
| Be[3:0] | O | 字节使能 |
| ade | O | 地址不对齐 |

表 5‑3 功能定义

|  |  |  |
| --- | --- | --- |
| 指令类型 | 地址 | 字节使能 |
| byte | 00 | 0001 |
| 01 | 0010 |
| 10 | 0100 |
| 11 | 1000 |
| half | 00 | 0011 |
| 01 | 报错 |
| 10 | 1100 |
| 11 | 报错 |
| word | 00 | 1111 |
| 01-11 | 报错 |

#### 数据寄存器dm.v

表 5‑4 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[11:2] | I | 数据地址 |
| Din[31:0] | I | 待写入数据 |
| Be[3:0] | I | 字节使能 |
| We | I | 写使能信号 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| Dout[31:0] | O | 数据 |

表 5‑5 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 复位 | 复位信号有效时，dm同步复位于0x0000\_0000 |

### 写回

#### 模块规格W.v

表 6‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@W[31:0] | I |  |
| PC4@W[31:0] | I |  |
| AO@W[31:0] | I |  |
| DR@W[31:0] | I |  |
| MUX\_RF\_A3\_OUT[4:0] | O | 写回寄存器编号 |
| MUX\_RF\_WD\_OUT[31:0] | O | 写回数据 |
| RegWrite | O |  |

#### 数据扩展loadExt.v

表 6‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Dr[31:0] | I | 读出数据 |
| Addr[1:0] | I | 数据地址 |
| BEOp[1:0] | I | 数据扩展方式 |
| LoadExtOp | I | 是否有符号扩展 |
| Ext[31:0] | O | 扩展数据 |

### 功能多选器mux.v

#### MUX\_PC

表 7‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| add4[31:0] | I |  |
| NPC[31:0] | I |  |
| mf[31:0] | I |  |
| NPC\_Sel[1:0] | I |  |
| Branch | I | 是否跳转 |
| MUX\_PC\_OUT[31:0] | O |  |

表 7‑2 MUX\_PC功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | NPC\_Sel[1:0] | 数据来源 |
| add4 | 2’b00 |  |
| NPC | 2’b01 | NPC jump |
| NPC | 2’b10 | (Branch == 1) ? NPC branch : add4 |
| ra | 2’b11 | MF\_RS\_F |

#### MUX\_ALU\_B

表 7‑3 MUX\_ALU\_B功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | ALUSrc\_Sel | 数据来源 |
| Rd2[31:0] | 1’b0 | MF\_RT\_E |
| Ext[31:0] | 1’b1 | EXT@E |

#### MUX\_RF\_A3

表 7‑4 MUX\_RF\_A3功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | RFWA\_Sel[1:0] | 数据来源 |
| rd[4:0] | 2’b00 | IR@W[rd] |
| rt[4:0] | 2’b01 | IR@W[rt] |
| 31 | 2’b10 |  |

#### MUX\_RF\_WD

表 7‑5 MUX\_RF\_WD功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | RFWD\_Sel[1:0] | 数据来源 |
| C[31:0] | 2’b00 | AO@W |
| Dout[31:0] | 2’b01 | DR@W |
| PC4[31:0] | 2’b10 | PC8@W |

### 转发多选器mf.v

表 8‑1 功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | Forward\_Sel[2:0] | 数据来源 |
| PC4@E[31:0] | 3’b100 | PC4@E + 4 |
| AO[31:0] | 3’b011 | AO@M |
| PC4@M[31:0] | 3’b010 | PC4@M + 4 |
| Wd[31:0] | 3’b001 | MUX\_RF\_WD |
| Rd[31:0] | 3’b000 | RF.RD, RD@E, RD@M |

## 控制器设计

### 控制单元ctrl.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| NPC\_Sel@F[1:0] | O | 下一指令地址来源 |
| CmpOp@D[2:0] | O | 比较单元 |
| NPCOp@D | O | 指令跳转方式 |
| ExtOp@D[1:0] | O | 立即数扩展方式 |
| ALUASrc@E | O | ALU第一操作数来源 |
| ALUBSrc@E | O | ALU第二操作数来源 |
| ALUOp@E[2:0] | O | ALU运算方式 |
| MuDiOp[2:0] | O | 乘除单元运算方式 |
| MuDiWrite | O | 乘除单元写使能 |
| MuDiStart | O | 乘除单元开始运算信号 |
| AOSrc | O | E级运算结果来源 |
| BEOp[1:0] | O | 存取数据指令类型 |
| MemWrite@M | O | 数据存储器写使能信号 |
| RegDst@W[1:0] | O | 待回写寄存器地址来源 |
| LoadExtOp | O | 读出数据扩展方式 |
| DataSrc@W[1:0] | O | 待回写数据来源 |
| RegWrite@W | O | 寄存器写使能信号 |
| RI\_exp | O |  |
| iseret | O |  |
| iscp0 | O |  |
| cp0\_we | O |  |
| instr[31:0] | I |  |
| A1[5:0] | O | 第一个读寄存器编号，如不需要则为0 |
| A2[5:0] | O | 第二个读寄存器编号，如不需要则为0 |
| A3[5:0] | O | 写寄存器编号，如不需要则为0 |
| ic[3:0] | O | 指令类型 |

### 冲突单元hazard.v

表 2‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr[31:0] | I | D级流水线寄存器指令 |
| MuDiBusy | I | 乘除单元占用 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| StallF | O |  |
| StallD | O |  |
| FlushE | O |  |
| Forward\_RS\_D[2:0] | O |  |
| Forward\_RT\_D[2:0] | O |  |
| Forward\_RS\_E[2:0] | O |  |
| Forward\_RT\_E[2:0] | O |  |
| Forward\_RT\_M[2:0] | O |  |

## 中断控制器设计

### expF.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| MUX\_PC\_OUT[31:0] | I | 下一指令地址 |
| ExcCodeF[6:2] | O | 异常类型 |

表 1‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 判断AdELF | 若下一指令地址低两位非0，或地址超出范围，则报AdEL，直接传递至D级处理。否则输出默认值0b1111。 |

### expD.v

表 2‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ExcCodeF[6:2] | I | F级异常类型 |
| IRD[31:0] | I | D级指令 |
| ExcCodeE[6:2] | O | 异常类型 |

表 2‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 判断异常类型 | 若IRD不能识别，则报RI。否则传递ExcCodeF。 |

### expE.v

表 3‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ExcCodeE[6:2] | I | D级异常类型 |
| V | I | 运算溢出 |
| IRE[31:0] | I | E级指令 |
| ExcCodeM[6:2] | O | 异常类型 |

表 3‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 判断异常类型 | 若V为1，根据指令判断是否需要报异常，以及异常类型。如果不需要报异常，则传递ExcCodeD。 |

### expM.v

表 4‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ExcCodeM[6:2] | I | E级异常类型 |
| AOM[31:0] | I | 读取内存地址 |
| IRM[31:0] | I | M级指令 |
| ExcCode[6:2] | O | 异常类型 |

表 4‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 判断异常类型 | 若AOM低两位非0，则根据IRM判断是否需要报异常以及异常类型。若不需要，则传递ExcCodeE。 |

### exception.v

表 5‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Clk | I |  |
| Reset | I |  |
| bdM | I | M级是否为延迟槽指令 |
| IRM[31:0] | I | M级指令 |
| D2write[31:0] | I | M级待写入数据 |
| PC4M[31:0] | I |  |
| IRW[31:0] | I |  |
| ExcCode[6:2] | I |  |
| HWInt[7:2] | I |  |
| EXLSet | O |  |
| EPC[31:0] | O |  |
| Dread[31:0] | O |  |
| jr\_error\_addr[31:0] | I |  |

### cp0.v

表 6‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[4:0] | I | 读写cp0编号 |
| Din[31:0] | I | cp0写入数据 |
| bdM | I | M级是否为延迟槽指令 |
| PC4M[31:0] | I | M级pc4 |
| IRW[31:0] | I | W级指令 |
| ExcCodeM[6:2] | I | M级指令异常类型 |
| HWInt[15:10] | I | 6个设备中断 |
| We | I | cp0写使能（mtc0） |
| EXLSet | I |  |
| EXLClr | I | 清除EXL（eret） |
| Clk | I |  |
| Reset | I |  |
| IntReq | O | 中断请求 |
| ePC[31:2] | O | 输出EPC至NPC |
| exl | O |  |
| Dout[31:0] | O | cp0输出数据 |
| isjrM | I |  |
| jr\_error\_addr[31:0] | I |  |

表 6‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 写入Cause | 时钟上升沿到来时，写入IP |
| 进入中断 | 更新Cause寄存器和EPC寄存器，EXL置位，清空全部流水线，PC跳入异常处理程序 |
| 更新Cause | 若IRW是跳转/分支指令，则BD位置1。写入ExcCode |
| 更新EPC | 若IRW是跳转/分支指令，则写入pc4-8，否则写入pc4-4 |
| 退出中断 | 将EPC加载至pc，EXL清零，清空D级流水线 |

表 6‑3 寄存器功能定义

|  |  |  |  |
| --- | --- | --- | --- |
| 寄存器 | 编号 | 描述 | 读写 |
| SR | 12 |  | RW |
| CAUSE | 13 |  | R（软件保证） |
| EPC | 14 | 保存受害指令地址。如果是跳转/分支延迟槽指令异常，则指向跳转/分支指令。如果是jr异常，将跳转地址对齐后存入。 | RW |
| PrID | 15 |  | R（软件保证） |

表 6‑4 SR寄存器功能定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 31:16 | IM[7:2]([15:10]) | 9:2 | EXL | IE |
|  | 当前6个硬件哪些中断被允许 |  | 异常级 | 全局中断使能 |

表 6‑5 Cause寄存器功能定义

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BD | 30:16 | IP[7:2]([15:10]) | 9:7 | ExcCode[6:2] | | 1:0 |
|  |  | 当前6个硬件哪些中断正在有效 |  | 0 | Int |  |
|  |  |  |  | 4 | AdEL |  |
|  |  |  |  | 5 | AdES |  |
|  |  |  |  | 10 | RI |  |
|  |  |  |  | 12 | Ov |  |

## 测试程序

### 中断测试

.ktext 0x00004180

mfc0 $1, $12

mfc0 $1, $13

mfc0 $1, $14

add $1, $1, $t0

add $1, $1, $t0

mtc0 $1, $14

eret

.text

mtc0 $0, $12

mtc0 $0, $13

mtc0 $0, $14

ori $t0, $0, 4

lui $t1, 0x7fff

ori $t1, $t1, 0xffff

lui $t2, 0x8000

ori $t2, $t2, 0x0001

# timer

ori $a0, $0, 0x00000080

sw $a0, 0x00007f04

ori $a0, $0, 0x0000000b

sw $a0, 0x00007f00

sw $a0, 0x00007f08

nop

sw $a0, 0x00007f0c

nop

sw $a0, 0x00007f05

nop

lw $s0, 0x00007f00

lw $s1, 0x00007f04

lw $s2, 0x00007f08

lw $s3, 0x0000700c

lw $s4, 0x00007f10

lw $s5, 0x00007f14

lw $s6, 0x00007f18

lw $s7, 0x00007f1c

# ri

la $s1, 0x00000004

nop

lb $s1, 0x00000003

nop

beq $0, $0, mark1

sb $s1, 0x00000007

sh $s1, 0x00000012

mark1:

# ov

add $s0, $t1, $t1

nop

add $s1, $t2, $t2

nop

sub $s2, $t1, $t2

nop

sub $s3, $t2, $t1

nop

beq $0, $0, mark2

add $s0, $t1, $t1

mark2:

beq $0, $0, mark3

add $s1, $t2, $t2

mark3:

beq $0, $0, mark4

sub $s2, $t1, $t2

mark4:

beq $0, $0, mark5

sub $s3, $t2, $t1

mark5:

# adel\_e

lw $s0, -5($t2)

nop

lw $s0, 5($t1)

nop

beq $0, $0, mark6

lw $s0, -5($t2)

mark6:

beq $0, $0, mark7

lw $s0, 5($t1)

mark7:

# adel\_m

lw $s0, -2($t2)

nop

lw $s0, 3($t1)

nop

beq $0, $0, mark8

lw $s0, -2($t2)

mark8:

beq $0, $0, mark9

lw $s0, 3($t1)

mark9:

mfc0 $s0, $14

lw $t0, 0($s0)

nop

beq $0, $0, mark10

lw $t0, 0($s0)

mark10:

# ades\_e

sw $s0, -5($t2)

nop

sw $s0, 5($t1)

nop

beq $0, $0, mark11

sw $s0, -5($t2)

mark11:

beq $0, $0, mark12

sw $s0, 5($t1)

mark12:

# ades\_m

sw $s0, -2($t2)

nop

sw $s0, 3($t1)

nop

beq $0, $0, mark13

sw $s0, -2($t2)

mark13:

beq $0, $0, mark14

sw $s0, 3($t1)

mark14:

mfc0 $s0, $14

sw $t0, 0($s0)

nop

beq $0, $0, mark15

sw $t0, 0($s0)

mark15:

# adel\_f

ori $s1, $0, 0x00003011

jr $s1

nop

ori $s1, $0, 0x00002ffc

jr $s1

nop

### 延迟槽测试

.text

li $t0, 1

beq $t0, $0, mark1

sw $t1, 0x00000001

mark1:

nop

beq $0, $0, mark2

sw $t1, 0x00000001

mark2:

.ktext 0x00004180

mfc0 $k0, $13

ori $1, $k0, 0x80000000

mfc0 $k0, $14

addi $k0, $k0, 4

mtc0 $k0, $14

eret

# 思考题

## 欢迎来到玄学领域

硬件软件接口指计算机与程序设计过程中，软件（汇编程序）与硬件（电路）交互的领域。

## 沟通外部设备与计时器

DM应与外设在同一内存上，cpu通过片选信号加载数据。

不是，有些外设可能自身存储量小于1字节，不需要使能信号。

## 异常与中断

.ktext

sw $t0, 0($sp)

addi $sp, $sp, 4

sw $t1, 0($sp)

addi $sp, $sp, 4

lui $t0, 0xffff

ori $t0, $t0, 0xff83

mfc0 $t1, $12

ori $t1, $t1, 0xffffff83

bne $t1, $t0, not\_int

lw $t0, 0x00007f00

ori $t0, $t0, 1

sw $t0, 0x00007f00

subi $sp, $sp, 4

lw $t1, 0($sp)

subi $sp, $sp, 4

lw $t0, 0($sp)

eret

nop

not\_int:

.text

ori $t0, $0, 0x00000008

sw $t0, 0x00007f04

ori $t1, $0, 0x00000009

sw $t1, 0x00007f00

鼠标和键盘被操作后，数据被发送至内存，同时产生中断。中断处理程序识别中断类型然后从内存中读取相关数据。