P8 mips微系统设计文档

1. 处理器需求

流水线32位处理器，带有中断异常处理机制。

32位MIPS流水线CPU

IM: 8KB，使用FPGA内置IP核（具体什么是IP核，后文会有讲解）

DM: 8KB，使用FPGA内置IP核

支持MIPS-C5指令集 = { LB、LBU、LH、LHU、LW、SB、SH、SW、ADD、ADDU、SUB、SUBU、SLL、SRL、SRA、SLLV、SRLV、SRAV、 AND、OR、XOR、NOR、ADDI、ADDIU、ANDI、ORI、XORI、LUI、SLT、SLTI、 SLTIU、SLTU、BEQ、BNE、BLEZ、BGTZ、BLTZ、BGEZ、J、JAL、JALR、JR、ERET、MFC0、MTC0 }

1. 模块端口
2. mips

端口说明：

表1-mips端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |

功能定义：

表2-mips功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当复位信号有效时，所有子模块的reset复位有效 |
| 2 | 执行各子模块功能 | 执行各子模块功能 |

1. datapath

端口说明及功能定义：

表3-datapath端口说明及功能定义

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| times | I | 记录当前始终上升沿信号数 |
| ExtOp[1:0] | I | 扩展方式选择 10：立即数移至高16位，低位补零；  01：符号扩展；00：无符号扩展 |
| PC\_sel[1:0] | I | 回写PC的选择信号  10：RF\_RD1\_AM;  01：NPC  00：ADD4 |
| nPC\_sel | I | 选择下一条指令为B类指令或是J类指令  1：J类 NPC = imm26 0：B类NPC = PC4 + imm(EXT) |
| ALU\_A\_sel | I | 选择进入ALU进行运算的第一个参数 1：IR\_E[10:6]；0：V1\_E\_AMF |
| ALU\_B\_sel | I | 选择进入ALU进行运算的第二个参数 1：EXT32\_E；0：V2\_E\_AMF |
| ALUctr[3:0] | I | 运算选择信号： 000：加；001：减；010：或立即数； 其他情况备用未定义 |
| Mem\_to\_Reg\_sel[1:0] | I | 选择写寄存器值的写回路径  10：jal指令写回PC+4(PC8\_W)； 01：数据存储器DM写回（DR\_W）；  00：运算单元ALU写回（ALUOut\_W） |
| MemWrite | I | 数据存储器DM写使能信号，1：写入 |
| RegWrite | I | 寄存器堆RF写使能信号。1：写入 |
| S\_Instr[2:0] | I | Store类指令编号 |
| L\_Instr[2:0] | I | Load类指令编号 |
| A3\_in | I | M级传入控制器给出的A3写入地址 |
| PC\_En | I | PC冻结条件，1时暂停则PC使能关闭 |
| D\_En | I | D级寄存器冻结条件，1时暂停则使能关闭 |
| E\_reset | I | E级寄存器清空条件，1时暂停则清零 |
| MF\_CMP\_V1\_E\_RD1\_sel[2:0] | I | D级RF\_RD1的foward转发器选择信号 |
| MF\_CMP\_V2\_E\_RD2\_sel[2:0] | I | D级RF\_RD2的foward转发器选择信号 |
| MF\_ALU\_A\_sel[1:0] | I | E级ALU的SrcA端口的foward转发器选择信号 |
| MF\_ALU\_B\_V2\_M\_sel[1:0] | I | E级ALU的SrcB端口的foward转发器选择信号 |
| MF\_WD\_sel | I | M级DM写入数据WD的foward转发器选择信号 |
| Start | I | 启动乘除法单元信号 |
| Mul\_Div\_ctr[2:0] | I | 选择乘除单元的计算指令 |
| HiLo | I | 选择写回的hi lo寄存器 |
| Zero | O | ALU中比较SrcA与SrcB是否相等是否相等 1：相等；0：不等 |
| LT0 | O | ALU中比较SrcA与SrcB是否大于  1：是；0：否 |
| BG0 | O | ALU中比较SrcA与SrcB是否小于  1：是；0：否 |
| Busy | O | 乘除法单元是否在运行 |
| IR\_D[31:0] | O | 当前D级读取到PC下的32位指令码 |

* 1. multiplexor

端口说明：

表4-multiplexor端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| in0 | I | 0号输入信号 |
| in1 | I | 1号输入信号 |
| in2 | I | 2号输入信号 |
| in3 | I | 3号输入信号 |
| ... | I | ... |
| sel | O | 选择信号 |
| out | O | 输出信号 |

功能定义：

表5-multiplexor功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 多选器 | 根据sel信号  输出out = in(sel) |

实例化模块：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | in0 | in1 | in2 | in3 | in4 | in5 | sel | out |
| Mux5to1 MUX\_PC | ADD4 | NPC | RF\_RD1\_AM | `RI(0x00004018) | ERET |  | PC\_sel | N\_PC |
| Mux6to1 MF\_CMP\_V1\_E\_RD1 | RF\_RD1 | DR\_W | ALUOut\_W | ALUOut\_M | PC8\_W | PC8\_M | MF\_CMP\_V1\_E\_RD1\_sel | RF\_RD1\_AM |
| Mux6to1 MF\_CMP\_V2\_E\_RD2 | RF\_RD2 | DR\_W | ALUOut\_W | ALUOut\_M | PC8\_W | PC8\_M | MF\_CMP\_V2\_E\_RD2\_sel | RF\_RD2\_AM |
| Mux4to1 MF\_ALU\_A | V1\_E | DR\_W | ALUOut\_W | ALUOut\_M |  |  | MF\_ALU\_A\_sel | V1\_E\_AMF |
| MF\_ALU\_B\_V2\_M | V2\_E | DR\_W | ALUOut\_W | ALUOut\_M |  |  | MF\_ALU\_B\_V2\_M\_sel | V2\_E\_AMF |
| Mux2to1 MUX\_ALU\_B | V2\_E\_AMF | EXT32\_E |  |  |  |  | ALU\_B\_sel | V2\_E\_AM |
| Mux2to1 MF\_WD | V2\_M | DR\_W |  |  |  |  | MF\_WD\_sel | V2\_M\_AMF |
| Mux3to1 MUX\_Mem\_to\_Reg | ALUOut\_W | DR\_W | PC8\_W |  |  |  | Mem\_to\_Reg\_sel | RegData |
| Mux3to1  MUX\_MD\_ALU | ALUResult | LO | HI |  |  |  | HiLo | MD\_ALU\_out |

Mux3to1 MUX\_PC(ADD4, NPC, RF\_RD1\_AM, PC\_sel, N\_PC)

* 1. R\_Pipe

端口说明：

表6-R-Pipe端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| datain | I | 流水线待输入数据 |
| dataout | O | 流水线输出数据 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| En | I | 使能信号  1：可以寄存 0：不能寄存 |

功能定义：

表7-R\_Pipe功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 流水线寄存器 | 在时钟上升沿时，  若reset，则清零寄存器  若En，则进行数据流水 |

模块实例化：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R\_Pipe | datain | dataout | clk | reset | En |
| R\_IR\_D | Instr\_Out | IR\_D | clk | reset | D\_En |
| R\_PC4\_D | ADD4 | PC4\_D | clk | reset | D\_En |
| R\_PC8\_D | ADD8 | PC8\_D | clk | reset | D\_En |
| R\_Current\_PC\_D | Current\_PC | Current\_PC\_D | clk | reset | D\_En |
| R\_times\_D | times | times\_D | clk | reset | 1'b1 |
| R\_times\_E | times\_D | times\_E | clk | reset | 1'b1 |
| R\_times\_M | times\_E | times\_M | clk | reset | 1'b1 |
| R\_times\_W | times\_M | times\_W | clk | reset | 1'b1 |
| R\_ALUOut\_M | ALUResult | ALUOut\_M | clk | reset | `M\_En |
| R\_V2\_M | V2\_E\_AMF | V2\_M | clk | reset | `M\_En |
| R\_PC8\_M | PC8\_E | PC8\_M | clk | reset | `M\_En |
| R\_Current\_PC\_M | Current\_PC\_E | Current\_PC\_M | clk | reset | `M\_En |
| R\_ALUOut\_W | ALUOut\_M | ALUOut\_W | clk | reset | `W\_En |
| R\_DR\_W | DMOut | DR\_W | clk | reset | `W\_En |
| R\_PC8\_W | PC8\_M | PC8\_W | clk | reset | `W\_En |
| R\_A3\_W | A3\_in | A3\_W | clk | reset | `W\_En |
| R\_Current\_PC\_W | Current\_PC\_M | Current\_PC\_W | clk | reset | `W\_En |

* 1. E\_R\_Pipe

端口说明：

表8- E\_R\_Pipe端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| datain | I | 流水线待输入数据 |
| dataout | O | 流水线输出数据 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| E\_reset | I | E级寄存器复位信号（暂停时有效） 1：复位清零 0：正常工作 |
| En | I | 使能信号  1：可以寄存 0：不能寄存 |

功能定义：

表9- E\_R\_Pipe功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 流水线寄存器 | 在时钟上升沿时，  若reset，则清零寄存器  若E\_reset，则清零寄存器  若En，则进行数据流水 |

实例化模块：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E\_R\_Pipe | datain | dataout | clk | reset | E\_reset | En |
| R\_V1\_E | RF\_RD1\_AM | IR\_D | clk | reset | E\_reset | E\_En |
| R\_V2\_E | RF\_RD2\_AM | V2\_E | clk | reset | E\_reset | E\_En |
| R\_EXT32\_E | ExOut | EXT32\_E | clk | reset | E\_reset | E\_En |
| R\_PC8\_E | PC8\_D | PC8\_E | clk | reset | E\_reset | E\_En |
| R\_Current\_PC\_E | Current\_PC\_D | Current\_PC\_E | clk | reset | E\_reset | E\_En |

2.4 PC

端口说明：

表10-PC端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平 0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| PC\_En | I | PC冻结条件，1时暂停则PC使能关闭 |
| N\_PC[31:0] | I | 由多选器MUX\_PC传来的下一个PC地址 |
| Current\_PC | O | 经过PC的32位MIPS指令地址 |

功能定义：

表11- PC功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当复位信号有效时，PC内的索引地址被置为0x00003000 |
| 2 | 传递PC | 每个上升沿将N\_PC传入Current\_PC |

2.5 IM

端口说明：

表12-IM端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| Instr\_addr[31:0]（Current\_PC） | I | 当前指令索引PC |
| Instr\_Out[31:0] | O | 输出索引PC内的指令 |

功能定义：

表13-IM功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 取指令 | 根据当前PC值在IM中取出该地址中的指令 |

2.6 NPC

端口说明：

表14-NPC端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| imm(26)[25:0] | I | 存放jal/beq指令地址的26位立即数 |
| PC4 | I | D级PC4\_D的输入 |
| nPC\_sel | I | 控制NPC的输出信号  1：J类指令写回  0：B类指令写回 |
| NPC | O | 输出下一个传给PC寄存器的信号 |

功能定义：

表15-NPC功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 地址有符号扩展 | 将16（26）位立即数地址符号位扩展，并左移两位得到32位offset（jal跳转的地址） |
| 2 | 计算下一个指令的地址 | 控制NPC的输出，nPC\_sel:  1：J类指令写回  0：B类指令写回 |

2.7 RF

端口说明：

表16-RF端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| A1[4:0] | I | 当前周期使用rs域的寄存器地址 |
| A2[4:0] | I | 当前周期使用rt域的寄存器地址 |
| A3[4:0] | I | 当前周期写入RF的寄存器地址 |
| Current\_PC[31:0] | I | 当前D级指令32位索引PC地址 |
| RegData[31:0] | I | 32位待写入数据 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| times\_W[31:0] | I | W级当前时钟周期数 |
| RegWrite | I | 寄存器写使能信号，1时可以写入 |
| RD1[31:0] | O | 32位输出rs中数据 |
| RD2[31:0] | O | 32位输出rt中数据 |

功能定义：

表17-RF功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当复位信号有效时，寄存器堆所有寄存器的值被置为0x00000000 |
| 2 | 读寄存器 | 根据输入的A1A2读寄存器并向后输出 |

2.8 EXT

端口说明：

表18-EXT端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| imm16[15:0](IR\_D[15:0]) | I | 待扩展的16位数据 |
| ExtOp[1:0] | I | 扩展选择信号 01：符号扩展；00：无符号扩展；10：立即数移至高16位，低位补零 |
| ExOut[31:0] | O | 扩展后的32位数据 |

功能定义：

表19-EXT功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 符号扩展 | 符号扩展 |
| 2 | 无符号扩展 | 无符号扩展 |

2.9 CMP

端口说明：

表20-CMP端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| RD1(RF\_RD1\_AM) | I | RF的rs域对应寄存器的值的输出 |
| RD2(RF\_RD2\_AM) | I | RF的rt域对应寄存器的值的输出 |
| Zero(Zero) | O | ALU中比较SrcA与SrcB是否相等是否相等 1：相等；0：不等 |

功能定义：

表21-CMP功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 比较寄存器 | 比较RF输出的RD1与RD2经过转发后是否相等 |

2.10 ALU

端口说明：

表22-ALU端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| SrcA[31:0] (V1\_E\_AM) | I | 做加法运算的第一个32位数据 |
| SrcB[31:0] (V2\_E\_AM) | I | 做加法运算的第二个32位数据 |
| ALUctr[2:0] | I | 运算选择信号： 000：加；001：减；010：或立即数； 其他情况备用未定义 |
| ALUResult[31:0] | O | ALU运算后32位数据输出 |

功能定义：

表23-ALU功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加 | 计算ScrA+ScrB并输出 |
| 2 | 减 | 计算ScrA-ScrB并输出 |
| 3 | 或立即数 | 计算ScrA|ScrB并输出 |

2.11 BE

端口说明：

表24-BE端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| Byte\_in[1:0] (ALUOut\_M[1:0]) | I | 选择DM中地址的2位字节偏移 |
| S\_Instr[2:0] | I | 指示当前为何种s指令 |
| Byte\_out[3:0] | O | 4位偏移地址独热码输出 |

功能定义：

表25-BE功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 产生DM写入地址的字节使能信号 | 解析DM写入地址的字节偏移，产生对应使能信号 |

2.12 DM

端口说明：

表24-DM端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位置零；0：维持原状 |
| times\_M[31:0] | I | 当前M级时钟周期数 |
| Instr\_addr[31:0]（Current\_PC\_M） | I | 当前M级指令32位索引地址 |
| Byte\_sel[3:0] | I | 选择可以写入内存的字节偏移  Byte\_sel[3]：写入MemAddr[31:24]  Byte\_sel[2]：写入MemAddr[23:16]  Byte\_sel[1]：写入MemAddr[15:8]  Byte\_sel[0]：写入MemAddr[7:0] |
| MemWrite | I | 写使能信号 1：可写入 |
| MemAddr[31:0] | I | 待导出的32位数据的32位地址 |
| DataIn[31:0]( V2\_M\_AMF) | I | M级待写入内存的32位数据 |
| MemOut[31:0](DMOut) | O | 待从DM导出的32位数据 |

功能定义：

表25-DM功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 当复位信号有效时，内存中所有数据被置为0x00000000 |
| 2 | 读内存并输出 | 根据输入的地址结果读出对应位置的内存并输出 |
| 3 | 写入内存 | 根据输入的地址结果在相应位置写入数据 |

2.13 DM\_EXT

端口说明：

表24-DM\_EXT端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| DM\_EXT\_in[31:0]  (DR\_W[31:0]) | I | 当前DM字地址读出的32位数据，且经过W级寄存器 |
| Byte\_in[1:0]（ALUOut\_W[1:0]） | I | 当前DM字地址的末2位字节偏移，且经过W级寄存器 |
| DM\_EXT\_sel[2:0]  (L\_Instr[2:0]) | I | 选择可以写入寄存器的数据字段  000：无扩展（lw）  001：无符号字节数据扩展（lbu）  010：符号字节数据扩展(lb)  011：无符号半字数据扩展(lhu)  100：符号半字数据扩展(lh) |
| DM\_EXT\_out[31:0]  ( DM\_EXT\_out[31:0]) | O | 扩展后的DM的32位数据 |

功能定义：

表25-DM\_EXT功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 扩展DM中数据 | 根据DM\_EXT\_sel将从W级读出的DM中数据进行字节选择与符号/无符号扩展 |

2.14 MUL\_DIV

端口说明：

表22-MUL\_DIV端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| SrcA[31:0] (V1\_E\_AM) | I | 1.做乘除法运算的第一个32位数据  2.mthi/mtlo指令的写入数据 |
| SrcB[31:0] (V2\_E\_AM) | I | 做乘除法运算的第二个32位数据 |
| Mul\_Div\_ctr[2:0] | I | 运算选择信号：000：使能为零 001：mthi；010：mtlo  011：multu（无符号乘法）；100：mult（符号乘法）  101：divu（无符号除法）；110：div（符号除法） |
| Start | I | 通知MUL\_DIV产生busy信号(只有效一个周期)  1.启动 |
| Busy | O | 指示当前是否进行乘除类运算  1：暂停（执行）  0：不暂停（未执行） |
| HI[31:0] | O | 运算后高32位数据/余数32位输出 |
| LO[31:0] | O | 运算后低32位数据/商32位输出 |

功能定义：

表23-MUL-DIV功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 计算乘除法 | 计算ScrA\*ScrB、SrcA/SrcB、SrcA%SrcB并  存入Hi与Lo寄存器 |
| 2 | 计算mthi与mtlo | 计算mthi与mtlo  Hi <= Rs / Lo <= Rs |
| 3 | 产生乘除法暂停延迟 | Start启动Busy信号  并持续5（mult）/10（div）  周期 |

1. controller

端口说明及功能定义：

表26-controller端口说明及功能定义

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| Instr[31:0] | I | 数据通路传过来的当前D级读取到PC下的32位指令码 |
| Zero | I | 数据通路中ALU中比较SrcA与SrcB是否相等是否相等 1：相等；0：不等 |
| LT0 | I | ALU中比较SrcA与SrcB是否大于  1：是；0：否 |
| BG0 | I | ALU中比较SrcA与SrcB是否小于  1：是；0：否 |
| Busy | I | 乘除法单元是否在运行 |
| ExtOp[1:0] | O | 扩展方式选择 10：立即数移至高16位，低位补零；  01：符号扩展；00：无符号扩展 |
| PC\_sel[1:0] | O | 回写PC的选择信号  10：RF\_RD1\_AM;  01：NPC  00：ADD4 |
| nPC\_sel | O | 选择下一条指令为B类指令或是J类指令  1：J类 NPC = imm26 0：B类NPC = PC4 + imm(EXT) |
| ALU\_A\_sel | O | 选择进入ALU进行运算的第一个参数 1：IR\_E[10:6]；0：V1\_E\_AMF |
| ALU\_B\_sel\_E | O | (controller的E级传输)选择进入ALU进行运算的第二个参数 1：EXT32\_E；0：V2\_E\_AMF |
| ALUctr\_E[3:0] | O | (controller的E级传输)运算选择信号： 000：加；001：减；010：或立即数； 其他情况备用未定义 |
| Mem\_to\_Reg\_sel\_W[1:0] | O | (controller的W级传输)  选择写寄存器值的写回路径  10：jal指令写回PC+4(PC8\_W)； 01：数据存储器DM写回（DR\_W）；  00：运算单元ALU写回（ALUOut\_W） |
| MemWrite\_M | O | (controller的M级传输)  数据存储器DM写使能信号，1：写入 |
| RegWrite\_W | O | (controller的W级传输)  寄存器堆RF写使能信号。1：写入 |
| S\_Instr[2:0] | O | Store类指令编号 |
| L\_Instr[2:0] | O | Load类指令编号 |
| A3\_M | O | M级传给datapath的M级，datapath在W级写入的数据地址 |
| PC\_En | O | PC冻结条件，1时暂停则PC使能关闭 |
| D\_En | O | D级寄存器冻结条件，1时暂停则使能关闭 |
| E\_reset | O | E级寄存器清空条件，1时暂停则清零 |
| MF\_CMP\_V1\_E\_RD1\_sel[2:0] | O | D级RF\_RD1的foward转发器选择信号 |
| MF\_CMP\_V2\_E\_RD2\_sel[2:0] | O | D级RF\_RD2的foward转发器选择信号 |
| MF\_ALU\_A\_sel[1:0] | O | E级ALU的SrcA端口的foward转发器选择信号 |
| MF\_ALU\_B\_V2\_M\_sel[1:0] | O | E级ALU的SrcB端口的foward转发器选择信号 |
| MF\_WD\_sel | O | M级DM写入数据WD的foward转发器选择信号 |
| Busy\_start\_E | O | 启动乘除法单元信号 |
| Mul\_Div\_ctr\_E | O | 选择乘除单元的计算指令 |
| HiLo\_E | O | 选择写回的hi lo寄存器 |

* 1. main\_control

端口说明：

表27-main\_controller端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| Instr[31:0] | I | 数据通路传过来的当前D级读取到PC下的32位指令码 |
| Op[5:0] | I | 数据通路传过来的当前D级读取到PC下的32位指令码中[31:26] |
| Funct[5:0] | I | 数据通路传过来的当前D级读取到PC下的32位指令码中[5:0] |
| Rt[4:0] | I | 32位指令码rt域 |
| Zero | I | 数据通路中ALU中比较SrcA与SrcB是否相等是否相等 1：相等；0：不等 |
| LT0 | I | ALU中比较SrcA与SrcB是否大于  1：是；0：否 |
| BG0 | I | ALU中比较SrcA与SrcB是否小于  1：是；0：否 |
| ExtOp[1:0] | O | 扩展方式选择 10：立即数移至高16位，低位补零；  01：符号扩展；00：无符号扩展 |
| PC\_sel[1:0] | O | 回写PC的选择信号  10：RF\_RD1\_AM;  01：NPC  00：ADD4 |
| nPC\_sel | O | 选择下一条指令为B类指令或是J类指令  1：J类 NPC = imm26 0：B类NPC = PC4 + imm(EXT) |
| ALU\_A\_sel | O | 选择进入ALU进行运算的第一个参数 1：IR\_E[10:6]；0：V1\_E\_AMF |
| ALU\_B\_sel | O | (controller的E级传输)选择进入ALU进行运算的第二个参数 1：EXT32\_E；0：V2\_E\_AMF |
| ALUctr[3:0] | O | (controller的E级传输)运算选择信号： 000：加；001：减；010：或立即数； 其他情况备用未定义 |
| Mem\_to\_Reg\_sel[1:0] | O | (controller的W级传输)  选择写寄存器值的写回路径  10：jal指令写回PC+4(PC8\_W)； 01：数据存储器DM写回（DR\_W）；  00：运算单元ALU写回（ALUOut\_W） |
| A3\_sel[1:0] | O | comtroller传给A\_T\_coder的当前D级PC所指指令的写回RF的地址  10：31；  01：Rt[20:16]；  00：Rd[15:11]； |
| MemWrite | O | (controller的M级传输)  数据存储器DM写使能信号，1：写入 |
| RegWrite | O | (controller的W级传输)  寄存器堆RF写使能信号。1：写入 |
| S\_Instr[2:0] | O | Store类指令编号 |
| L\_Instr[2:0] | O | Load类指令编号 |
| M\_D\_Instr | O | mul,div类指令在D级信号 |
| Busy\_start | O | 启动乘除法单元信号 |
| Mul\_Div\_ctr | O | 选择乘除单元的计算指令 |
| HiLo | O | 选择写回的hi lo寄存器 |

功能定义：

表28- main\_controller功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 产生并传递各**流水部件功能信号**以及**功能多选器的信号** | 根据D级datapath传入的当前PC的Instr译码产生**流水部件功能信号、功能多选器的信号**以及**A3的选择信号** |

* 1. A\_T\_coder

端口说明：

表29-A\_T\_coder端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| Instr[31:0] | I | 数据通路传过来的当前D级读取到PC下的32位指令码 |
| A3\_sel[1:0] | I | comtroller传给A\_T\_coder的当前D级PC所指指令的写回RF的地址  10：31；  01：Rt[20:16]；  00：Rd[15:11]； |
| A1\_D[4:0] | O | IR\_D中rs[25:21] |
| A2\_D[4:0] | O | IR\_D中rt[20:16] |
| A3\_D[4:0] | O | IR\_D中经过选择的写回RF的寄存器地址 |
| Tuse\_RS0 | O | 当前指令在D级是否还需0拍就要用rs（A1）的值 |
| Tuse\_RS1 | O | 当前指令在D级是否还需1拍就要用rs（A1）的值 |
| Tuse\_RT0 | O | 当前指令在D级是否还需0拍就要用rt（A2）的值 |
| Tuse\_RT1 | O | 当前指令在D级是否还需1拍就要用rt（A2）的值 |
| Tuse\_RT2 | O | 当前指令在D级是否还2拍就要用rt（A2）的值 |
| Res | O | 当前指令产生新的寄存器值的位置  `PC：11（D级可用）；  `DM：10（W级可用）；  `ALU：01（M级可用）；  `NW：00 |

功能定义：

表30-A\_T\_coder功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 生成暂停、转发指示信号 | 产生Tuse和Tnew（Res）的暂停转发指示信号 |

* 1. hazard\_detector

端口说明：

表31-hazard\_detector端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| A1\_D[4:0] | I | IR\_D中rs[25:21] |
| A2\_D[4:0] | I | IR\_D中rt[20:16] |
| A3\_D[4:0] | I | IR\_D中经过选择的写回RF的寄存器地址 |
| Res\_E | I | E级流水线输出的Res的值 |
| Res\_M | I | M级流水线输出的Res的值 |
| Tuse\_RS0 | I | 当前指令在D级是否还需0拍就要用rs（A1）的值 |
| Tuse\_RS1 | I | 当前指令在D级是否还需1拍就要用rs（A1）的值 |
| Tuse\_RT0 | I | 当前指令在D级是否还需0拍就要用rt（A2）的值 |
| Tuse\_RT1 | I | 当前指令在D级是否还需1拍就要用rt（A2）的值 |
| Tuse\_RT2 | I | 当前指令在D级是否还2拍就要用rt（A2）的值 |
| Res [1:0] | I | 当前指令产生新的寄存器值的位置  `PC：11（D级可用）；  `DM：10（W级可用）；  `ALU：01（M级可用）；  `NW：00 |
| Busy | I | 乘除法单元是否在运行 |
| Busy\_start\_E | I | 启动乘除法单元信号 |
| M\_D\_Instr | I | mul,div类指令在D级信号 |
| Stall | O | 暂停信号输出 |

功能定义：

表32-hazard\_detector功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 生成暂停控制信号 | 由D级Tuse和E、M、W级Tnew（Res）的暂停转发指示信号通过T矩阵形式定位冲突暂停（地址）Stall\_RS0\_E1等等，再或起来生成Stall |

* 1. stall\_controller

端口说明：

表33- stall\_controller端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| Stall | I | 暂停信号输出 |
| PC\_En | O | PC冻结条件，1时暂停则PC使能关闭 |
| D\_En | O | D级寄存器冻结条件，1时暂停则使能关闭 |
| E\_reset | O | E级寄存器清空条件，1时暂停则清零 |

功能定义：

表34-stall\_controller功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 生成暂停信号 | 若Stall = 1  PC冻结  D级寄存器冻结  E级寄存器清空 |

* 1. forward\_controller

端口说明：

表35- forward\_controller端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| A1\_D[4:0] | I | IR\_D中rs[25:21] |
| A2\_D[4:0] | I | IR\_D中rt[20:16] |
| A1\_E[4:0] | I | IR\_E中rs[25:21] |
| A2\_E[4:0] | I | IR\_E中rt[20:16] |
| A2\_M[4:0] | I | IR\_M中rt[20:16] |
| A3\_M[4:0] | I | M级中经过选择的写回RF的寄存器地址 |
| A3\_W[4:0] | I | W级中经过选择的写回RF的寄存器地址 |
| Res\_M | I | M级流水线输出的Res的值 |
| Res\_W | I | W级流水线输出的Res的值 |
| MF\_CMP\_V1\_E\_RD1\_sel[2:0] | O | D级RF\_RD1的foward转发器选择信号  5：M级PC;  3：M级ALU  4：W级PC  2：W级ALU  1：W级DM  0：不转发 |
| MF\_CMP\_V2\_E\_RD2\_sel[2:0] | O | D级RF\_RD2的foward转发器选择信号  5：M级PC;  3：M级ALU  4：W级PC  2：W级ALU  1：W级DM  0：不转发 |
| MF\_ALU\_A\_sel[1:0] | O | E级ALU的SrcA端口的foward转发器选择信号  3：M级ALU  2：W级ALU  1：W级DM  0：不转发 |
| MF\_ALU\_B\_V2\_M\_sel[1:0] | O | E级ALU的SrcB端口的foward转发器选择信号  3：M级ALU  2：W级ALU  1：W级DM  0：不转发 |
| MF\_WD\_sel | O | M级DM写入数据WD的foward转发器选择信号  1：W级DM  0：不转发 |

功能定义：

表36-forward\_controller功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 生成转发信号 | 根据上表中输入信号进行转发部件与位置的判断并输出转发信号 |

4. Bridge

端口说明：

表22-Bridge端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| PrAddr[31:0]  (ALUOut\_M[31:0]]) | I | 待写入设备的32位选择地址  (HitTimer0 = PrAddr[31:3] == ‘h00007f0;  HitTimer1 = PrAddr[31:3] == ‘h00007f1;) |
| PrWD[31:0]  (V2\_M\_AMF) | I | 待写入设备的32位数据 |
| PrBE[3:0]  (Byte\_Sel) | I | 向写入设备输入的4位独热码使能(xxxx) |
| PrWe（流水线M级新产生） | I | 给设备终端的写使能信号 |
| PrRD[31:0]  (回到流水线M级与DMOut形成2to1) | O | 从选择设备读出的32位数据进入M级  (PrRD = HitTimer1 ? Timer0\_RD:  HitTimer0?Timer1\_RD:  FFFFFFFF) |
| HWInt[7:2]  (写回CP0寄存器) | O | 6中终端设备中断请求  1：请求中断 0：未中断 |
| Timer0IRQ | I | Timer0的中断请求 |
| Timer1IRQ | I | Timer1的中断请求 |
| Timer\_Addr[3:2] | O | 选择设备timer的内部地址 |
| Timer0\_RD[31:0] | I | Timer0的读出数据输入 |
| Timer1\_RD[31:0] | I | Timer1的读出数据输入 |
| Timer\_WD[31:0] | O | 写进Timer的32位数据 |
| Timer0We | O | Timer0的内部寄存器写使能  （Timer0We = PrWe & HitTimer0） |
| Timer1We | O | Timer1的内部寄存器写使能 |

功能定义：用于连接CPU与外部设备并进行数据与地址交互

5． Timer

端口说明：

表22-Timer端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| ADDR[3:2]  (Timer\_Addr[3:2]) | I | 选择设备timer的内部地址  00：CTRL  01：PRESET  10：COUNT |
| WE(Timer0/1We) | I | Timer的内部寄存器写使能 |
| DataIn[31:0]  (PrWD[31:0]) | I | 待写入设备的32位数据 |
| DataOut[31:0]  (Timer0 /1[31:0]) | O | 从选择设备读出的32位数据 |
| IRQ  (给HWInt[3]/[2]) | O | 终端设备中断请求  1：请求中断 0：未中断 |

功能定义：外部设备，用于产生两种类型的中断

6． CP0

端口说明：

表22-Timer端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| clk | I | 时钟信号  1：高电平；0：低电平 |
| reset | I | 复位信号 1：复位清零 0：正常工作 |
| A1[4:0]  （control中A2[4:0]） | I | 读CP0寄存器编号(MFC0) |
| A2[4:0]  （control中A3[4:0]） | I | 写CP0寄存器编号(MTC0) |
| PC[31:0]  （PC8\_M - 8） | I | 中断/异常时在M级的PC |
| Datain[31:0] | I | GRF写入CP0的32位数据(MTC0) |
| ExcCode[6:2]  （Exc\_M） | I | 中断/异常类型 |
| HWInt[5:0]  (Bridge HWInt[7:2]) | I | 6中终端设备中断请求  1：请求中断 0：未中断 |
| We(CP0\_We,根据M级IR译码或We流水至M级) | I | CP0寄存器写使能（MTC0） |
| EXLSet  (EI\_ctr) | I | 用于置位SR的EXL(EXL为1) |
| EXLClr  (control\_M) | I | 用于清除SR的EXL(EXL为0) |
| BD\_I | I | 记录当前指令是否为延迟槽指令（引入IR\_W进入M级，若IR\_W为j，jal，jalr，jr，beq，bne，blez，bltz，bgez，bgtz则RI\_control输出BD\_I为1） |
| IntReq  (EI\_ctr) | O | 中断请求，输出至CPU控制器 |
| EPC[31:0]  （nPC\_sel加一路） | O | EPC寄存器输出至NPC |
| DataOut[31:0]  (m\_to\_reg) | O | CP0中被选择的寄存器的输出数据 |

功能定义：用于处理中断和异常

7． EI\_ctr

端口说明：

表22-Timer端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向名 | 描述 |
| IntReq  (EI\_ctr) | I | 中断请求，输出至CPU控制器 |
| ExcReq  (exc\_occur流水产生) | I | 时钟信号  1：高电平；0：低电平 |
| IR\_M[31:0] | I | M级流水线32位指令值 |
| IR\_W[31:0] | I | W级流水线32位指令值,用于确定M级是否为延迟槽指令 |
| PC8\_D[31:0] | I | D级PC8的32位地址 |
| PC8\_E[31:0] | I | E级PC8的32位地址 |
| PC8\_M[31:0] | I | M级PC8的32位地址 |
| dp\_flush | O | 冲掉D\E\M\W各级流水线寄存器的值 |
| EXLSet  (CP0) | O | 用于置位SR的EXL(EXL为1) |
| EXLClr  (CP0) | O | 用于清除SR的EXL(EXL为0) |
| BD | O | 当前指令为延迟槽指令的指示 |
| PC\_EI | O | PC\_sel进入中断处理的信号 |
| PC\_ERET | O | PC\_sel返回正常程序的信号 |
| PCWrong | O | 传给CP0（PC）的PC出错地址 |

功能定义：中断与异常的控制信号的产生中枢

1. 控制信号真值表
2. main\_control(完整版见通路表)

四．思考题

**1.FPGA技术**

FPGA（Field－Programmable Gate Array），即现场可编程门阵列，它是在PAL、GAL、CPLD等可编程器件的基础上进一步发展的产物。它是作为专用集成电路（ASIC）领域中的一种半定制电路而出现的，既解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。

优点：

1）采用FPGA设计ASIC电路(专用集成电路)，用户不需要投片生产，就能得到合用的芯片。

2）FPGA可做其它全定制或半定制ASIC电路的中试样片。

3）FPGA内部有丰富的触发器和I/O引脚。

4）FPGA是ASIC电路中设计周期最短、开发费用最低、风险最小的器件之一。

5) FPGA采用高速CMOS工艺，功耗低，可以与CMOS、TTL电平兼容。

缺点：

1）掉电后一般会丢失原有逻辑配置

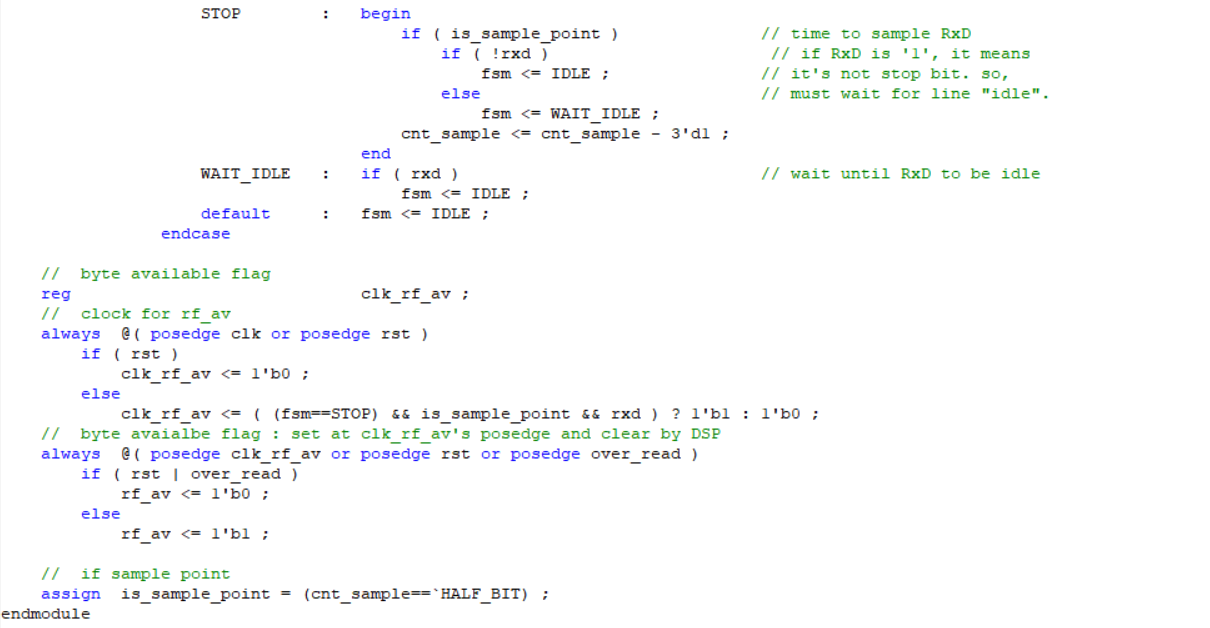
2）时序难规划

3）不能处理多事件

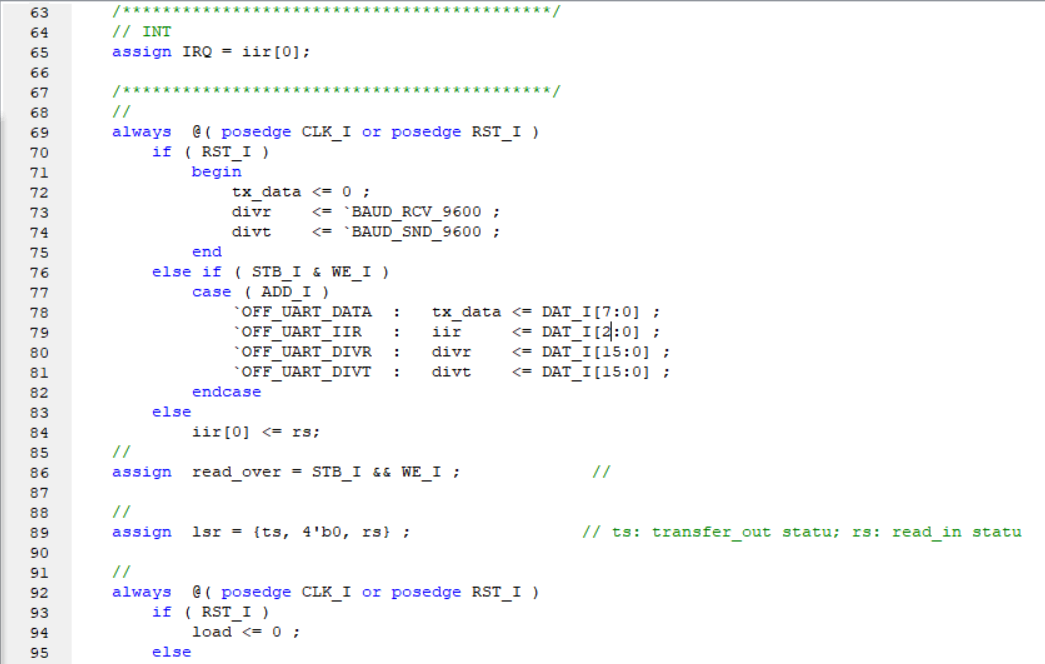
4）不适合条件操作

2.中断实现方案

根据要求，在UART接收到一次完整数据之后中断相应一次，那么根据UART原理以及所给的UART文件，应该从接收单元入手，在UART接收到一次完整的数据时，状态机进行转移，此时产生信号，根据此信号可设置相应的中断信号



[注]：rf\_av为所需信号



[注]：顶层如上所示

附：

数据通路图

