MIPS流水线处理器设计文档

MIPS单周期处理器设计文档由顶层模块、数据通路、控制器、冲突模块、测试程序组成。

1. 顶层模块mips.v

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 功能描述 |
| 1 | clk | 1 | I | pc、grf、dm的时钟信号 |
| 2 | reset | 1 | I | 当reset=1且时钟为上跳沿时，pc被设置为0x00003000，gpr所有寄存器置零，dm所有数据置零 |

顶层模块将数据通路模块datapath.v与控制器模块control.v,hazard.v相连。

1. 数据通路datapath.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | reset | 1 | I | 复位信号 |
| 2 | clk | 1 | I | 时钟信号 |
| 3 | PCOP | 2 | I | muxPCOP模块控制信号 |
| 4 | RegDst | 2 | I | muxRegDst模块控制信号 |
| 5 | ExtOP | 1 | I | Ext模块控制信号 |
| 6 | RegWrite | 1 | I | 寄存器文件写使能 |
| 7 | RegWData | 1 | I | muxRegWData模块控制信号 |
| 8 | ALUSrc | 1 | I | muxALUSrc模块控制信号 |
| 9 | ALUOP | 4 | I | 选择ALU运算类型的信号 |
| 10 | MemWrite | 1 | I | dm写入控制信号 |
| 11 | Instr\_original | 32 | O | IM输出指令 |
| 12 | Instr\_D | 32 | O | D级指令 |
| 13 | Instr\_E | 32 | O | E级指令 |
| 14 | Instr\_W | 32 | O | W级指令 |

datapath.v将pc.v, dm.v , im.v , add4.v,npc.v,alu.v , grf.v , ext.v , muxPCOP.v , muxRegWData.v , muxRegDst.v , muxALUSrc.v 等模块连接在一起。下面将介绍这些模块。

1. grf.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | RA1 | 5 | I | 读寄存器文件时第一个寄存器的下标 |
| 2 | RA2 | 5 | I | 读寄存器文件时第二个寄存器的下标 |
| 3 | Waddr | 5 | I | 写入文件的寄存器下标 |
| 4 | WData | 32 | I | 寄存器文件写入数据 |
| 5 | clk | 1 | I | 时钟信号 |
| 6 | reset | 1 | I | 复位信号 |
| 7 | RegWrite | 1 | I | 寄存器文件写使能 |
| 8 | RData1 | 32 | O | 读寄存器文件时第一个寄存器的输出 |
| 9 | RData2 | 32 | O | 读寄存器文件时第二个寄存器的输出 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读寄存器 | RData1输出RA1所寻址的寄存器的文件数据，RData2输出RA2所寻址的寄存器的文件数据 |
| 2 | 写寄存器 | 当时钟上升沿到来时，并且reset=0且RegWrite有效时，WD被写入WA所寻址的寄存器 |
| 3 | 复位 | reset=1 且 clk为上升沿时gpr所有寄存器置零 |

1. alu.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALU\_A | 32 | I | ALU的第一个操作数 |
| 2 | ALU\_B | 32 | I | ALU的第二个操作数 |
| 3 | ALUOP | 4 | I | 选择ALU运算类型的信号 |
| 4 | ALU\_C | 32 | O | ALU的运算结果 |
| 5 | equal | 1 | O | 判断ALU\_A与ALU\_B是否相等，若相等输出1，否则输出0 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加法 | 输出端ALU\_C=ALU\_A+ALU\_B |
| 2 | 减法 | 输出端ALU\_C=ALU\_A-ALU\_B |
| 3 | 或 | 输出端ALU\_C=ALU\_A|ALU\_B |
| 4 | 判断相等 | 若ALU\_A=ALU\_B，equal信号输出1，否则输出0 |

1. dm.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | addr | 32 | I | 写入数据储存器的数据的地址/读取数据储存器的数据的地址 |
| 2 | din | 32 | I | 写入数据储存器的数据 |
| 3 | MemWrite | 1 | I | 写入控制信号 |
| 4 | clk | 1 | I | 时钟信号 |
| 5 | reseet | 1 | I | 复位信号 |
| 6 | dout | 32 | O | 从数据存储器读取的数据 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数据储存器 | dout输出数据储存器在addr[11:2]地址的数据 |
| 2 | 写数据储存器 | 当MemWrite有效,reset=0,并且时钟为上升沿时，din被写入数据存储器地址为addr[11:2]的区域 |
| 3 | 复位 | reset=1并且时钟为上升沿时,DM所有数据置零 |

1. ext.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ext\_in | 16 | I | 位扩展输入信号 |
| 2 | EXTOP | 1 | I | 信号为1时，进行符号扩展，信号为0时，进行0扩展 |
| 3 | ext\_out | 32 | O | 位扩展输出信号 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 符号扩展 | 当EXTOP信号为1时，ext\_in符号扩展后由ext\_out输出 |
| 2 | 0扩展 | 当EXTOP信号为0时，在ext\_in前置16个0后由ext\_out输出 |

1. im.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | address | 32 | I | 指令地址 |
| 2 | instr | 32 | O | 指令 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 取指令 | 取出位于(address-0x00003000)[11:2]处指令 |

1. pc.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PC\_in | 32 | I | 下一条指令PC值 |
| 2 | clk | 1 | I | 时钟信号 |
| 3 | reset | 1 | I | 复位信号 |
| 4 | PC\_out | 32 | O | 当前PC值 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出指令地址 | 时钟上升沿且reset=0是PC输出指令地址 |
| 2 | 复位 | 时钟上升沿且reset=1，将PC置为0x00003000 |

1. muxPCOP.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PCOP | 2 | I | muxPCOP控制信号 |
| 2 | PC4 | 32 | I | PC+4 |
| 3 | NPC | 32 | I | NPC结果 |
| 4 | newPC | 32 | O | 下条指令地址 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出下条指令地址 | PCOP==2’b00:2'b00:newPC<=PC4;  PCOP==2’b01:2'b01:newPC<=NPC; |

1. muxRegDst.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | RegDst | 2 | I | muxRegDst控制信号 |
| 2 | IR\_W | 32 | I | W级指令 |
| 3 | WAddr | 5 | O | 写入文件的寄存器下标 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出写入文件的寄存器下标 | RegDst=2’b00，WAddr=rd  RegDst=2’b01，WAddr=rt |

1. muxRegWData.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALUC\_W | 32 | I | W级ALU结果 |
| 2 | DM\_W | 32 | I | W级从数据存储器读取的数据 |
| 3 | RegWData | 2 | I | muxRegWData的控制信号 |
| 4 | WData | 32 | O | 寄存器文件写入数据 |
| 5 | IR\_W | 32 | I | W级指令 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 输出寄存器文件写入数据 | RegWData=2’b00:WData<=ALUC\_W;  RegWData=2’b01:WData<=DM\_W; |

1. muxALUSrc.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | ALUSrc | 1 | I | muxAlUSrc的控制信号 |
| 2 | RD2 | 32 | I | 读寄存器文件时第二个寄存器的下标 |
| 3 | EXT\_out | 32 | I | 位扩展输出信号 |
| 4 | ALU\_B | 32 | O | ALU的第二个操作数 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 选择ALU的第二个操作数 | ALUSrc=0时：ALU\_B=RD2  ALUSrc=1时：ALU\_B=EXT\_out |

11、add4.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | PC | 32 | I | PC |
| 3 | PC4 | 32 | O | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | PC+4 | PC4=PC+4 |

1. IF\_ID.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_IM | 32 | I | IM输出指令 |
| 4 | PC4\_ADD4 | 32 | I | PC+4 |
| 5 | PC4\_D | 32 | O | PC+4 |
| 6 | IR\_D | 32 | O | D级指令 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | D级流水线寄存器 | 传递IR与数据 |

1. ID\_EX.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_D | 32 | I | D级指令 |
| 4 | RD1 | 32 | I | 读寄存器文件时第一个寄存器的输出 |
| 5 | RD2 | 32 | O | 读寄存器文件时二个寄存器的输出 |
| 6 | EXT\_out | 32 | O | D级指令 |
| 7 | PC4\_D | 32 | O | PC+4 |
| 8 | IR\_E | 32 | O | E级指令 |
| 9 | RD1\_E | 32 | O | 读寄存器文件时第一个寄存器的输出 |
| 10 | RD2\_E | 32 | O | 读寄存器文件时第二个寄存器的输出 |
| 11 | EXT\_E | 32 | O | EXT结果 |
| 12 | PC4\_E | 32 | O | PC+4 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | E级流水线寄存器 | 传递IR与数据 |

1. EX\_MEM.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_E | 32 | I | E级指令 |
| 4 | RD2\_E | 32 | I | 读寄存器文件时二个寄存器的输出 |
| 5 | ALU\_out | 32 | I | ALU结果 |
| 6 | PC4\_E | 32 | I | PC+4 |
| 7 | IR\_M | 32 | O | M级指令 |
| 8 | ALUC\_M | 32 | O | ALU结果 |
| 9 | PC4\_M | 32 | O | PC+4 |
| 10 | RD2\_M | 32 | O | 读寄存器文件时二个寄存器的输出 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | M级流水线寄存器 | 传递IR与数据 |

1. MEM\_WB.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | clk | 1 | I | 时钟信号 |
| 2 | reset | 1 | I | 复位信号 |
| 3 | IR\_M | 32 | I | M级指令 |
| 4 | DM\_Data | 32 | I | 从数据存储器读取的数据 |
| 5 | ALU\_M | 32 | I | ALU结果 |
| 6 | PC4\_M | 32 | I | PC+4 |
| 7 | IR\_W | 32 | O | W级指令 |
| 8 | ALUC\_W | 32 | O | ALU结果 |
| 9 | PC4\_W | 32 | O | PC+4 |
| 10 | DM\_W | 32 | O | 从数据存储器读取的数据 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | W级流水线寄存器 | 传递IR与数据 |

1. MF\_RS\_E.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RS\_E\_OP | 2 | I | MF\_RS\_E控制信号 |
| 2 | RD1\_E | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | RegWData | 32 | I | muxRegWData结果 |
| 5 | RD1 | 32 | O | 转发结果 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10时转发 |

1. MF\_RT\_E.v

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | MF\_RT\_E\_OP | 2 | I | MF\_RT\_E控制信号 |
| 2 | RD2\_E | 32 | I | 读寄存器文件时第二个寄存器的值 |
| 3 | ALUC\_M | 32 | I | ALU结果 |
| 4 | RegWData | 32 | I | muxRegWData结果 |
| 5 | RD2 | 32 | O | 转发结果 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 控制信号为01,10时转发 |

1. 控制器control.v

以下为控制信号产生的真值表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| func | 100001 | 100011 | 无 | | | |
| opcode | 000000 | 000000 | 001101 | 000010 | 001111 |  |
| 控制信号 | addu | subu | ori | j | lui | nop |
| PCOP | 00 | 00 | 00 | 01 | 00 | 0 |
| EXTOP | x | x | 0 | x | x | x |
| RegDst[1:0] | 00 | 00 | 01 | x | 01 | x |
| RegWrite | 1 | 1 | 1 | 0 | 1 | 0 |
| RegWData[1:0] | 00 | 00 | 00 | x | 10 | x |
| ALUSrc | 0 | 0 | 1 | x | x | x |
| ALUOP[3:0] | 0000 | 0001 | 0010 | x | x | x |
| MemWrite | 0 | 0 | 0 | 0 | 0 | 0 |

control.v信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | OP | 6 | I | OPcode |
| 2 | FUNC | 6 | I | Function code |
| 3 | PCOP | 2 | O | muxPCOP控制信号 |
| 4 | RegDst | 2 | O | muxRegDst控制信号 |
| 5 | ExtOP | 1 | O | ext控制信号 |
| 6 | RegWrite | 1 | O | gpr写入控制信号 |
| 7 | RegWData | 2 | O | muxRegWData的控制信号 |
| 8 | ALUSrc | 1 | O | muxAlUSrc的控制信号 |
| 9 | ALUOP | 4 | O | alu控制信号 |
| 10 | MemWrite | 1 | O | dm写入控制信号 |

1. 冲突模块

信号端口

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 信号 | 位数 | 方向 | 描述 |
| 1 | IR\_D | 1 | I | D级指令 |
| 2 | IR\_E | 1 | I | E级指令 |
| 3 | IR\_M | 32 | I | M级指令 |
| 4 | IR\_W | 32 | I | W级指令 |
| 5 | MF\_RT\_E\_OP | 32 | O | MF\_RT\_E控制信号 |
| 6 | MF\_RS\_E\_OP | 32 | O | MF\_RS\_E控制信号 |

模块功能

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 转发 | 输出MF\_RT\_E控制信号，输出MF\_RS\_E控制信号 |

1. 测试代码

测试期望：

$ 8 <= 00000042

$ 9 <= 00000058

$10 <= 0000009a

$11 <= 00000058

$12 <= 00000042

$13 <= ffffffbe

$14 <= ffffffea

$ 8 <= 00420000

$ 9 <= 00580000

$10 <= 009a0000

$11 <= 00580000

$12 <= 00420000

$13 <= ffbe0000

$14 <= ffea0000

$ 8 <= 00420000

$ 8 <= 00420000

$16 <= 00420000

$17 <= ffbe0000

$ 8 <= 00000000

$ 9 <= 00000000

$ 9 <= 00000000

$ 8 <= 00000042

$ 9 <= 00000058

$10 <= 0000009a

$11 <= 00000058

$12 <= ffffffbe

$13 <= 00000042

$14 <= 0000009a

$ 8 <= 00420000

$ 9 <= 00580000

$10 <= 009a0000

$11 <= 00580000

$12 <= ffbe0000

$13 <= 00420000

$14 <= 009a0000

$17 <= ffbe0000

ori $8,$0,0x0042

ori $9,$0,0x0058

addu $10,$8,$9

addu $11,$0,$9

subu $12,$10,$11

subu $13,$11,$10

subu $14,$12,$11

lui $8,0x0042

lui $9,0x0058

addu $10,$8,$9

addu $11,$0,$9

subu $12,$10,$11

subu $13,$11,$10

subu $14,$12,$11

j label

lui $8,0x0042

lui $9,0x0058

label:

lui $8,0x0042

subu $16,$10,$11

subu $17,$11,$10

addu $8,$16,$17

addu $9,$17,$16

addu $9,$17,$16

ori $8,$0,0x0042

ori $9,$0,0x0058

addu $10,$9,$8

addu $11,$9,$0

subu $12,$11,$10

subu $13,$10,$11

subu $14,$11,$12

lui $8,0x0042

lui $9,0x0058

addu $10,$8,$9

addu $11,$0,$9

subu $12,$11,$10

subu $13,$10,$11

subu $14,$11,$12

ori $0,$0,0x0042

addu $17,$0,$17

34080042

34090058

01095021

00095821

014b6023

016a6823

018b7023

3c080042

3c090058

01095021

00095821

014b6023

016a6823

018b7023

08000c11

3c080042

3c090058

3c080042

014b8023

016a8823

02114021

02304821

02304821

34080042

34090058

01285021

01205821

016a6023

014b6823

016c7023

3c080042

3c090058

01095021

00095821

016a6023

014b6823

016c7023

34000042

00118821

六、思考题

令CAL\_R={addu,subu},CAL\_I={lui,ori}

1. E级为CAL\_R，M/W级为CAL\_R，E级RS等于M/W级RD

优先转发M级ALUC\_M，再转发W级WData

样例：addu $1,$2,$3

addu $2,$1,$3

或

addu $1,$2,$3

无关指令

addu $2,$1,$3

1. E级为CAL\_R，M/W级为CAL\_R，E级RT等于M/W级RD

优先转发M级ALUC\_M，再转发W级WData

样例：addu $1,$2,$3

addu $2,$3,$1

或

addu $1,$2,$3

无关指令

addu $2,$3,$1

1. E级为CAL\_R，M/W级为CAL\_I，E级RS等于M/W级RT

优先转发M级ALUC\_M，再转发W级WData

样例：ori $1,$2,0x0001

addu $2,$1,$3

或

ori $1,$2,0x0001

无关指令

addu $2,$1,$3

1. E级为CAL\_R，M/W级为CAL\_I，E级RT等于M/W级RT

优先转发M级ALUC\_M，再转发W级WData

样例：ori $1,$2,0x0001

addu $2,$3,$1

或

ori $1,$2,0x0001

无关指令

addu $2,$3,$1

5、E级为CAL\_I，M/W级为CAL\_R，E级RT等于M级RD

优先转发M级ALUC\_M，再转发W级WData

样例：addu $1,$2,$3

ori $2,$1,0x00001

或

addu $1,$2,$3

无关指令

ori $2,$1,0x00001

6、E级为CAL\_I，M/W级为CAL\_I，E级RT等于M级RT

优先转发M级ALUC\_M，再转发W级WData

样例：lui $1,0x00002

ori $2,$1,0x00001

或

lui $1,0x00002

无关指令

ori $2,$1,0x00001