P3单周期CPU设计文档

1. 模块规格
   1. IFU：内部包括PC，IM和相关逻辑。

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| BrOffset[15:0] | I | Beq指令中的偏移量 |
| Reset | I | 复位有效时，将PC清零 |
| IF\_BR | I | 是否为分支指令如beq |
| ALU\_0 | I | ALU结果是否为0的信号，1表示结果为0 |
| Clk | I | 全局时钟信号（由顶层统一接入时钟） |
| Instruct[31:0] | O | 输出当前指令 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 加载指令 | 初始化时通过加载raw文件将指令读取到IM中 |
| 2 | 读出指令 | 通过PC输出的地址输出IM中相应指令 |
| 3 | 计算NPC（下一条指令地址） | 用当前地址PC和相关信号计算NPC |

注：PC起始地址为0x00000000；

IM容量为32字且ROM寻址方式为字地址故而取PC[6:2]

* 1. GRF（通用寄存器组）

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A1[4:0] | I | 准备进行读取的第一个寄存器编号 |
| A2[4:0] | I | 准备进行读取的第二个寄存器编号 |
| A3[4:0] | I | 准备写入的寄存器编号 |
| WD[31:0] | I | 准备写入寄存器的32位数据 |
| RD1[31:0] | O | 读取的第一个寄存器中的数据 |
| RD2[31:0] | O | 读取的第二个寄存器中的数据 |
| RegWr | I | 寄存器写入使能 |
| Clk | I | 全局时钟信号 |
| Reset | I | 复位有效时，将所有寄存器清零 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 读取数据 | 输出指定编号（A1,A2）寄存器的内容 |
| 2 | 写入数据 | 将来自于WD的数据写入指定编号（A3）寄存器 |
| 3 | 存储数据 | 将数据保存在寄存器当中即使断电也不会清零 |

注：寄存器总数为 32 个；

0 号寄存器的值保持为 0。

* 1. ALU（算术逻辑单元）:

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A[31:0] | I | ALU进行运算的第一个操作数 |
| B[31:0] | I | ALU进行运算的第二个操作数 |
| Eq | O | 判断两个输入是否相等，相等则输出1 |
| ALUop[1:0] | I | ALU功能选择信号，决定输出哪一个运算结果 |
| Out[31:0] | O | 运算结果输出 |

ALUop信号定义如下

|  |  |
| --- | --- |
| ALUop | Func |
| 00 | 加法 |
| 01 | 减法 |
| 10 | 比较大小，当且仅当A>B时输出1 |
| 11 | 按位或运算 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 加 | 执行A+B |
| 2 | 减 | 执行A-B |
| 3 | OR | 按位计算A or B |
| 4 | 比较大小 | 当且仅当A > B 输出1 |
| 5 | 判断相等关系 | 当且仅当A = B 输出1 |

1. DM（数据存储器）

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| WrData[31:0] | I | 将写入内存的数据 |
| Ad[4:0] | I | 准备进行读写的内存地址 |
| DMWr | I | 内存写使能 |
| DM[31:0] | O | 输出指定地址中存放的数据 |
| Clk | I | 全局时钟信号 |
| Reset | I | 复位有效时，将Mem清零 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 读取 | 按照指定地址输出存储的数据 |
| 2 | 写入 | 将输入数据写入指定地址内存中 |

注：内存起始地址：0x00000000。

1. EXT：

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| In[15:0] | I | 输入16位带扩展数字 |
| EXT\_ZS | I | 扩展功能选择 |
| EXT[31:0] | O | 输出扩展结果 |

EXT\_ZS信号定义如下

|  |  |
| --- | --- |
| EXT\_ZS | Func |
| 0 | 零扩展 |
| 1 | 符号扩展 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 零扩展 | 将输入数据扩展为32位并用0填充高16位 |
| 2 | 符号扩展 | 将输入数据扩展为32位并用原数据符号位填充高16位 |

1. SH（移位器）

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| In[15:0] | I | 输入十六位待移位数据 |
| Dis[4:0] | I | 移位距离 |
| SH\_LR | I | 控制移位方向 |
| SH[31:0] | O | 移位结果 |

功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能定义 | 功能描述 |
| 1 | 16位左移 | 将输入数据零扩展为32位后左移指定位数 |
| 2 | 16位右移 | 将输入数据零扩展为32位后右移指定位数 |

1. 控制器设计
   1. 首先列出各个模块需要的输入信号来源。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | NPC | PC | IM | GRF | | ALU | | DM | |
| WR | A3 | A | B | DATA | ADDR |
| Addu |  | NPC.PC4 | PC | ALU | IM[15:11] | RD1 | RD2 |  |  |
| Subu |  | NPC.PC4 | PC | ALU | IM[15:11] | RD1 | RD2 |  |  |
| Ori |  | NPC.PC4 | PC | ALU | IM[20:16] | RD1 | EXT.Z |  |  |
| Lw |  | NPC.PC4 | PC | DM | IM[20:16] | RD1 | EXT.S |  | ALU |
| Sw |  | NPC.PC4 | PC |  |  | RD1 | EXT.S | RD2 | ALU |
| Beq | ALU\_0  S\_EXT | NPC.NPC  /PC4 | PC |  |  | RD1 | RD2 |  |  |
| lui |  | NPC.PC4 | PC | SH | IM[20:16] |  |  |  |  |
| nop |  | NPC.PC4 | PC |  |  |  |  |  |  |
| 合并 | ALU\_0  S\_EXT | NPC.NPC  /PC4 | PC | ALU/  DM/SH | IM[15:11]/  IM[20:16] | RD1 | RD2/  EXT.S/  EXT.Z | RD2 | ALU |

* 1. 汇总后为具有多个输入的模块添加选择信号确定输入来源，并根据指令做出真值表确定其他命令信号

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FUNC | 100001 | 100011 | N/A | | | | | |
| OP | 000000 | 000000 | 001101 | 100011 | 101011 | 000100 | 001111 | 000000 |
| 控制  信号 | addu | subu | ori | lw | sw | beq | lui | nop |
| IF\_J | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| IF\_Br | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| RegDst | 0 | 0 | 1 | 1 | X | X | 1 | X |
| gpf\_WDsrc | 00 | 00 | 00 | 01 | X | X | 10 | X |
| ALUsrc | 0 | 0 | 1 | 1 | 1 | 0 | X | X |
| EXT\_ZS | X | X | 0 | 1 | 1 | 1 | X | X |
| SH\_LR | X | X | X | X | X | X | 0 | X |
| RegWr | 1 | 1 | 1 | 1 | 0 | 0 | 1 | X |
| MemWr | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| ALUop | 00 | 01 | 11 | 00 | 00 | 10 | X | X |

|  |  |
| --- | --- |
| **ALUop** | **Func** |
| 00 | 加法 |
| 01 | 减法 |
| 10 | 比较大小，当且仅当A>B时输出1 |
| 11 | 按位或运算 |

* 1. 具体参数

端口定义

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| **Funct[5:0]** | **I** | 当前指令中的Func字段 |
| **Op[5:0]** | **I** | 当前指令中的操作码字段 |
| **IF\_J** | **O** | 当前指令为J型时输出1 |
| **IF\_Br** | **O** | 当前指令为beq时输出1 |
| **RegDst** | **O** | 控制数据存储在GRF中的位置，0取rd字段作为地址，1取rt字段 |
| **WDsrc[1:0]** | **O** | GRF写入数据来源 |
| **ALUsrc** | **O** | ALU操作数来源，0为RD2，1为EXT |
| **EXT\_ZS** | **O** | 选择扩展器进行零扩展（0）或者符号扩展（1） |
| **SH\_LR** | **O** | 移位器移位方向选择（左0 右1） |
| **RegWr** | **O** | 寄存器堆写使能 |
| **MemWr** | **O** | 内存写使能 |
| **ALUop[1:0]** | **O** | **ALU功能选择信号** |

|  |  |
| --- | --- |
| **WDsrc** | **Src** |
| **00** | **ALU计算结果** |
| **01** | **Mem**（内存） |
| **10** | **SH（移位器）** |
| **11** | 未定义 |

1. 测试程序

.text

nop

ori $s1, $0 ,0x4

addu $s0, $s1, $0

li $t0, 0x12345678

lui $t1,0x00001234

sw $t0, 4($s0)

nop

lw $t2, 4($s0)

subu $t2, $t2, $t1

nop

beq $0, $0, 0xffff

00000000

34110004

02208021

3c011234

34285678

3c091234

ae080004

00000000

8e0a0004

01495023

00000000

1000ffff

测试程序期望结果是：

1. 此程序最后进入死循环
2. $t0 = 0x12345678
3. $t1 = 0x12340000
4. $t2 = 0x00005678