

CPU 设计文档

一、 模块规格

(1) IFU（取指令单元）

模块端口说明如下：

表 1 IFU 端口说明

序号	信号名	方向	描述
1	Addr[25:0]	I	当前32位指令的低26位
2	Jump[1:0]	I	Jump功能的选择信号
3	Branch[1:0]	I	Branch功能的选择信号
4	Clk[1:0]	I	时钟信号
5	Reset[1:0]	I	复位信号
6	Instr[31:0]	O	输出的下一条指令

模块功能定义如下：

表 2 IFU 功能定义

序号	功能名称	功能描述
1	分支	当时钟上升沿到来时，若Branch信号为1， 且Jump信号为0， 则将PC置为 $PC + 4 + \text{sign_extend}(\text{offset} \parallel 0^2)$ ， offset为输入信号Addr的低16位
2	跳转	当时钟上升沿到来时， 若Jump信号为1， 则将PC置为 $PC[31:28] \parallel \text{Addr} \parallel 0^2$
3	复位	当复位信号有效时，将PC置为0x00000000
4	输出	输出指令存储器中地址为PC[6:2]的指令值

(2) GRF（通用寄存器组）

模块端口说明如下：

表 3 GRF 端口说明

序号	信号名	方向	描述
1	A1[4:0]	I	读寄存器地址1
2	A2[4:0]	I	读寄存器地址2
3	A3[4:0]	I	写寄存器地址
4	WD[31:0]	I	写入寄存器的数据
5	WE	I	寄存器写使能信号
6	Clk	I	时钟信号
7	RD1[31:0]	O	输出地址A1的寄存器中的数据
8	RD2[31:0]	O	输出地址A2的寄存器中的数据

模块功能定义如下：

表 4 GRF 功能定义

序号	功能名称	功能描述
1	读寄存器	输出端口RD1和RD2分别输出以输入信号A1和A2为地址的寄存器中的数据
2	写寄存器	当始终上升沿到来时，若写使能信号为1，则将输入信号WD中的数据写入以输入信号A3为地址的寄存器中

(3) ALU（算术逻辑单元）

模块端口说明如下：

表 5 ALU 端口说明

序号	信号名	方向	描述
1	SrcA[31:0]	I	参与ALU运算的第一个值
2	SrcB[31:0]	I	参与ALU运算的第二个值
3	ALUOp[2:0]	I	ALU功能的选择信号 000 ： ALU进行加法运算 100 ： ALU进行减法运算 001 ： ALU进行与运算 010 ： ALU进行或运算
4	ALUResult[31:0]	O	ALU的计算结果
5	Zero	O	如果ALU的计算结果为0，则输出1，否则输出0

模块功能定义如下：

表 6 ALU 功能定义

序号	功能名称	功能描述
1	加法运算	$ALUResult = A + B$
2	减法运算	$AUResult = A - B$
3	与运算	$ALUResult = A \& B$
4	或运算	$ALUResult = A \mid B$

(3) DM（数据存储器）

模块端口说明如下：

表 7 DM 端口说明

序号	信号名	方向	描述
1	MemAddr[4:0]	I	数据存储器读写的地址
2	MemData[31:0]	I	将要写进数据存储器的数据
3	MemWrite	I	数据存储器的写使能端
4	Data[31:0]	O	输出从数据存储器中读取的值

模块功能定义如下：

表 8 DM 功能定义

序号	功能名称	功能描述
1	读数据存储器	输出端口Data输出数据存储器在地址为MemAddr处的数据
2	写数据存储器	当时钟上升沿到来时，若MemWrite为1，则将输入信号MemData中的数据写入数据存储器在MemAddr所对应的地址中

(4) EXT（数据扩展单元）

模块端口说明如下：

表 9 EXT 端口说明

序号	信号名	方向	描述
1	Ext_In[15:0]	I	扩展单元的输入信号
2	ExtOp[1:0]	I	扩展方式的选择信号 00：进行符号扩展 01：进行零扩展 10：进行低位零扩展
3	Ext_Out[31:0]	O	扩展单元的输出信号

模块功能定义如下：

表 10 EXT 功能定义

序号	功能名称	功能描述
1	符号扩展	输出信号的低16位与输入信号相同，高16位为输入信号的符号位
2	零扩展	输出信号的低16位与输入信号相同，高16位为0
3	低位零扩展	输出信号的高16位与输入信号相同，低16位为0

二、控制器设计

控制器端口说明如下：

表 11 控制器端口说明

序号	信号名	方向	描述
1	Op[5:0]	I	当前指令的Op字段（高6位）
2	Funct[5:0]	I	当前指令的Funct字段（低6位）
3	Branch	O	控制分支的信号
4	Jump	O	控制跳转的信号
5	MemtoReg	O	写入寄存器堆的数据的选择信号
6	MemWrite	O	数据存储器写使能信号
7	ALUOp[2:0]	O	ALU功能选择信号
8	ALUSrc	O	ALU的运算数据选择信号
9	ExtOp[1:0]	O	扩展单元功能选择信号
10	RegWrite	O	寄存器写使能信号
11	RegDst	O	寄存器写入地址选择信号

控制信号真值表如下：

表 12 控制信号真值表

	addu	subu	ori	lw	sw	beq	lui
Op	000000	000000	001101	100011	1010111	000100	001111
Funct	100001	100011					
Branch	0	0	0	0	0	1	0
Jump	0	0	0	0	0	0	0
MemtoReg	0	0	0	1	x	x	0
MemWrite	0	0	0	0	1	0	0
ALUOp[0]	0	0	0	0	0	0	0
ALUOp[1]	0	0	1	0	0	0	1
ALUOp[2]	0	1	0	0	0	1	0
ALUSrc	0	0	1	1	1	0	1
ExtOp[0]	x	x	1	0	0	x	0
ExtOp[1]	x	x	0	0	0	x	1
RegWrite	1	1	1	1	0	0	1
RegDst	1	1	0	0	x	x	0

控制信号意义如下：

表 13 控制信号意义

序号	控制信号	意义
1	Branch	控制分支的信号，分支指令需要将该信号置为1
2	Jump	控制跳转的信号，跳转指令需要将该信号置为1
3	MemtoReg	当此信号为0时，选择ALU的计算结果作为寄存器堆的写入值； 当此信号为1时，选择从数据存储器中取出的信号作为寄存器堆的写入值.
4	MemWrite	数据存储器写使能信号，当需要写数据存储器时将此信号置为1
5	ALUOp[2:0]	ALU功能选择信号，根据指令需要执行的运算种类来设置相应的值
6	ALUSrc	当此信号为0时，选择指令的rt字段（[20:16]）为地址的寄存器中的数据作为ALU的第二个运算数； 当此信号为1时，选择经扩展后的立即数作为ALU的第二个运算数。
7	ExtOp[1:0]	Ext功能选择信号，根据指令需要进行的扩展类型来设置为相应的值
8	RegWrite	寄存器写使能信号，但需要些寄存器时将此信号置为1
9	RegDst	当此信号为0时，选择指令的rt字段（[20:16]）为寄存器的写入地址； 当此信号为1时，选择指令的rd字段（[15:11]）为寄存器的写入地址

三、测试程序

测试程序源代码如下：

```
    lui $s0, 0x2333
    ori $s1, $s0, 0x6666
    addu $s2, $s1, $s0
    subu $s3, $s1, $s0
    sw $s1, 0($0)
    sw $s2, 4($0)
    lw $s4, 4($0)
    ori $s5, $0, 0
    beq $s2, $s4, next
    ori $s5, $0, 0x5555
next:
    ori $s6, $0, 0x1234
    addu $s7, $s6, $0
end:
    nop
    beq $0, $0, end
```

期望运行结果:

寄存器 s0 - s7 的值分别为:

0x23330000、 0x23336666、 0x46666666、 0x00006666、

0x46666666、 0x00000000、 0x00001234、 0x00001234

数据存储器中地址 0x0、0x4 中的值分别为 0x23336666、 0x46666666