alu模块

# 作用

对输入的两个32位数进行逻辑运算、算术运算和移位运算[1]等，如下图所示，输出32位计算结果、carry(借位进位标志位)、zero(零标志位)、negative(负数标志位)和overflow(溢出标志位)。

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑运算 | 算术运算 | 移位运算 | 其它 |
| AND | ADD | SLL | LUI |
| ANDI | ADDI | SRL |  |
| OR | ADDU | SRA |  |
| ORI | ADDIU | SLLV |  |
| XOR | SUB | SRLV |  |
| XORI | SUBU | SRAV |  |

# 参数说明

module alu(

input [31:0] a, //input a

input [31:0] b, //input b

input [3:0] aluc, //alu control

output[31:0] r, //result

output zero, //zero flag

output carry, //carry flag

output negative, //negative flag

output overflow //overflow flag

);

# 设计流程



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Aluc[3] | Aluc[2] | Aluc[1] | Aluc[0] |
| Addu/addiu | 0 | 0 | 0 | 0 |
| Add/addi/lw/sw | 0 | 0 | 1 | 0 |
| Subu | 0 | 0 | 0 | 1 |
| Sub/beq/bne | 0 | 0 | 1 | 1 |
| And/andi | 0 | 1 | 0 | 0 |
| Or/ori | 0 | 1 | 0 | 1 |
| Xor/xori | 0 | 1 | 1 | 0 |
| nor | 0 | 1 | 1 | 1 |
| lui | 1 | 0 | 0 | X |
| Slt/slti | 1 | 0 | 1 | 1 |
| Sltu/sltiu | 1 | 0 | 1 | 0 |
| Sra/srav | 1 | 1 | 0 | 0 |
| Sll/sllv | 1 | 1 | 1 | X |
| Srl/srlv | 1 | 1 | 0 | 1 |

# 设计要点

1. 参数型常数[2]：用来定义延迟时间和变量宽度。在这里用来定义4选1选择器的变量宽度

格式： 模块名 #（常量）实例化

1. 系统任务$monitor和$display的区别

$display：显示当前的值

$monitor：当输出发生变化，显示发生变化的值，在某时刻，当多个值发生变化，在该时刻只输出显示一次

其中$display和$monitor的语法格式一致

**References:**

[1] MIPS, MIPS32TM Architecture For Programmers Volume II: The MIPS32 Instruction Set.

[2] 夏宇闻, 数据类型及其常量和变量, Verilog数字系统设计教程（第2版）, 北京航空航天出版社.