addsub32模块

# 作用

对32位无符号数和有符号数进行加法和减法操作，并且生成carry（进位借位标志位）、overflow（溢出标志位）、negative（负数标志位）。

# 参数说明

module addsub32(

input [31:0] a, //32bit inputA

input [31:0] b, //32bit inputB

input [1:0] aluc, //1: signed add or 0: unsigned add

output reg [31:0] c, //32bit output

output reg overflow, //overflow flag

output reg carry, //carry flag

output reg negative //negative flag

//----------------for test------------！！！要使用测试数据，请添加，

// output reg [31:0] c\_normal,

// output reg overflow\_normal,

// output reg carry\_normal,

// output reg negative\_normal

);

# 设计流程



## 标志位规则（根据ARM文档[1]）

|  |  |
| --- | --- |
| zero标志位 | 1.Z＝1表示运算结果是0，Z＝0表示运算结果不是零 |
| 2.对于CMP指令，Z＝1表示进行比较的两个数大小相等(不在alu中实现) |
| carry标志位 | 1.加法，有进位，无符号数运算发生上溢出 |
| 2.减法，借位，无符号数发生下溢出 |
| 3.包含移位的非加减法，最后一次被溢出的位的数值(在移位模块实现) |
| negative标志位 | 1.当两个补码表示的有符号整数运算时，N＝1表示运算的结果为负数，N＝0表示结果为正数或零 |
| overflow标志位 | 1.对于加/减法运算指令，当操作数和运算结果为二进制的补码表示的带符号数时，V＝1表示符号位溢出 |

## aluc编码

|  |  |  |
| --- | --- | --- |
|  | aluc[1] | aluc[0] |
| 无符号加 | 0 | 0 |
| 无符号减 | 0 | 1 |
| 有符号加 | 1 | 0 |
| 有符号减 | 1 | 1 |

# 设计要点

1.标志位判断条件：

|  |  |  |
| --- | --- | --- |
| 事件 | 指令 | 判断条件 |
| 溢出 | ADD、ADDI、SUB | 双符号位10或者01 |
| 进位 | ADDU、ADDIU | 32位运算进位addu\_cout=1 |
| 借位 | SUBU | 32位运算进位addu\_cout=0 |

注：进位和借位推导过程参考文献[2]

2.当有符号数溢出时，其结果保持上一次运算结果，所以其负数标志位为上次运算结果的最高位

negative = c[31];

3.运用加法器实现减法[3]

[a-b]补=[a]补 + [-b]补=[a]补 + ~[b]补 + 1

结合32位先行进位加法器可以得到32位减法的结果

4.双符号位[4]

32位先行进位加法器只是得到32位数的加法结果，如果要进行溢出判断还需要扩展符号位对其溢出情况进行判断。

要得到双符号位的结果，可以利用一位全加器实现，将32位加法器的结果的溢出作为一位全加器的进位，这个进位和操作数a和操作数b的符号位进行一位加法运算就得到了双符号位的另一位符号位的结果。

将一位全加器得到的结果和32位全加器的第31位符号位进行比较，即可得到溢出情况。

# 测试模块

对无符号数加法、无符号数减法、有符号数加法和有符号数减法分别进行运算。

具体设计流程如下：



**参考文献**

[1] Arm. ARM Architecture Reference Manual[Z].

[2] 杨洪斌. 兼容MIPS指令集的超标量微处理器ALU设计[J]. 计算机工程与应用. 2005.

[3] 王爱英. 原码、补码、反码及其加减法运算[M]. 计算机组成与结构（第4版）, 清华大学出版社.

[4] 王爱英. 加减法运算的溢出处理[M]. 计算机组成与结构（第4版）, 清华大学出版社.