同济大学计算机系

数字逻辑课程实验报告



学	号	1652286	_
姓	名	李辉	
专	业	计算机	
授课	老师	张冬冬	

一、实验内容

在本次实验中,我们将使用 Verilog HDL 语言实现行为级 ALU 的设计和仿真。

二、模块建模

```
`timescale 1ns / 1ns
                                                                                    carry=1;
module alu(
                                                                                    else
    input [31:0] a,//32 位输入,操作数 1
                                                                                    carry=0;
    input [31:0] b,//32 位数,操作数 2
                                                                                    negative=r[31];
    input [3:0] aluc,//4 位输入,由 aluc 控制
                                                                                    overflow=0;
    output reg[31:0] r,//32 位输出, 由 a\b 经过 aluc 的操
                                                                               end
作生成
                                                                               SUBU: begin
    output reg zero,//0 标志位
                                                                                    r=a-b;
    output reg carry,//进位标志位
                                                                                    if(!r)
    output reg negative,//负数标志位
                                                                                    zero=1;
    output reg overflow//溢出标志位
                                                                                    else
                                                                                    zero=0;
    );
   parameter ADDU=4'b0000;
                                                                                    if(a < b)
         parameter ADD=4'b0010;
                                                                                    carry=1;
         parameter SUBU=4'b0001;
                                                                                    else
         parameter SUB=4'b0011;
                                                                                    carry=0;
         parameter AND=4'b0100;
                                                                                    negative=r[31];
         parameter OR=4'b0101;
                                                                                    overflow=0;
         parameter XOR=4'b0110;
                                                                               end
                                                                               ADD: begin
         parameter NOR=4'b0111;
                                                                                   r=a+b;
         parameter LUI1=4'b1000;
         parameter LUI2=4'b1001;
                                                                                   if(!r)
         parameter SLT=4'b1011;
                                                                                   zero=1;
                                                                                   else
         parameter SLTU=4'b1010;
         parameter SRA=4'b1100;
                                                                                   zero=0;
         parameter SLL=4'b1110;
                                                                                   carry=0;
         parameter SLR=4'b1111;
                                                                                   negative=r[31];
         parameter SRL=4'b1101;
                                                          if(a[31]==1\&\&b[31]==1\&\&r[31]==0)
          reg [32:0] mm;
                                                                                   overflow=1;
          reg[31:0] cc;
                                                          if(a[31]==0\&\&b[31]==0\&\&r[31]==1)
            always@(*)begin
                                                                                   overflow=1;
                case(aluc)
                                                                                   else
                     ADDU: begin
                                                                                   overflow=0;
                         r=a+b;
                                                                               end
                                                                               SUB: begin
                         mm=a+b;
                         if(!r)
                                                                                    r=a-b;
                                                                                    if(!r)
                         zero=1;
                         else
                                                                                    zero=1;
                         zero=0;
                                                                                    else
                         if(mm[32]==1)
                                                                                    zero=0;
```

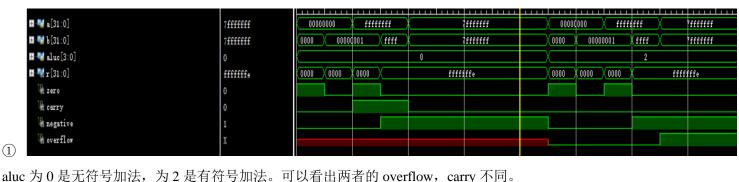
	I
carry=0;	r=a&b
negative=r[31];	if(!r)
	zero=1;
if((a[31]==1&&b[31]==0&&r[31]==0) (a[31==0]&&b[31]==0) (a[31]==0) (a[31]=	else
]==1&&r[31]==1))	zero=0;
overflow=1;	carry=0;
else	negative=r[31];
overflow=0;	overflow=0;
end	end
SRA: begin	OR: begin
if(a>0)	r=a b;
carry=b[a-1];	if(!r)
else	zero=1;
carry=0;	else
r=b>>>a;	zero=0;
if(!r)	carry=0;
zero=1;	negative=r[31];
else	overflow=0;
zero=0;	end
negative=r[31];	XOR: begin
overflow=0;	r=a^b;
end	if(!r)
SRL: begin	zero=1;
if(a>0)	else
carry=b[a-1];	zero=0;
else	carry=0;
carry=0;	negative=r[31];
r=b>>a;	overflow=0;
if(!r)	end
zero=1;	NOR: begin
else	r=~(a b);
zero=0;	if(!r)
negative=r[31];	zero=1;
overflow=0;	else
end	zero=0;
SLL: begin	carry=0;
r=b << a;	negative=r[31];
if(!r)	overflow=0;
r=1;	end
else	SLTU: begin
r=0;	r=a <b?1:0;< td=""></b?1:0;<>
negative=r[31];	if((a-b)==0)
overflow=0;	zero=1;
if(a>0)	else
carry=b[32-a];	zero=0;
else	if(a <b)< td=""></b)<>
carry=0;	carry=1;
end	else
AND: begin	carry=0;
-	•

negative=r[31];	alu uut (.a(a),				
overflow=0;	.b(b),				
end	.aluc(aluc),				
SLT: begin	.r(r),				
cc=a-b;	.zero(zero),				
if(cc[31]==1)					
	•				
else					
r=0;					
if((a-b)==0)	initial				
zero=1;	begin				
else	aluc=4'b0000;				
zero=0;	a=0;				
carry=0;	b=0;				
if(cc[31]==1)	#10				
negative=1;	a=0;				
else	b=1;				
negative=0;	#10				
	$a={32{1'b1}};$				
end	b=1;				
LUI1,LUI2: begin	#10				
$r=\{b[15:0],16'b0\};$	$a={32{1'b1}};$				
if(!r)	b={32{1'b1}};				
zero=1;	#10				
else	a={1'b0,{31{1'b1}}};				
zero=0;	b={1'b0,{31{1'b1}}};				
carry=0;	#50				
negative=r[31];					
overflow=0;	aluc=4'b0010;				
end	a=0;				
default:	b=0;				
r=a+b;	#10				
endcase	overflow=0; end SLT: begin				
end	b=1;				
endmodule	#10				
	$a={32{1'b1}};$				
	b=1;				
三、测试模块建模	#10				
二、以及大是民	$a={32{1'b1}};$				
`timescale 1ns / 1ps	$b={32{1'b1}};$				
module alu_tb;	#10				
reg [31:0]a;	a={1'b0,{31{1'b1}}};				
reg [31:0]b;	$b={1'b0,{31{1'b1}}};$				
reg [3:0]aluc;	#50				
wire [31:0]r;					
wire zero;	aluc=4'b0001;				
wire carry;	a=0;				
wire negative;	b=0;				
wire overflow;	#10				

```
a=0;
                                                                              aluc=4'b0111;
b=1;
                                                                              a = \{\{8\{1'b1\}\}, \{8\{1'b0\}\}, \{8\{1'b0\}\}, \{8\{1'b1\}\}\};
#10
                                                                              b = \{\{8\{1'b1\}\}, \{8\{1'b1\}\}, \{8\{1'b0\}\}, \{8\{1'b0\}\}\}\};
a={32{1'b1}};
b=1;
                                                                            aluc = 4'b1000;
#10
                                                                             b = \{\{16\{1'b0\}\}, \{16\{1'b1\}\}\};
a = \{32\{1'b1\}\};
                                                                             #10
b = \{32\{1'b1\}\};
                                                                              aluc =4'b1001;
 #10
                                                                              b = \{\{16\{1b1\}\}, \{16\{1b0\}\}\};
a={1'b0,{31{1'b1}}};
b = \{1'b0, \{31\{1'b1\}\}\};
                                                                              aluc = 4'b1011;
#50
                                                                              a = \{1'b1, \{31\{1'b0\}\}\};
                                                                             b = \{1'b0, \{31\{1'b1\}\}\};
   aluc=4'b0011;
                                                                             #10
                                                                              aluc = 4'b1010;
   a=0;
   b=0;
   #10
                                                                           aluc=4'b1100;
   a=0;
                                                                            b = \{1'b0, \{31\{1'b1\}\}\};
                                                                            a=32'd32;
   b=1;
   #10
                                                                            #10
   a = {32{1'b1}};
                                                                            b = \{1'b1, \{31\{1'b0\}\}\};
                                                                            a=32'd32;
   b=1;
   #10
                                                                            #10
                                                                            b=32'b101010;
   a = \{32\{1'b1\}\};
   b={32{1'b1}};
                                                                            a=32'd3;
   #10
                                                                            aluc=4'b1101;
   a = \{1'b0, \{31\{1'b1\}\}\};
                                                                            b = \{1'b0, \{31\{1'b1\}\}\};
                                                                            a=32'd32;
   b = \{1'b0, \{31\{1'b1\}\}\};
 #10
                                                                            #10
 a = \{1'b0, \{31\{1'b1\}\}\};
                                                                            b = \{1'b1, \{31\{1'b0\}\}\};
 b = \{1'b1, \{31\{1'b1\}\}\};
                                                                            a=32'd32;
                                                                            #10
 aluc=4'b0100;
                                                                            b=32'b101010;
 a = \{\{8\{1b1\}\}, \{8\{1b0\}\}, \{8\{1b0\}\}, \{8\{1b1\}\}\};
                                                                            a=32'd3;
 b = \{\{8\{1'b1\}\}, \{8\{1'b1\}\}, \{8\{1'b0\}\}, \{8\{1'b0\}\}\}\};
 #10
                                                                            aluc=4'b1111;
 aluc=4'b0101;
                                                                             b = \{16'b0, \{16\{1'b1\}\}\};
 a = \{\{8\{1b1\}\}, \{8\{1b0\}\}, \{8\{1b0\}\}, \{8\{1b1\}\}\};
                                                                             a=32'd16;
 b = \{\{8\{1b1\}\}, \{8\{1b1\}\}, \{8\{1b0\}\}, \{8\{1b0\}\}\}\};
                                                                             #10
 #10
                                                                             b = \{16'b1, \{16\{1'b0\}\}\};
 aluc=4'b0110;
                                                                              a=32'd16;
 a = \{\{8\{1b1\}\}, \{8\{1b0\}\}, \{8\{1b0\}\}, \{8\{1b1\}\}\}\};
 b = \{\{8\{1b1\}\}, \{8\{1b1\}\}, \{8\{1b0\}\}, \{8\{1b0\}\}\}\};
                                                                      end
 #10
                                                                      endmodule
```

四、实验结果

仿真波形分析:



aluc 为 0 是无符号加法, 为 2 是有符号加法。可以看出两者的 overflow, carry 不同。

⊞- ₩ a[31:0]	7 ffffff	00000000	ffffffff	X	fffffff	(0000	0000 X ff1	ffffff	Ī	76666666
⊞-₩ b[31:0]	7fffffff	X 0000 X 00	000001 /ffff	X	fffffff	X 0000	00000001	ffff	7fff	ffffffff
■-W aluc[3:0]	0	χ		1		\square		3		
⊞ r [31:0]	fffffffe	0000 Xffff	ffff		00000000		ffff (ffff	00000	000	80000000
₩ zero	0									
Th carry	0									
negative	1									
w overflow	X									
,										

aluc 为 1 是无符号减法, 3 是有符号减法。结果正确。

(2)



分别进行了与、或、异或、或非运算,结果正确。



进行了置高位立即数运算,结果正确。



进行了有符号数和无符号数的比较,r正确,其余标志位也正确。

		0.00000000	0.000000013	0.00000002	0.00000003	0.000000002	0.000000000	0.000000
⊞ - ™ a[31:0]	00000003	<u> </u>	00000	020		(00000	003
⊞ - ™ b[31:0]	0000002a	7fffffff	80000000	7fffffff	80000000		00000	02a
	d							
± - ¶ r [31:0]	00000005	00000000	ffffffff	00000	000		00000	005
le zero	0							
1 carry	0							
le negative	0							
We overflow	X							
6								

进行了数的算术、逻辑右移,前者中结果体现为最高位补的位为符号位,后者补 0。Carry 位为最后一个移出的位,正确。



左移,无论是算术还是逻辑,全都在最低位补0。