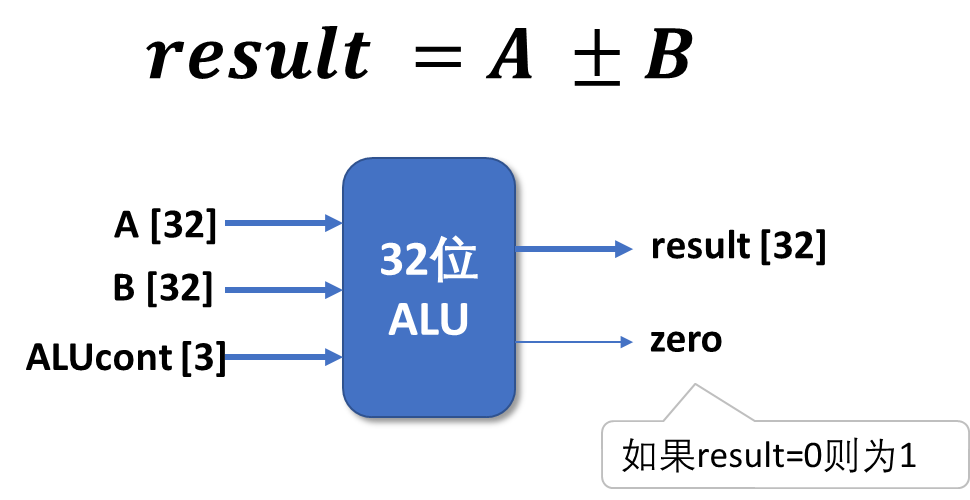
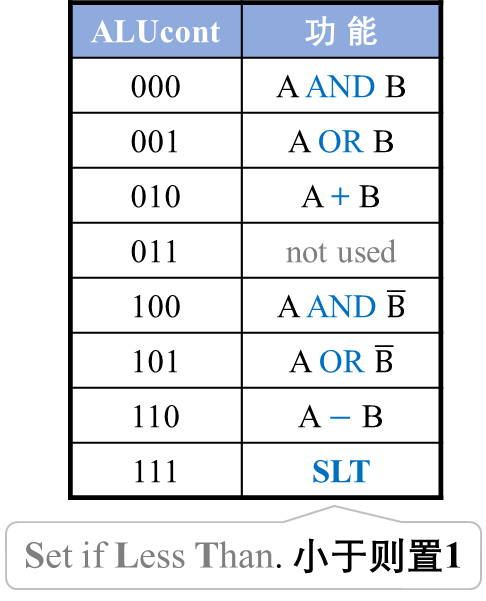
计算机体系结构实验1：32-bit ALU设计

15307110200 廖文

1. 实验要求
2. 仔细阅读教材5.2.4节(P153)内容
3. 根据下面的图表设计32位ALU，编写Verilog HDL代码



1. 补全下页表格，编写测试代码测试ALU
2. 实验过程
3. 功能分析

输入：A，B为32bit二进制树，ALUcont为控制位

输入：32位二进制整数，零符号位

运算：算术运算(+、-)，位运算(AND, OR, AND’, OR’)，SLT

1. ALU实现

**module** cpu\_32bit**(**A**,**B**,**ALUcont**,** result**,** zero**);**

**input** **wire[**31**:**0**]** A**,** B**;**

**input** **wire[**2**:**0**]** ALUcont**;**

**output** **wire[**31**:**0**]** result**;**

**output** **wire** zero**;**

**wire[**31**:**0**]** minus**;**

**assign** minus **=** A**-**B**;**

**assign** result **=** **(**ALUcont **===** 3'b000**)** **?** A**&**B **:**

**(**ALUcont **===** 3'b001**)** **?** A**|**B **:**

**(**ALUcont **===** 3'b010**)** **?** A**+**B **:**

**(**ALUcont **===** 3'b100**)** **?** A**&~**B **:**

**(**ALUcont **===** 3'b101**)** **?** A**|~**B **:**

**(**ALUcont **===** 3'b110**)** **?** minus **:**

**(**ALUcont **===** 3'b111**)** **?** **{**31'b0**,** minus**[**31**]** **}** **:** 32'b0**;**

**assign** zero **=** **~|**result**;**

**endmodule**

1. ALU测试

按照表格给出的测试用例进行测试。

`define DELAY 20

**module** cpu\_32bit\_test**();**

**reg[**31**:**0**]** A**,**B**;**

**reg[**3**:**0**]** ALUcont**;**

**wire[**31**:**0**]** result**;**

**wire** zero**;**

cpu\_32bit f**(**A **,** B**,** ALUcont**,** result**,** zero**);**

**initial** **begin**

A **=** 32'h0**;** B **=** 32'h0**;** ALUcont **=** 3'd2**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd2**;**

**#**`DELAY**;**

A **=** 32'h1**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd2**;**

**#**`DELAY**;**

A **=** 32'hFFFFFFFF**;** B **=** 32'h1**;** ALUcont **=** 3'd2**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'h0**;** ALUcont **=** 3'd6**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd6**;**

**#**`DELAY**;**

A **=** 32'h1**;** B **=** 32'h0**;** ALUcont **=** 3'd6**;**

**#**`DELAY**;**

A **=** 32'h100**;** B **=** 32'h0**;** ALUcont **=** 3'd6**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'h0**;** ALUcont **=** 3'd7**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'h1**;** ALUcont **=** 3'd7**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd7**;**

**#**`DELAY**;**

A **=** 32'h1**;** B **=** 32'h0**;** ALUcont **=** 3'd7**;**

**#**`DELAY**;**

A **=** 32'hFFFFFFFF**;** B **=** 32'h0**;** ALUcont **=** 3'd7**;**

**#**`DELAY**;**

A **=** 32'hFFFFFFFF**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd0**;**

**#**`DELAY**;**

A **=** 32'hFFFFFFFF**;** B **=** 32'h12345678**;** ALUcont **=** 3'd0**;**

**#**`DELAY**;**

A **=** 32'h12345678**;** B **=** 32'h87654321**;** ALUcont **=** 3'd0**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd0**;**

**#**`DELAY**;**

A **=** 32'hFFFFFFFF**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd1**;**

**#**`DELAY**;**

A **=** 32'h12345678**;** B **=** 32'h87654321**;** ALUcont **=** 3'd1**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'hFFFFFFFF**;** ALUcont **=** 3'd1**;**

**#**`DELAY**;**

A **=** 32'h0**;** B **=** 32'h0**;** ALUcont **=** 3'd1**;**

**#**`DELAY**;**

**end**

**initial**

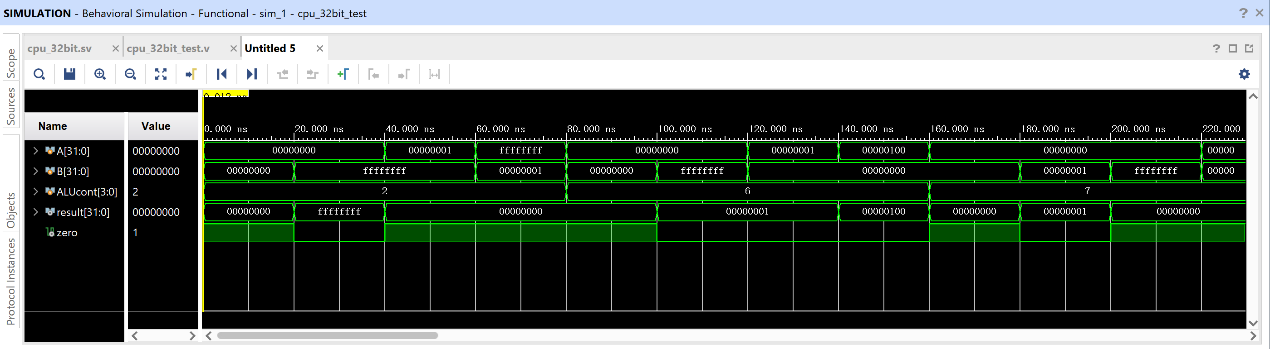
**begin**

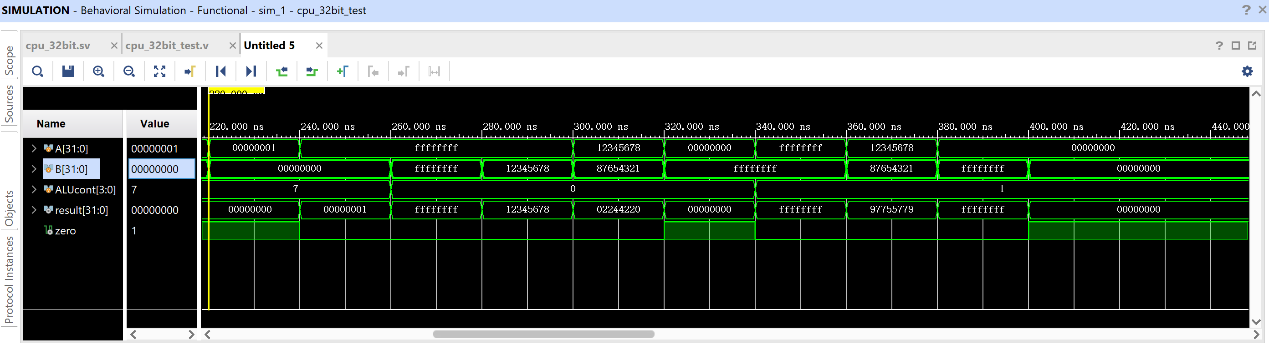
$monitor**(**"time = %2d, A =%32b, B=%32b, ALUcont=%3b, result=%32b , zero=%3b"**,** $time**,** A**,** B**,** ALUcont**,** result**,** zero**);**

**end**

**endmodule**

模拟的结果如下。

****

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | **ALUcont** | **A** | **B** | **result** | **zero** |
| ADD 0+0 | 2 | 00000000 | 00000000 | 00000000 | 1 |
| ADD 0+(-1) | 2 | 00000000 | FFFFFFFF | FFFFFFFF | 0 |
| ADD 1+(-1) | 2 | 00000001 | FFFFFFFF | 00000000 | 1 |
| ADD FF+1 | 2 | 000000FF | 00000001 | 00000100 | 0 |
| SUB 0-0 | 6 | 00000000 | 00000000 | 00000000 | 1 |
| SUB 0-(-1) | 6 | 00000000 | FFFFFFFF | 00000001 | 0 |
| SUB 1-1 | 6 | 00000001 | 00000001 | 00000000 | 1 |
| SUB 100-1 | 6 | 00000100 | 00000001 | 000000FF | 0 |
| SLT 0,0 | 7 | 00000000 | 00000000 | 00000000 | 1 |
| SLT 0,1 | 7 | 00000000 | 00000001 | 00000001 | 0 |
| SLT 0,-1 | 7 | 00000000 | FFFFFFFF | 00000000 | 1 |
| SLT 1,0 | 7 | 00000001 | 00000000 | 00000000 | 1 |
| SLT –1,0 | 7 | FFFFFFFF | 00000000 | 00000001 | 0 |
| AND FFFFFFFF, FFFFFFFF | 0 | FFFFFFFF | FFFFFFFF | FFFFFFFF | 0 |
| AND FFFFFFFF, 12345678 | 0 | FFFFFFFF | 12345678 | 12345678 | 0 |
| AND 12345678, 87654321 | 0 | 12345678 | 87654321 | 02244220 | 0 |
| AND 00000000, FFFFFFFF | 0 | 00000000 | FFFFFFFF | 00000000 | 1 |
| OR FFFFFFFF, FFFFFFFF | 1 | FFFFFFFF | FFFFFFFF | FFFFFFFF | 0 |
| OR 12345678, 87654321 | 1 | 12345678 | 87654321 | 97755779 | 0 |
| OR 00000000, FFFFFFFF | 1 | 00000000 | FFFFFFFF | FFFFFFFF | 0 |
| OR 00000000, 00000000 | 1 | 00000000 | 00000000 | 00000000 | 1 |