



**实验报告**

**课程名称**  **计算机系统高级课程**

**题目名称 ALU设计**

**学生学院 计算机学院**

**专业班级 信息安全2018级2班**

**学 号 3118005414**

**学生姓名 孔止**

**指导教师 胡志斌**

2020 年 6 月 15 日

**实验内容与设计要求**

**1 设计目标**

本次实验的主要内容是利用设计工具，设计一个ALU。

（1）融会贯通本课程各章节的内容，通过知识的综合运用，加深对计算机系统各功能部件的工作原理及相互联系的认识，加深计算机工作中“时间-空间”概念的理解，从而清晰地建立计算机的整机概念；

（2）学习设计和调试计算机的基本步骤和方法，提高使用Quartus等软件仿真工具和集成电路的基本技能；

（3）培养科学研究的独立工作能力，取得工程设计与组装调试的实践和经验。

**2 设计内容**

第一步可先练习例2-3mux21，例2-19一位全加器，都要做出仿真波形

设计一个按74181的功能表工作的8位数据输入的ALU，用VHDL 语言完成设计并调试成功。

**3 设计要求**

（1）根据给定的要求，设计一个ALU。

（2）根据设计图，在QUARTUS环境下仿真调试成功。

（3）在调试成功的基础上，整理出设计图纸和相关文件，包括：

（a）ALU逻辑图；

（b）设计说明书；电路原理图、仿真波形图。

**4 成绩评定**

程序设计方案是否合理；程序设计是否正确；调试结果；设计说明书的质量；答辩时回答问题情况；课程设计表现情况。

已启用抄袭检查系统，不得雷同，不得抄袭，一旦发现成绩为零。

目 录

[一 ALU功能 1](#_Toc531768839)

[二 设计原理与电路图 1](#_Toc531768840)

[三 系统调试情况 2](#_Toc531768841)

[四 参考资料 4](#_Toc531768842)

[附录 VHDL程序 5](#_Toc531768843)

# 一 ALU功能

本实验按照74LS181的功能表，设计出可以进行8位运算的简单ALU。

此ALU由时钟信号控制，可以执行16种逻辑运算和16种算术运算，由选择输入端S和模式控制端M执行不同的运算。所有端口都是高电平有效，且规定高电平为1，低电平为0。当控制端M为高电平时，进行逻辑运算，且无视进位输入。当控制端M为低电平时，进行算术运算。进行算术运算时，分两种情况：进位输入低电平，进位输入高电平。运算操作数在输入端A8和输入端B8，运算结果存储在输出端F8和进位输出端Cout。除了基本运算之外，还可以对操作数进行相等的判断：如果A8等于B8，则输出端Eq为高电平，否则为低电平。

另外，此ALU以二进制补码进行算术运算。

功能表(*plus*为算术加法，*minus*为算术减法，其他符号均为逻辑运算符)：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S | | | | M=1 | M=0 | |
| Cin=0 | Cin=1 |
| S3 | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 |  |  |  |
| 0 | 0 | 0 | 1 |  |  |  |
| 0 | 0 | 1 | 0 |  |  |  |
| 0 | 0 | 1 | 1 |  |  |  |
| 0 | 1 | 0 | 0 |  |  |  |
| 0 | 1 | 0 | 1 |  |  |  |
| 0 | 1 | 1 | 0 |  |  |  |
| 0 | 1 | 1 | 1 |  |  |  |
| 1 | 0 | 0 | 0 |  |  |  |
| 1 | 0 | 0 | 1 |  |  |  |
| 1 | 0 | 1 | 0 |  |  |  |
| 1 | 0 | 1 | 1 |  |  |  |
| 1 | 1 | 0 | 0 |  |  |  |
| 1 | 1 | 0 | 1 |  |  |  |
| 1 | 1 | 1 | 0 |  |  |  |
| 1 | 1 | 1 | 1 |  |  |  |

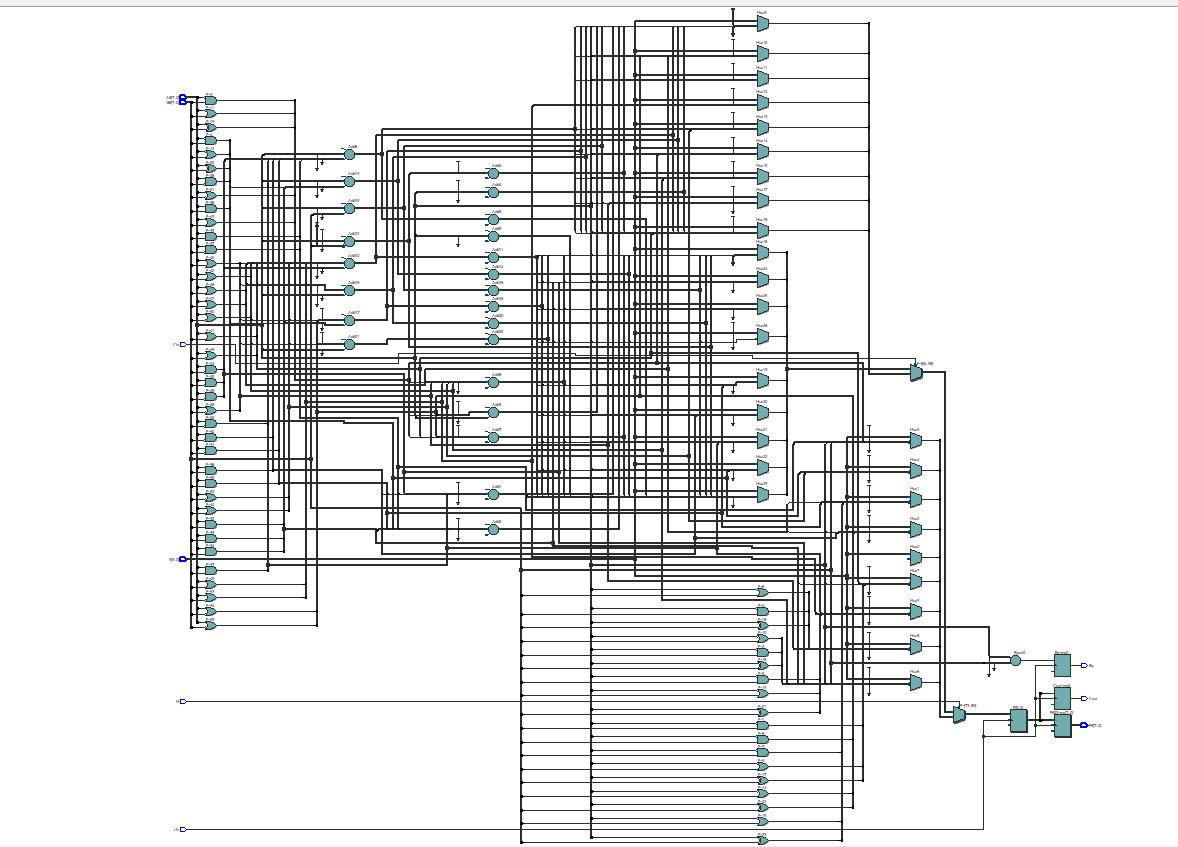
：为的二进制补码

# 二 设计原理与电路图

（1）框图



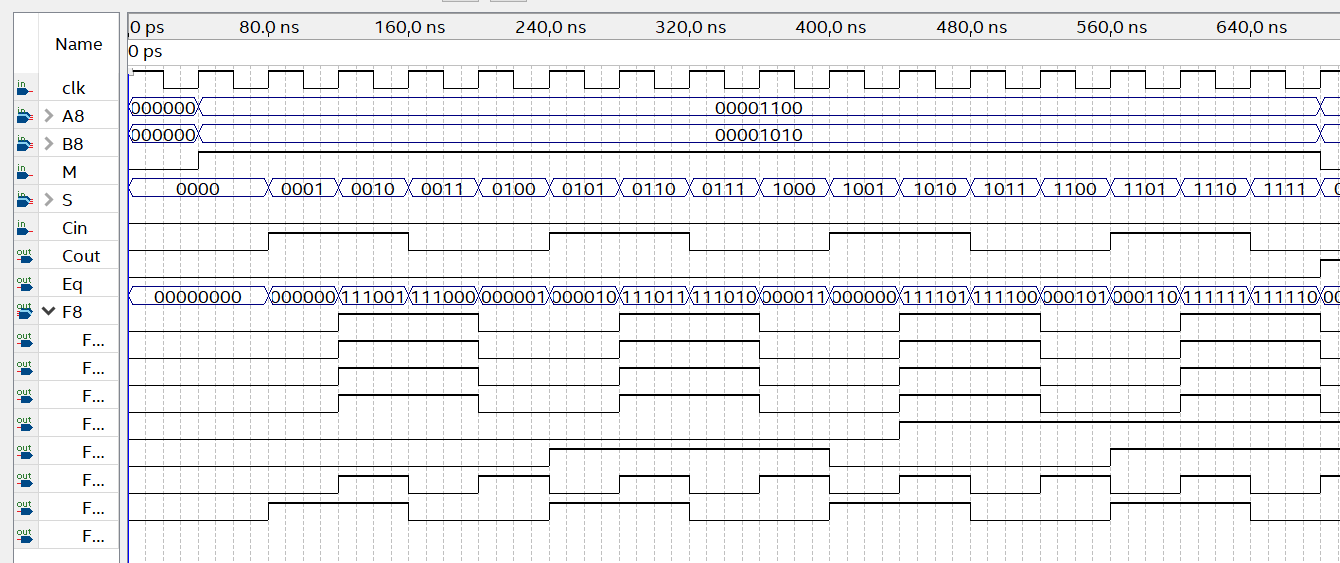
（2）电路图



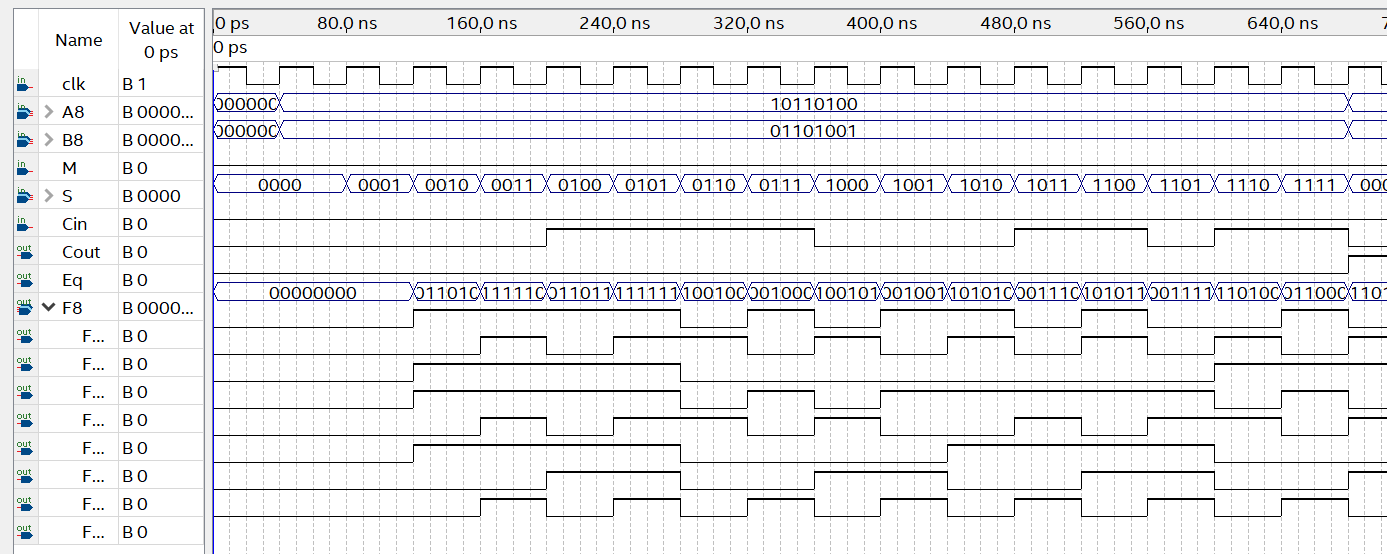
# 三 系统调试情况

1. 测试结果（波形图）：

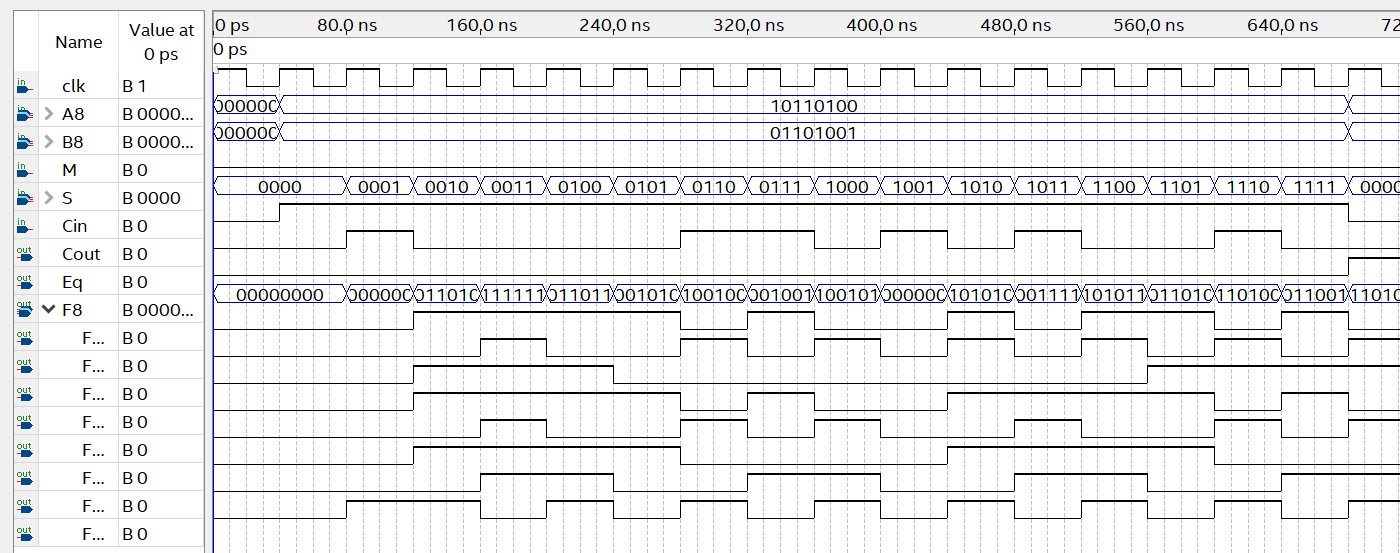
逻辑运算：

****

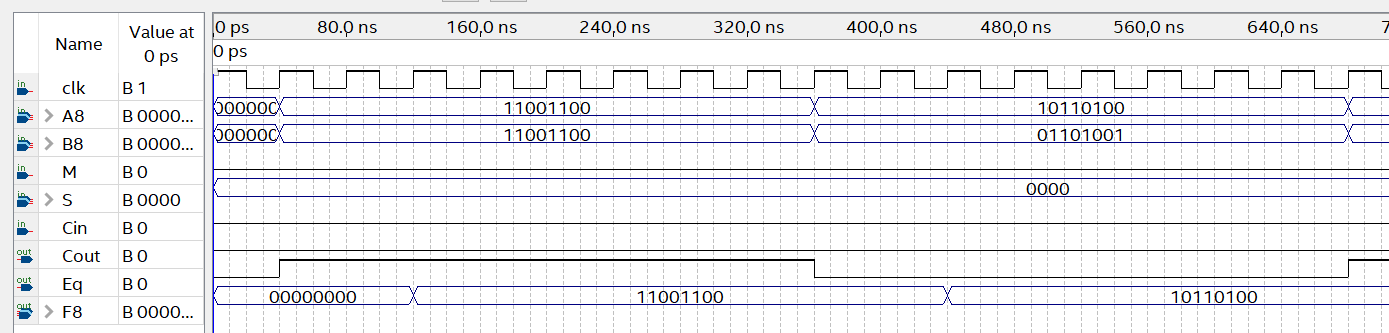
无进位算术运算：



有进位算术运算：



判断是否相等：



# 四 参考资料

[1] 潘松,潘明. 现代计算机组成原理[M]. 北京：科学出版社,2007.

[2] 陈华光. 计算机组成原理[M].北京：机械工业出版社，2004：5-10.

[3] 侯伯亨等. VHDL硬件描述语言与数字逻辑电路设计（修订版）[M]，西安：西安电子科技大学出版社，1999：75-78.

[4] 杨东旭等.计算机组成原理实用教程[M]. 北京：清华大学出版社, 2001：120-122.

# 附录 VHDL程序

算术运算器 ALU

--ALU.vhd

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

--高电平有效

entity ALU is

port( S: in std\_logic\_vector(3 downto 0);

A8,B8: in std\_logic\_vector(7 downto 0);

M: in std\_logic;

Cin: in std\_logic;

clk: in std\_logic;

F8: out std\_logic\_vector(7 downto 0);

Eq: out std\_logic;

Cout: out std\_logic);

end ALU;

architecture one of ALU is

signal A,B,F: std\_logic\_vector(8 downto 0);

begin

A <= '0' & A8; B <= '0' & B8;

process(clk, A, B, M, Cin)

begin

if clk'event and clk='1' then

if M = '1' then --M=H

case s is

when "0000" => F <= not A;

when "0001" => F <= not (A or B);

when "0010" => F <= (not A) and B;

when "0011" => F <= (others=>'0');

when "0100" => F <= not (A and B);

when "0101" => F <= not B;

when "0110" => F <= A xor B;

when "0111" => F <= A and (not B);

when "1000" => F <= (not A) or B;

when "1001" => F <= not (A xor B);

when "1010" => F <= B;

when "1011" => F <= A and B;

when "1100" => F <= (others=>'1');

when "1101" => F <= A or (not B);

when "1110" => F <= A or B;

when "1111" => F <= A;

end case;

else

if Cin = '0' then --M=L, Cin=L

case s is

when "0000" => F <= A;

when "0001" => F <= A or B;

when "0010" => F <= A or (not B);

when "0011" => F <= (others=>'1');

when "0100" => F <= A + (A and (not B));

when "0101" => F <= (A or B) + (A and (not B));

when "0110" => F <= A - B - 1;

when "0111" => F <= (A and (not B)) - 1;

when "1000" => F <= A + (A and B);

when "1001" => F <= A + B;

when "1010" => F <= (A or (not B)) + (A and B);

when "1011" => F <= (A and B) - 1;

when "1100" => F <= A + A;

when "1101" => F <= (A or B) + A;

when "1110" => F <= (A or (not B)) + A;

when "1111" => F <= A - 1;

end case;

else --M=L, Cin=H

case s is

when "0000" => F <= A + 1;

when "0001" => F <= (A or B) + 1;

when "0010" => F <= (A or (not B)) + 1;

when "0011" => F <= (others=>'0');

when "0100" => F <= A + (A and (not B)) + 1;

when "0101" => F <= (A or B) + (A and (not B)) + 1;

when "0110" => F <= A - B;

when "0111" => F <= A and (not B);

when "1000" => F <= A + (A and B) + 1;

when "1001" => F <= A + B + 1;

when "1010" => F <= (A or (not B)) + (A and B) + 1;

when "1011" => F <= A and B;

when "1100" => F <= A + A + 1;

when "1101" => F <= (A or B) + A + 1;

when "1110" => F <= (A or (not B)) + A + 1;

when "1111" => F <= A;

end case;

end if;

end if;

if A = B then

Eq <= '1';

else

Eq <= '0';

end if;

end if;

end process;

process(clk, F)

begin

if clk'event and clk='1' then

F8 <= F(7 downto 0);

Cout <= F(8);

end if;

end process;

end one;