**实验二 运算器仿真实验**

**1 实验目的**

1. 理解运算器的功能。
2. 掌握运用Proteus软件设计能够实现算术逻辑运算的运算器的方法。

(3) 基于Proteus仿真工具完成要求的算术逻辑运算，理解各控制信号的作用及时序。

**2 实验设备**

安装有Proteus软件的PC 机一台。

**3 实验原理**

本实验的原理如图1所示。

OUT[7…0]

三态门（245）

算术逻辑运算单元

FZ CN

FC

S3…S0 M

暂存器B

暂存器A

T4 T4

LDA LDB

三态门（245）

IN[7…0]

图1 运算器原理图

运算器ALU由两个74LS181通过级联实现，输入的数据单元IN[7…0]通过8位的双总线锁存器74LS245存到两个8位暂存器74LS2473（暂存器A、暂存器B）。 使用两个74L273构成2个8位数据暂存器，用来存储运算过程中使用的数据。然后通过两个级联的4位运算器运算A、B两个8位数据，输出到两个7段数码管显示运算结果；

执行的运算类型由74LS181的控制端S3…S0、M和CN决定：M=1时执行逻辑运算；M=0时执行算术运算（其中包括几个逻辑运算），此时若CN=1表示低位没有进位，而CN=0则表示低位有进位。运算结果有两个标志位：溢出标志FC(Flag of Carry)，由ALU进位输出端反向形成，CF为1时（即FC标志灯亮时）运算结果有进位（有时也可称为溢出）；零标志FZ(Flag of Zero)由8位输出结果或非形成，FZ为1时（即ZF标志灯亮时）表明运算结果为零。

ALU执行的算术逻辑运算类型由74LS181芯片功能决定，具体功能如表1所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表1 74LS181功能表** | | | | | | |
|  |  |  |  | M = H Logic Functions | M = L Arithmetic Operations（Include several Logic Functions） | |
| S3 | S2 | S1 | S0 |  | C = 1 | CN = 0 |
| L | L | L | L | F = /A | F = A | F = A plus 1 |
| L | L | L | H | F = /(A + B) | F = A + B | F = ( A + B ) plus 1 |
| L | L | H | L | F = (/A)B | F = A + /B | F = ( A + /B) plus 1 |
| L | L | H | H | F = 0 | F = minus 1 (2s Comp) | F = ZERO |
| L | H | L | L | F = /(AB) | F = A plus A(/B) | F = A plus A(/B) plus 1 |
| L | H | L | H | F = /B | F = ( A + B ) plus A(/B) | F = ( A + B ) plus A(/B) plus 1 |
| L | H | H | L | F = A xor B | F = A minus B minus 1 | F = A minus B |
| L | H | H | H | F = A(/B) | F = A(/B) minus 1 | F = A(/B) |
| H | L | L | L | F = /A+ B | F = A plus AB | F = A plus AB plus 1 |
| H | L | L | H | F = /(A xor B) | F = A plus B | F = A plus B plus 1 |
| H | L | H | L | F = B | F = ( A + /B ) plus AB | F = ( A + /B) plus AB plus 1 |
| H | L | H | H | F = AB | F = AB minus 1 | F = AB |
| H | H | L | L | F = 1 | F = A plus A | F = A plus A plus 1 |
| H | H | L | H | F = A + /B | F = ( A + B ) plus A | F = ( A + B ) plus A plus 1 |
| H | H | H | L | F = A + B | F = ( A + /B) plus A | F = ( A + /B) plus A plus 1 |
| H | H | H | H | F = A | F = A minus 1 | F = A |
| 注：+ 是或的意思， /是非， plus是加， minus是减， xor是异或 | | | | | | |

**4 实验步骤**

1. 按图 2 绘制实验电路，并检查无误。
2. 设置IN\_R0、ALU\_OUT为1、CE\_R0、CE\_OUT为1，LDA、LDB设置为0，启动仿真。

(3) 用输入开关输入2个8位的二进制数据，分别存入两个暂存器。

① 拨动 DSW1 数据开关，形成二进制数 00000001（或其它数值）。

② 将CE\_R0拨至低电平，LDA拔至高电平，则将二进制数00000001 置入暂存器A中，再将LDA拔至低电平；

③拨动 DSW1 数据开关，形成二进制数 00000011（或其它数值）。

④将CE\_R0拨至低电平，LDB拔至高电平，则将二进制数00000011 置入暂存器B中，再将LDB拔至低电平；

(4) 设置M、CN、S3…S0的值，观察运算器的运算结果是否正确；

(5) 设置CE\_OUT为低电平，观察LED灯的显示是否正确；

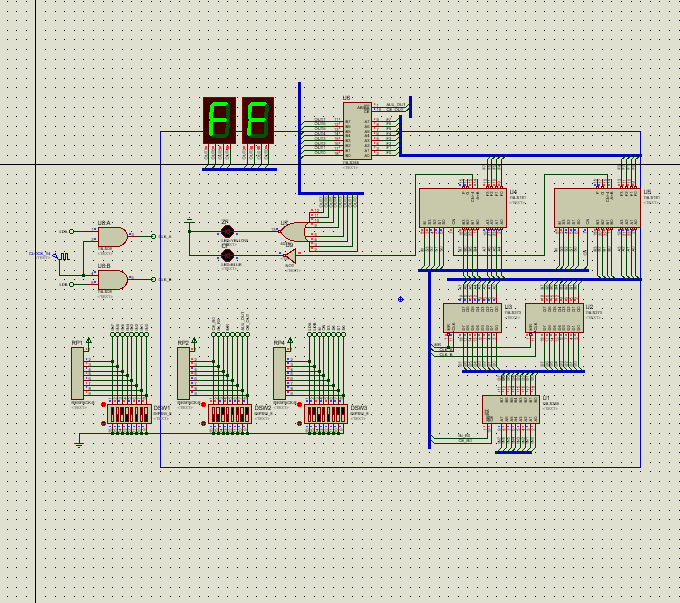
**5 实验结果**

根据M、CN、S3…S0的值，记录数据的运算结果。**选取其中2个运算（算术、逻辑各一）把其运行状态截图提交。**

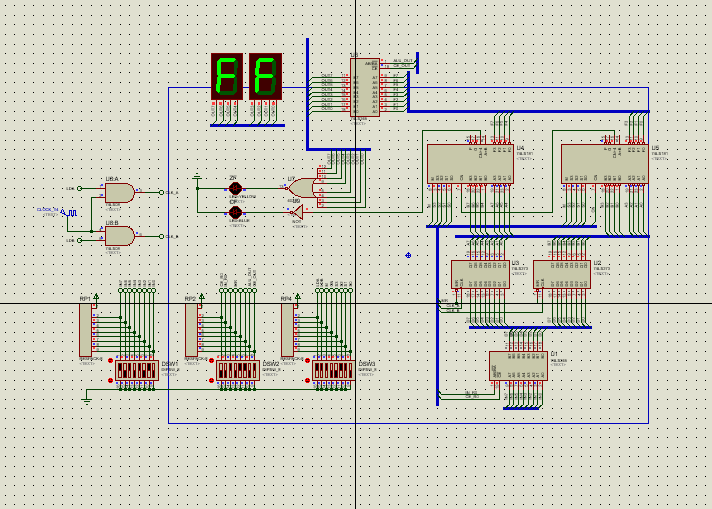
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **S3** | **S2** | **S1** | **S0** | **M=0（算术运算）** | | | | | | **M=1**  **(逻辑运算)** | | |
| **CN=1** | | | **CN=0** | | | **OUT** | **FC** | **FZ** |
| 数据1 | 数据2 | **OUT** | **FC** | **FZ** | **OUT** | **FC** | **FZ** |
| 10101010 | 01010101 | 0 | 0 | 0 | 0 | F=A  0xFA | 0 | 1 | F = A plus 1  0xFB | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 0 | 0 | 0 | 1 | F = A + B  0x00 | 0 | 0 | F = ( A + B ) plus 1  0x01 | 1 | 1 | 00 | 1 | 1 |
| 10101010 | 01010101 | 0 | 0 | 1 | 0 | F = A + /B  0xFA | 0 | 1 | F = ( A + /B) plus 1  0xFB | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 0 | 0 | 1 | 1 | F = minus 1 (2s Comp)  0xFF | 0 | 0 | F = ZERO  0x00 | 1 | 1 | 00 | 1 | 1 |
| 10101010 | 01010101 | 0 | 1 | 0 | 0 | F = A plus A(/B)  0x6A | 1 | 1 | F = A plus A(/B) plus 1  0x6B | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 0 | 1 | 0 | 1 | F = ( A + B ) plus A(/B)  0x6F | **0** | 0 | F = ( A + B ) plus A(/B) plus 1  0x70 | 1 | 1 | 00 | 1 | 1 |
| 10101010 | 01010101 | 0 | 1 | 1 | 0 | F = A minus B minus 1  0x6A | 0 | 1 | F = A minus B  0x6B | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 0 | 1 | 1 | 1 | F = A(/B) minus 1  0x6F | 0 | 0 | F = A(/B)  0x70 | 1 | 1 | 00 | 1 | 1 |
| 10101010 | 01010101 | 1 | 0 | 0 | 0 | F = A plus AB  0x84 | 0 | 1 | F = A plus AB plus 1  0x85 | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 1 | 0 | 0 | 1 | F=A plus B  0x89 | 0 | 0 | F = A plus B plus 1  0x8A | 1 | 1 | 00 | 1 | 1 |
| 10101010 | 01010101 | 1 | 0 | 1 | 0 | F = ( A + /B ) plus AB  0x84 | 1 | 1 | F = ( A + /B) plus AB plus 1  0x85 | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 1 | 0 | 1 | 1 | F = AB minus 1  0x89 | 1 | 1 | F = AB  0x8A | 0 | 0 | 00 | 1 | 1 |
| 10101010 | 01010101 | 1 | 1 | 0 | 0 | F = A plus A  0xF4 | 0 | 1 | F = A plus A plus 1  0xF5 | 0 | 0 | FF | 0 | 0 |
| 10101010 | 01010101 | 1 | 1 | 0 | 1 | F = ( A + B ) plus A  0xF9 | 0 | 0 | F = ( A + B ) plus A plus 1  0xFA | 1 | 1 | 00 | 1 | 1 |
| 1111 1010 | 1000 1111 | 1 | 1 | 1 | 0 | F = ( A + /B) plus A  0xF4 | 1 | 0 | F = ( A + /B) plus A plus 1  0xF5 | 1 | 0 | F = A + B  0xFF | 1 | 0 |
| 1111 1010 | 1000 1111 | 1 | 1 | 1 | 1 | F=A  0xFA | 1 | 0 | F = A plus 1  0xFB | 1 | 0 | F = A  0xFA | 1 | 0 |

截图说明运算状态（标清楚做的哪个运算）：

加法A plus B：



A or B:



**6 回答问题**

解释实验过程中这些控制信号是起什么作用？

1. IN\_R0、ALU\_OUT为1：控制输入输出
2. CE\_R0、CE\_OUT为1：清除输入输出
3. LDA设置为0：关闭A的存入
4. LDB设置为1：开始存入B
5. CE\_OUT设置为低电平：关闭输出
6. CE\_R0拨至低电平：清0输入

**7 实验总结**

学习了加法器存储和运算原理和相关器件的工作顺序。

1.对于相关开关的作用开始存在认知难点，通过查阅资料，可知加法器是对于1位的二进制加法，相关的有五个的量：1，被加数A，2，加数B，3，前一位的进位CIN，4，此位二数相加的和S，5，此位二数相加产生的进位COUT。前三个量为输入量，后两个量为输出量，五个量均为1位。同时了解到各个开关的控制量，并根据实验原理合理设置开关的开和关，得到了理想的运算结果。

2.对于Proteus仿真工具的使用存在一定问题：

阅读help和自身不断试错，网上查阅期间型号和相关快捷键熟练了Proteus仿真工具的使用。