计算机组成原理

实验报告

**实验二 ALU的设计**

**一、实验内容**

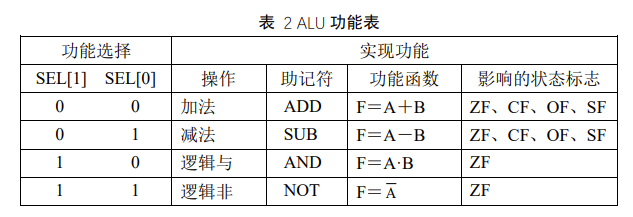
（1）测试加/减法器的功能。

（2）设计具有加法、减法、逻辑与、逻辑非4种功能且数据宽度为8bit的ALU，并进行功能仿真/验证。

**二、电路设计与实现**

1. 需求分析：

本实验要求 ALU 的数据宽度为 8 位，具有 4 种算术及逻辑运算功能，可产生结果状态标志 ZF、CF、OF、SF。 ALU 的功能如表 2 所示，其中，A、B 为数据入端，F 为数据出端，SEL 为功能选择（操作 控制）信号。可见，SEL 为 2 位（记为 SEL[1..0]），F 的位数与 A 及 B 相同。



1. 模块分析：

数据选择器用mux，四种功能对应2位sel进行选择性输入

四种运算：

加法：利用加法器。减法：用加法器转化为补码进行加法运算。

加减法选择：00为加（1），01为减（0），适合用同或

逻辑与：与门。逻辑非：非门。

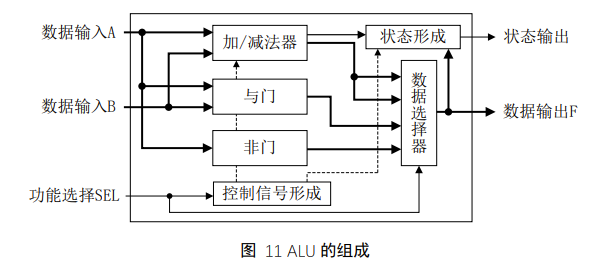
四个状态参数：

SF：符号位是结果的最高位，取S[7..7]

OF：连接加法器的overflow

ZF：运算结果是0则取1，用位间或运算再取反获得。

CF：减法是补码的加法，不借位时cout==1，借位时反而为0。加法器选择（加法）1且cout==1时CF=1(进位)，加法器选择（减法）0 且cout==0，仅这两种情况CF=1，适合采用同或.



1. 引脚组织：

lpm\_or, lpm\_xor左侧使用bus，input数据个数为bus宽，wide为1个，实现位间运算

Lpm\_and\_sub选用overflow溢出显示，cout进借位显示，and\_sub选择

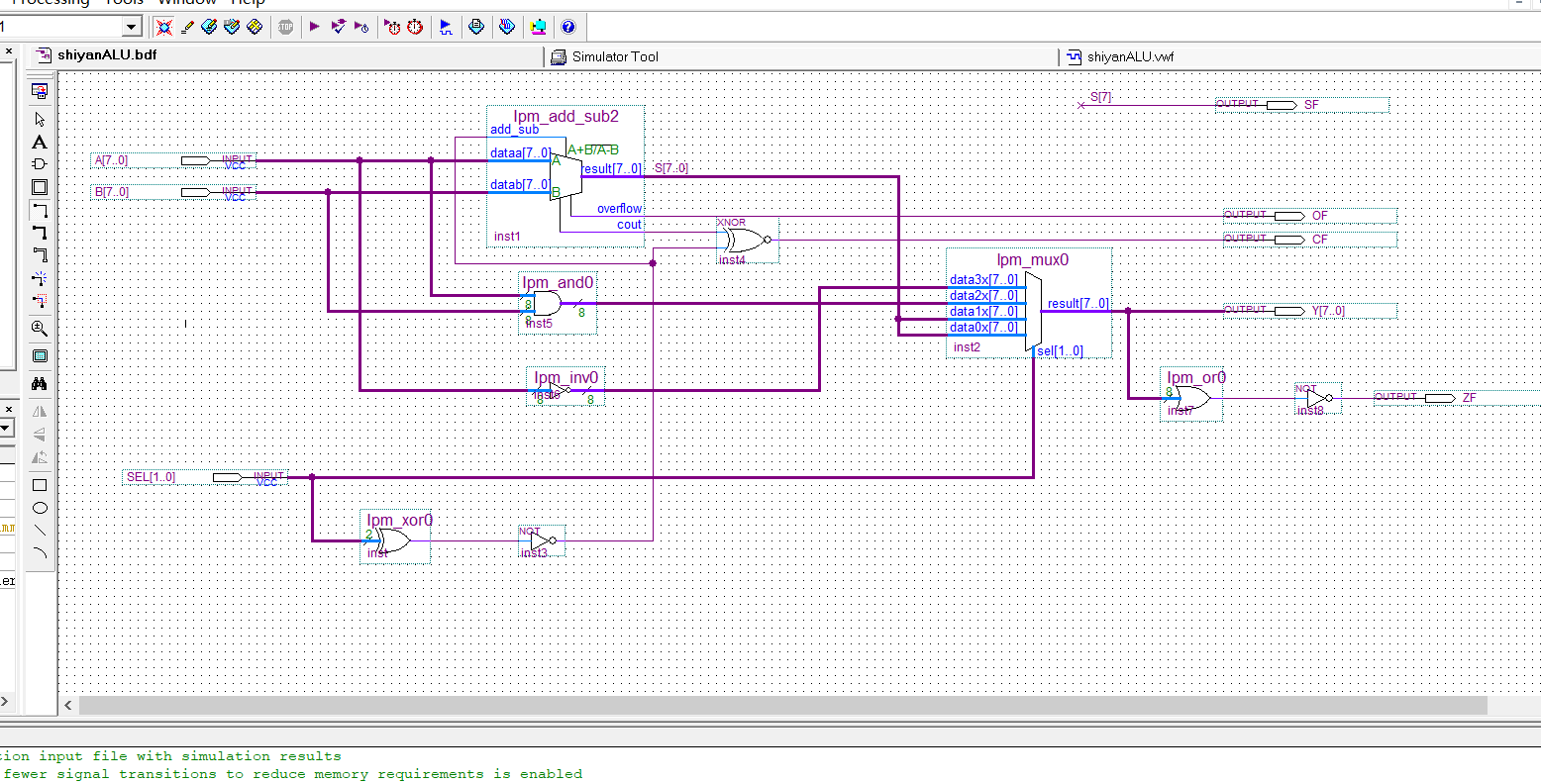
Lpm\_mux4个8位输入，1个2位控制与选择，1个8位输出

( cin 引脚的功能是实现带进位加法或带借位减法，本实验不需要设置该引脚；)

1. 器件选择：一个lpm\_add\_sub 实现加减法，一个lmp\_inv实现非，一个lmp\_and实现与，一个数据选择器lmp\_mux选择输出结果，一个lmp\_xor,一个not实现减法选择，再加上一个xnor实现CF进借位判断，一个lmp\_or与一个not实现ZF
2. 信号命名：输入数据A[7..0],B[7..0]，输出数据F[7..0]

输出选择信号SEL[1..0],四个标志位：CF,OF,SF,ZF

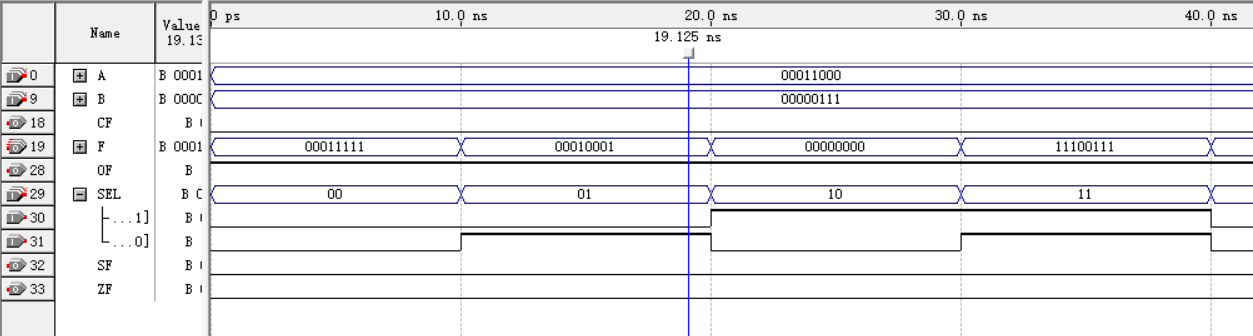
1. 电路组成：

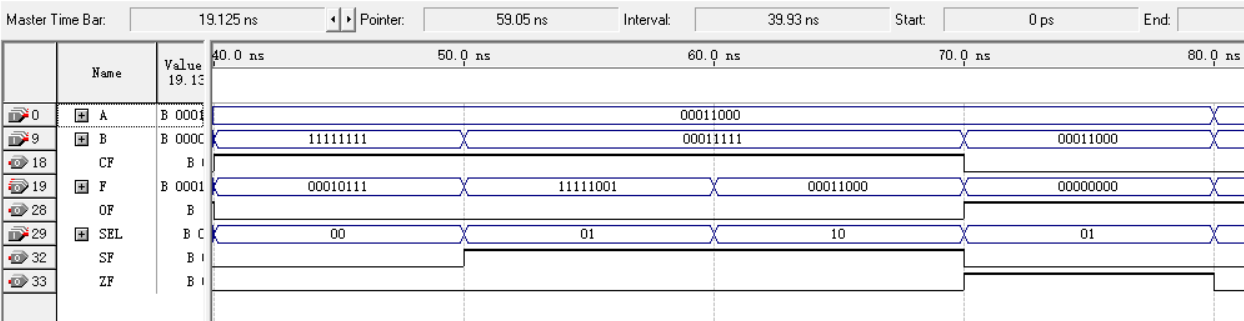


**三、电路正确性验证**

**1、电路仿真**

功能仿真波形图如图：





1. **结果分析**

运算结果完全正确（以下为无符号数）。

A=11000,B=00111,加不进位，减不借位，CF=0,OF=1,SF=0

A=00011000,B=11111111，加进位溢出，CF=1,OF=0

A=00011000,B=00011111,减借位，CF=1,OF=0，SF=1

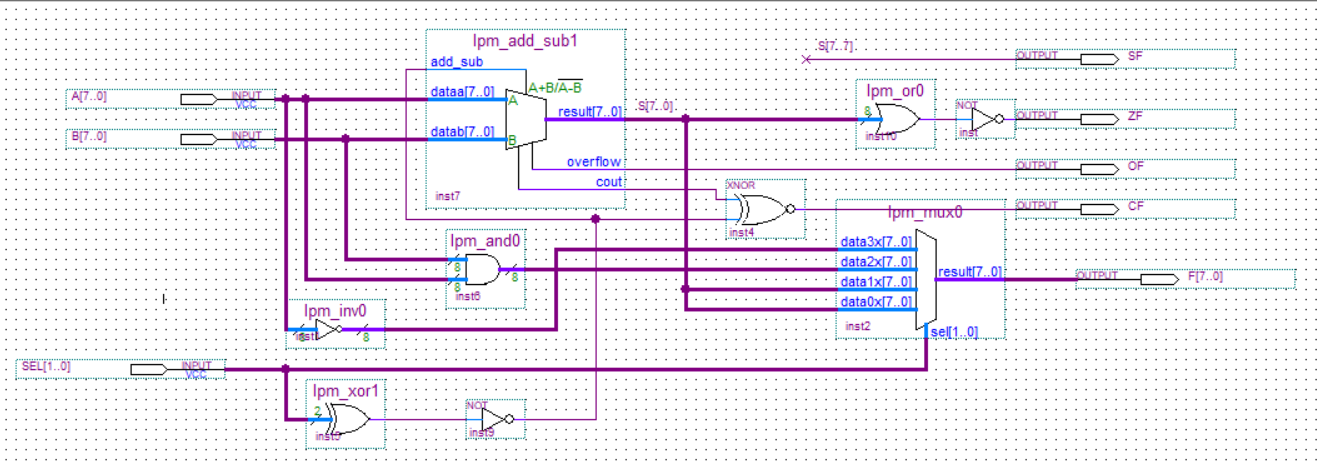
A=00011000,B=00011000,相减结果为0，ZF=0

至此，四个标志位检验完毕。

1. **实验小结**

本次实验无需学习新器件的使用和引脚组成，但需要的逻辑判断较复杂。尤其是CF，减法借位老师提示时还没有感觉，看到举例时才恍然大悟，补码运算的进位意味着原结果不借位。此外，用同或完成加减法运算的选择是非常巧妙耐思考的点，多位数的内部位间运算的引脚编辑也是通过多次尝试得到的实用技巧。动脑良多，也收益多多。一开始都觉得特别难，回头来检查一遍，发现理清了逻辑和符号定义，一切水到渠成。

另外，最初由于疏忽，ZF的理解有误，如下：



该ZF只对加法器有效，后经过对题目的再次理解，最终讨论出应该为整体结果的ZF，更改了电路，即报告最开始的电路图。

**五、教师评语**

教师签字：

日期：