

# 实验报告4：运算部件设计

本实验报告仅作学习使用，不得直接提交！

## 一.4位先行进位加法器设计

### 1.实验顶层方案设计

把进位传递函数  $P_i = X_i + Y_i$ 和进位生成函数 $G_i = X_iY_i$  列入到4位二进制数加法器进位C1-C4的逻辑表达式中，可以得到以下4个先行进位 $C_i$ 的逻辑表达式， $C_i$ 仅与 $X_i$ 、 $Y_i$ 和 $C_0$ 有关，相互间的进位没有依赖关系。只要 $X_1$ - $X_4$ 、 $Y_1$ - $Y_4$ 和 $C_0$ 同时到达，就可几乎同时形成 $C_1$ - $C_4$ ，并同时生成各个数位的和。

$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1G_0 + P_1P_0C_0$$

$$C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

$$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

更多位数的加法器可通过分组的方式来实现，采用组内和组间都并行的进位方式。为了实现组间并行，需要在先行进位部件中输出组间进位生成函数 $G_g$ 和组间进位传递函数 $P_g$ 。

$$P_g = P_3P_2P_1P_0$$

$$G_g = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$$

### 2.实验电路原理图



### 3.实验仿真测试数据



不产生进位的运算



产生进位的运算

## 4.错误现象及分析

在完成实验的过程中，没有遇到任何错误。

# 二.16位先行进位加法器设计

## 1.实验顶层方案设计

对于一个16位加法器，可以分成4组，每组用一个4位先行进位加法器CLA实现。组间使用Pg和Gg进位生成传递函数进行进位传递。

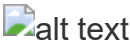


## 2.实验电路原理图



将输入数据分四组分别传入4位并行加法器，然后通过生成的进位函数实现进位功能。

## 3.实验仿真测试数据



不溢出的运算



产生进位的运算

## 4.错误现象及分析

在完成实验的过程中，没有遇到任何错误。

# 三.32 位快速加法器设计

## 1.实验顶层方案设计

通过将两个16位两级先行进位加法器串行级联构建一个 32 位加法器，并根据给出的标志位生成电路原理图，在32位加法器中生成CF、SF、OF、ZF标志位。

溢出标志位OF可用公式 $OF = \neg(A_{(n-1)}) \cdot \neg(B_{(n-1)}) \cdot F_{(n-1)} + A_{(n-1)} \cdot B_{(n-1)} \cdot \neg(F_{(n-1)})$ 来实现。结果

为0标志位ZF的可采用分组分级进行或运算的方式获取，避免扇入输入数量过多的问题。  
在我的实现方案中，使用四输入或非门加与门分级生成ZF标志位。

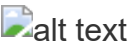


## 2.实验电路原理图

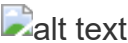


主体部分使用两个16位并行加法器实现电路。标志位部分通过逻辑推演后得出正确公式，然后生成。

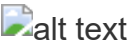
## 3.实验仿真测试数据



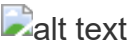
零标志位的生成



不溢出的运算



结果产生符号位的运算



溢出运算

## 4.错误现象及分析

在完成实验的过程中，没有遇到任何错误。

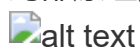
# 四.32位移位器设计

## 1.实验顶层方案设计与实验电路原理图

将实验2中的8位桶形移位器的电路图扩展中32位输入数据，移位位数扩展到5位二进制数，多路选择器增加两级，分别表示移动8位和16位。使用32位的多路选择器，分级实现对移动位数的操作与选择。在移位时，由于不需要保存数据，直接使用分线器进行数据的移位操作。

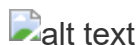


局部原理图



对控制信号的合并

## 2.实验仿真测试数据



一次对数据的循环左移操作

## 3.错误现象及分析

在完成实验的过程中，没有遇到任何错误。

# 五.32位ALU设计

## 1.实验顶层方案设计

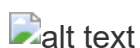


通过ALU操作控制信号生成部件生成ALUctr的信号，以供多路选择器使用。通过联合4.3中实现的32位加法器以及其他逻辑运算部件和移位部件，选择需要的结果输出。对于控制信号本身，采用最小项的最低风险法化简。



## 2.实验电路原理图

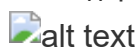
ALU主电路：





ALU控制信号生成单元：




主电路中使用的位扩展器、截取器：




 alt text

 alt text

### 3.实验仿真数据测试

 alt text

加法（无符号）

 alt text

小于

 alt text

逻辑右移

验收时记录的数据测试结果随同实验报告一并上传。


### 4.错误现象及分析

在完成实验的过程中，没有遇到任何错误。


## 思考题

#### 1.32位快速乘法器


选择使用阵列乘法器的算法实现。首先通过实验3中构建的4位快速乘法器作为基础，逐步通过错位相加的方式扩展运算位数。

 alt text

扩展到8位：

 alt text

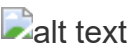
扩展到16位：

 alt text

扩展到32位时，由于Loggism只支持最高32位的数据输出，因此考虑到ALU的输出数据长度统一与实验过程中的数据可读性，只选取低32位数据输出，高32位如果非零则输出Muloverflow标志位。

 alt text

实验仿真数据测试



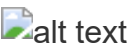
不溢出



溢出

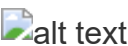
## 2.ALU指令扩展

选择将思考题1中的快速乘法器扩展进ALU中，选择1110为乘法指令。



此条指令对应的操作译码

## 3.32位除法器



选择采用不恢复余数法，第一个时钟直接将低32位被除数放入商寄存器，高32位减除数后作为“试商”结果放入余数寄存器中。随后计数器进行32次循环，在第33次循环时，余数寄存器进行恢复余数操作，商寄存器继续左移一次。为了使数据方便读取，我选择额外进行第34次循环将所有正确数据写入寄存器中，以方便检查结果的准确性。

实验仿真数据测试

初始数据



20次循环结果



最终结果

