

# 华东师范大学计算机科学技术系上机实验报告

课程名称: 计算机组成与结构实践    年级: 17 级  
指导教师: 金健    姓名: 朱桐  
上机实践名称: 十六位机运算器实验    学号: 10175102111  
实践编号: 实验 2    组号: A

上机实践成绩:  
创新实践成绩:  
上机实践日期: 2019/09/27  
上机实践时间: 2 学时

---

## 1 实验目的

1. 掌握十六位机字与字节运算的数据传输格式
2. 验证 ALU 及标志位控制的组合功能
3. 熟悉 ALU 运算控制位的运用

## 2 实验设备

Dais-CMX16<sup>+</sup> 设备一台

## 3 实验内容

使用手动搭接的方法, 实现 ALU 的输入并且完成各种运算输出

## 4 实验原理

### 4.1 数据通路

实验中所用的运算器数据通路如图 4.1 所示。ALU 运算器由 CPLD 描述。运算器的输出经过 2 片 74LS245 三态门与数据总线相连, 2 个运算寄存器 AX、BX 的数据输入端分别由 4 个 74LS574 锁存器锁存, 锁存器的输入端与数据总线相连, 准双向 I/O 输入输出端口用来给出参与运算的数据, 经 2 片 74LS245 三态门与数据总线相连。

### 4.2 运算器功能码

运算器功能码如图 4.2 所示。我们通过设置不同的功能码完成各种不同的运算

## 5 实验步骤

### 5.1 ALU 输入

从 I/O 到总线的输入已经在上一次实验了解过了, 这里与输出到通用寄存器组 CX 开启 RWX 不同的是, 我们开启开关 AWX 和 BWX 控制总线到 ALU 的输入寄存器

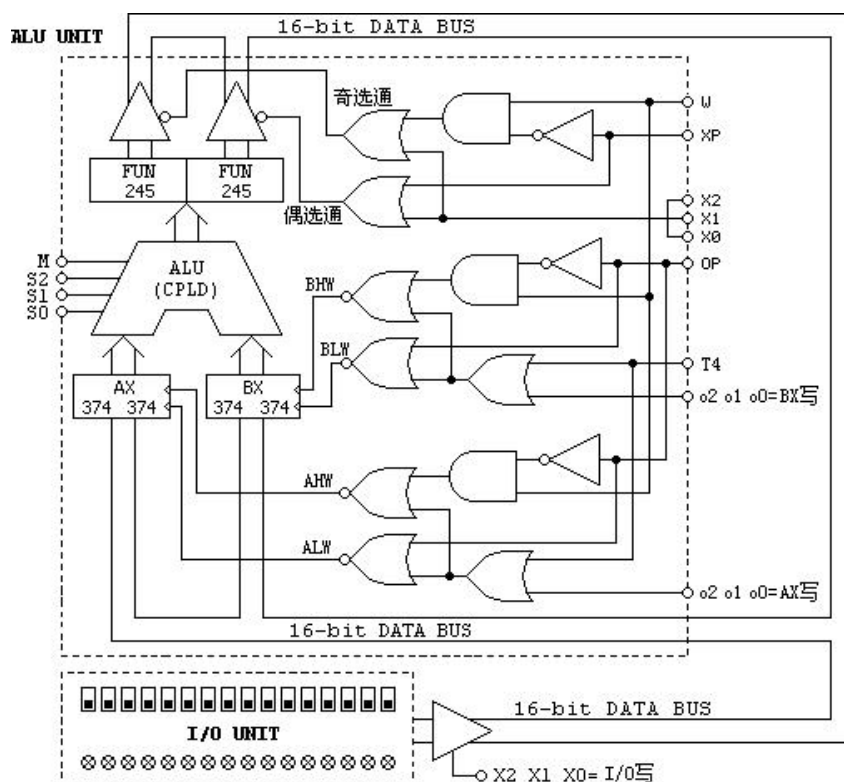


Fig. 4.1: 运算器数据通路

## 5.2 算术运算

### 5.2.1 字运算

- 令  $MS_2S_1S_0 = K15K13K12K_{11} = 1011$ ,  $FUN$  及总线单元显示  $AX \text{ } BX$  的结果
- 令  $MS_2S_1S_0 = K15K13K12K_{11} = 1010$ ,  $FUN$  及总线单元显示  $AX - BX$  的结果

### 5.2.2 字节运算

我们通过控制  $XP, W$  来控制  $AX, BX$  的有效位来进行字节运算

## 5.3 逻辑运算

### 5.3.1 字运算

- 令  $MS_2S_1S_0 = K15K13K12K_{11} = 0010$ ,  $FUN$  及总线单元显示  $AX \& BX$  的结果
- 令  $MS_2S_1S_0 = K15K13K12K_{11} = 0011$ ,  $FUN$  及总线单元显示  $AX | BX$  的结果

### 5.3.2 字节运算

我们通过控制  $XP, W$  来控制  $AX, BX$  的有效位来进行字节运算

算术运算					逻辑运算				
K15	K13	K12	K11	功能	K15	K13	K12	K11	功能
M	S2	S1	S0		M	S2	S1	S0	
1	0	1	0	RR	0	0	0	0	A
1	0	0	1	RL	0	0	0	1	A+1
1	0	1	0	A-B	0	0	1	0	A&B
1	0	1	1	A+B	0	0	1	1	A#B
1	1	0	0	RRC	0	1	0	0	A=0
1	1	0	1	RLC	0	1	0	1	A-1
1	1	1	0	A-B-C	0	1	1	0	/A
1	1	1	1	A+B+C	0	1	1	1	B

Fig. 4.2: 运算器功能码

## 5.4 移位运算

所谓循环移位，就是指移位时数据的首尾相连进行移位，即最高（最低）位的移出位又移入数据的最低（最高）位。根据循环移位时进位位是否一起参加循环，可将循环移位分为不带进位循环和带进位循环两类。其中不带进位循环是指进位“CY”的内容不与数据部分一起循环移位，也称小循环。带进位循环是指进位“CY”中的内容与数据部分一起循环移位，也称大循环。

- 不带进位循环左移：各位按位左移，最高位移入最低位。
- 不带进位循环右移：各位按位右移，最低位移入最高位。
- 带进位循环左移：各位按位左移，最高位移入 C 中，C 中内容移入最低位。
- 带进位循环右：各位按位右移，最低位移入 C 中，C 中内容移入最高位。

循环移位一般用于实现循环式控制、高低字节的互换，还可以用于实现多倍字长数据的算术移位或逻辑移位。

移位运算的过程如图 5.1 所示

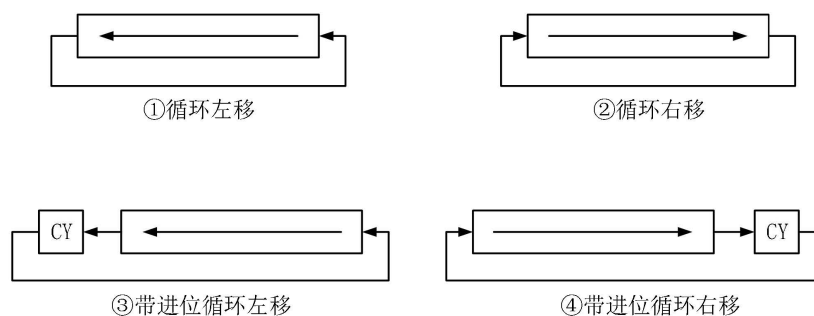


Fig. 5.1: 移位运算过程

### 5.4.1 字运算

完成 AX 输入后，按下表 5.2 输入，再按单排可观察到 AX 的移位变化

K15	K13	K12	K11	功能
M	S2	S1	S0	
1	0	0	0	RR 右移
1	0	0	1	RL 左移
1	1	0	0	RRC 带进位右移
1	1	0	1	RLC 带进位左移

Fig. 5.2: 移位运算功能表

## 5.5 带进位运算

### 5.5.1 进位控制

- 按 [返回] 初始化进位标志  $CY = 0$
- 设置  $CN = 1$ , 改变  $XP, W$
- 按单拍

其中进位控制编码如图 5.3 所示

进位标志位操作					进位	功能说明
K15	K14	K7	K6	DRCK	CY	
M	CN	XP	W			
0	1	1	0	↓	0	清零
0	1	0	0	↓	1	置位
0	1	X	1	↓	/CY	取反

Fig. 5.3: 进位控制编码

### 5.5.2 进位运算

- $X_2X_1X_0 = 001, XPW = 11$
- $M = 1S_2S_1 = 11$
- $S0 = 0$  为减法, 否则为加法

通过带进位运算我们就可以用 16 位的计算机完成 32 位数的运算操作, 先计算低 16 位, 然后保留进位再进行高 16 位的操作。

## 5.6 零标志

进行运算后, Z 标志按钮会随着运算结果变化而变化

## 5.7 ALU 到 BUS 和 BUS 到 IO

我们通过  $X_2X_1X_0 = 001$  来设置总线的输入设备位 ALU, 和上次实验一样, 设置  $IOW = 1$  并且给  $IRCK$  一个脉冲, 便可完成总线到 I/O 的输出。

## 5.8 小结

总结来说，运算控制码如下表 5.4 所示。

运算控制	运算表达式	K15	K13	K12	K11	AX	BX	运算结果
		M	S2	S1	S0			
带进位算术加	A+B+C	1	1	1	1	6655	AA77	FUN=( 10CC )
带借位算术减	A-B-C	1	1	1	0	6655	AA77	FUN=( BBDE )
带进位左移	RLC A	1	1	0	1	6655	AA77	FUN=( CCAA )
带进位右移	RRC A	1	1	0	0			FUN=( )
算术加	A+B	1	0	1	1			FUN=( )
算术减	A-B	1	0	1	0			FUN=( )
左移	RL A	1	0	0	1			FUN=( )
右移	RR A	1	0	0	0			FUN=( )
取 BX 值	B	0	1	1	1	6655	AA77	FUN=( AA77 )
AX 取反	NOT A	0	1	1	0	6655	AA77	FUN=( 99AA )
AX 减 1	A-1	0	1	0	1	6655	AA77	FUN=( 6654 )
清零	0	0	1	0	0			FUN=( )
逻辑或	A OR B	0	0	1	1			FUN=( )
逻辑与	A AND B	0	0	1	0			FUN=( )
AX 加 1	A+1	0	0	0	1			FUN=( )
取 AX 值	A	0	0	0	0			FUN=( )

Fig. 5.4: ALU 运算器真值表

## 6 调试过程、结果与分析

### 6.1 实验要求

1. 假设 CY 原来有值，在进行带进位加法时，加法结果可能会产生新的进位，也可能不产生。请分析这种情况，设计实验验证各种情况下加法完成后 CY 的值
2. 输入两个 16 位数 FFFE 和 0003，输出进位加法结果

### 6.2 分析

我们已经知道了手动控制 CY，和控制 CY 是否带入运算的方法，且 CY 于加法表示加法进位，减法表示减法进位。因此我们可以推断出

- 加法带进位运算  $MS_2S_1S_0 = 1111$ ，若原来  $AX = 0001, BX = 0003, CY = 1$ ，则结果  $DBUS = FUN = 0005$ ，不产生新的进位
- 加法带进位运算  $MS_2S_1S_0 = 1111$ ，若原来  $AX = 000F, BX = 000F, CY = 1$ ，则结果  $DBUS = FUN = 000F$ ，产生新的进位

- ### 6.3 结果

6

## 7 总结

第二次试验总结以下经验

1. 在开始之前最好首先了解每个开关的意义，帮助理解并且减少失误
2. 开关最好插在下方对应表示的开关口中，剩余的按照表格填入。这样方便记忆每个插口的位置
3. 每次开始之前先初始化
4. 仔细阅读实验指导书
5. 实验箱上每个模块的奇偶位输入都有对应的指示灯，方便指认电路是否连接错误

## 8 附件

无