

安徽大学电子信息工程学院

微机原理 实验报告

学号P12314000
实验日期10月29日

专业电子信息工程
实验成绩

姓名电子出森
教师签字

算术逻辑运算器实验

1 实验目的

1. 深入了解算术逻辑运算器 (74LS181) 的内部结构、功能特性及其在数字系统中的应用。通过实际操作，掌握该芯片的基本工作原理和功能实现机制。
2. 系统学习基本算术运算（如加法、减法）和逻辑运算（如与、或、非、异或等）在硬件层面的实现方法。理解计算机系统中运算器的核心作用，为后续学习计算机组成原理和体系结构打下坚实基础。
3. 通过对比有符号数和无符号数的运算结果，深入理解补码表示法在计算机运算中的重要性，掌握标志位（如进位标志 CF、符号标志 SF 等）的含义和判断方法。

2 实验原理

本实验原理分为两大部分：第一部分介绍实验所需的核心器件及其功能特性；第二部分说明实验电路的组成与工作原理。

2.1 实验器件

本实验使用的主要器件如表 1所示。

表 1: 实验器件清单

序号	器件名称	型号
1	算术逻辑运算器	74LS181
2	三态门	74LS244
3	地址/数据锁存器	74LS273
4	反相器	74LS04
5	或/或非门	74HC4078

1. 算术逻辑运算器（74LS181）

74LS181 是一款经典的 4 位算术逻辑运算器 (ALU)，是理解计算机运算器设计的典型器件。

其主要功能包括：

- 具有两个 4 位数据输入端（A3-A0, B3-B0）和一个 4 位数据输出端（F3-F0）
- 通过 4 个功能选择端（S3-S0）和方式控制端（M）选择运算类型
- 当 M=1 时执行逻辑运算（与、或、非、异或等 16 种逻辑操作）
- 当 M=0 时执行算术运算（加法、减法等 16 种算术操作）
- 提供进位输入（Cn）和进位输出（Cn+4），支持多片级联实现更高位数运算

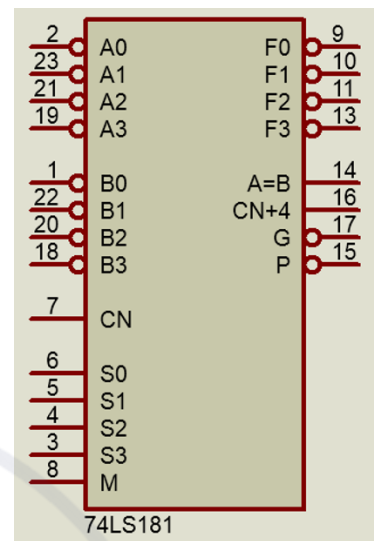


图 1: 74LS181 引脚图

其完整的逻辑功能表如表 2 所示。

表 2: 74LS181 逻辑功能表

选择输入				M=0（算术运算）		M=1
S3	S2	S1	S0	CN=1 无进位	CN=0 有进位	（逻辑运算）
0	0	0	0	$F = A$	$F = A \text{ 加 } 1$	$F = \bar{A}$
0	0	0	1	$F = A + B$	$F = (A + B) \text{ 加 } 1$	$F = \bar{A} + \bar{B}$
0	0	1	0	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ 加 } 1$	$F = \bar{A}B$
0	0	1	1	$F = 0 \text{ 减 } 1$	$F = 0$	$F = 0$
0	1	0	0	$F = A \text{ 加 } \bar{A}\bar{B}$	$F = A \text{ 加 } \bar{A}\bar{B} \text{ 加 } 1$	$F = \bar{A}\bar{B}$
0	1	0	1	$F = (A + B) \text{ 加 } \bar{A}\bar{B}$	$F = (A + B) \text{ 加 } \bar{A}\bar{B} \text{ 加 } 1$	$F = \bar{B}$
0	1	1	0	$F = A \text{ 减 } B \text{ 减 } 1$	$F = A \text{ 减 } B$	$F = A \oplus B$
0	1	1	1	$F = \bar{A}\bar{B} \text{ 减 } 1$	$F = \bar{A}\bar{B}$	$F = \bar{A}\bar{B}$
1	0	0	0	$F = A \text{ 加 } AB$	$F = A \text{ 加 } AB \text{ 加 } 1$	$F = A + B$
1	0	0	1	$F = A \text{ 加 } B$	$F = A \text{ 加 } B \text{ 加 } 1$	$F = A \oplus B$
1	0	1	0	$F = (A + \bar{B}) \text{ 加 } AB$	$F = (A + \bar{B}) \text{ 加 } AB \text{ 加 } 1$	$F = B$
1	0	1	1	$F = \bar{A}B \text{ 减 } 1$	$F = \bar{A}B$	$F = AB$
1	1	0	0	$F = A \text{ 加 } A$	$F = A \text{ 加 } A \text{ 加 } 1$	$F = 1$
1	1	0	1	$F = (A + B) \text{ 加 } A$	$F = (A + B) \text{ 加 } A \text{ 加 } 1$	$F = A + \bar{B}$
1	1	1	0	$F = (A + \bar{B}) \text{ 加 } A$	$F = (A + \bar{B}) \text{ 加 } A \text{ 加 } 1$	$F = A + B$
1	1	1	1	$F = A \text{ 减 } 1$	$F = A$	$F = A$

2. 三态门（74LS244）

74LS244 是一款八路三态缓冲器，用于总线隔离和数据传输控制：

- 包含 8 个独立的缓冲器，分为两组，每组 4 个
- 每组有独立的使能控制端（1G、2G），低电平有效
- 输出端具有三态特性：高电平、低电平和高阻态
- 高阻态时输出端呈现高阻抗，可实现多个器件共享数据总线

- 主要用于隔离运算器与数据总线，防止总线冲突

其逻辑功能表如表 3 所示。

表 3: 74LS244 逻辑功能表

INPUTS		OUTPUT
1G, 2G	D	
L	L	L
L	H	H
H	X	H (Z)

注：L 表示低电平，H 表示高电平，X 表示任意电平，Z 表示高阻态。

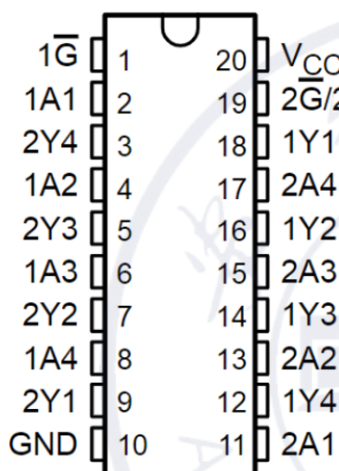


图 2: 74LS244 引脚图

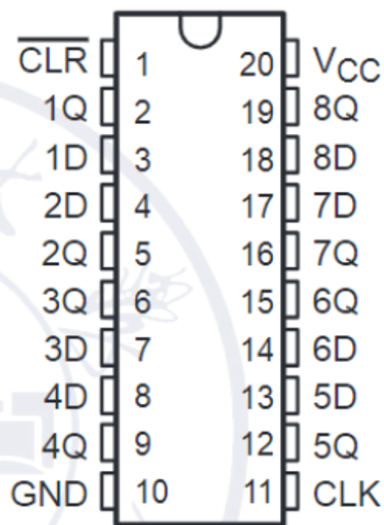


图 3: 74LS273 引脚图

3. 地址/数据锁存器（74LS273）

74LS273 是一款八路 D 触发器，用于数据锁存和同步：

- 包含 8 个独立的 D 触发器，共用一个时钟信号（CLK）
- 在时钟上升沿锁存 D 端输入数据，并保持在 Q 端输出
- 具有异步清零端（CLR），低电平有效，可将所有输出清零
- 在本实验中用于锁存地址信号或数据信号，确保数据稳定传输

其逻辑功能表如表 4 所示（每个触发器）。

表 4: 74LS273 逻辑功能表（每个触发器）

INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q ₀

注：L 表示低电平，H 表示高电平，X 表示任意电平， \uparrow 表示时钟上升沿， Q_0 表示保持原状态。

4. 反相器（74LS04）

74LS04 是一款六路反相器，用于信号取反：

- 包含 6 个独立的非门，每个非门有一个输入端和一个输出端
- 输出信号与输入信号相反：输入高电平时输出低电平，反之亦然
- 在本实验中用于产生控制信号的反相信号，或进行逻辑电平转换

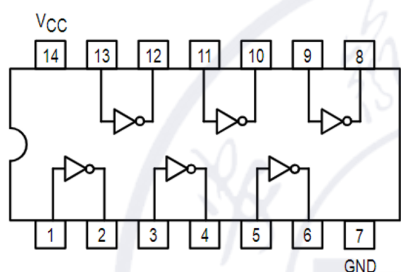


图 4: 74LS04 引脚图

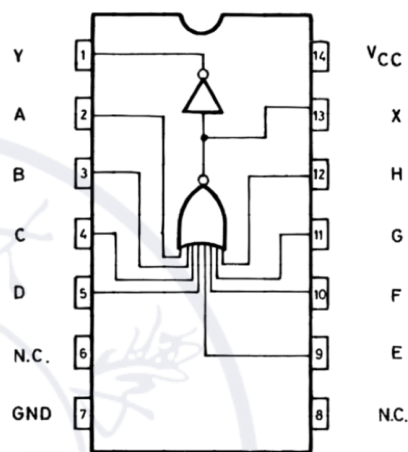


图 5: 74HC4078 引脚图

5. 或/或非门（74HC4078）

74HC4078 是一款八输入或/或非门，用于多路信号的逻辑运算：

- 具有 8 个输入端，可实现 8 路信号的或运算或或非运算
- 当任意输入端为高电平时，或门输出为高电平
- 或非门输出为或门输出的反相
- 在本实验中用于产生控制信号或进行多路信号判断

2.2 实验电路

本实验电路以 74LS181 算术逻辑运算器为核心，构建了一个完整的 8 位运算系统。整个电路可分为运算器核心电路、数据输入输出电路、控制信号电路和标志位检测电路四个部分。在运算器核心电路中，A、B 两组输入数据通过地址/数据锁存器（74LS273）锁存后送入 74LS181 的数据输入端，功能选择信号（S3-S0）和方式控制信号（M）由外部控制电路提供，用于选择具体的运算类型。数据输入输出部分采用 74LS273 锁存器确保输入数据的稳定性，运算结果则通过 74LS244 三态门输出到数据总线，其使能控制确保在需要时才将数据送上总线以避免总线冲突。控制信号电路使用 74LS04 反相器产生所需的反相控制信号，

74HC4078 或/或非门用于组合多个控制信号产生复杂的控制逻辑，包括锁存器时钟信号、三态门使能信号等。标志位检测电路负责产生运算结果的状态标志，其中进位标志（CF）由 74LS181 的进位输出端（Cn+4）直接产生，零标志（ZF）通过或非门检测运算结果是否全为 0，符号标志（SF）则直接取运算结果的最高位（F3）。

实验电路的整体结构如图 6 所示。

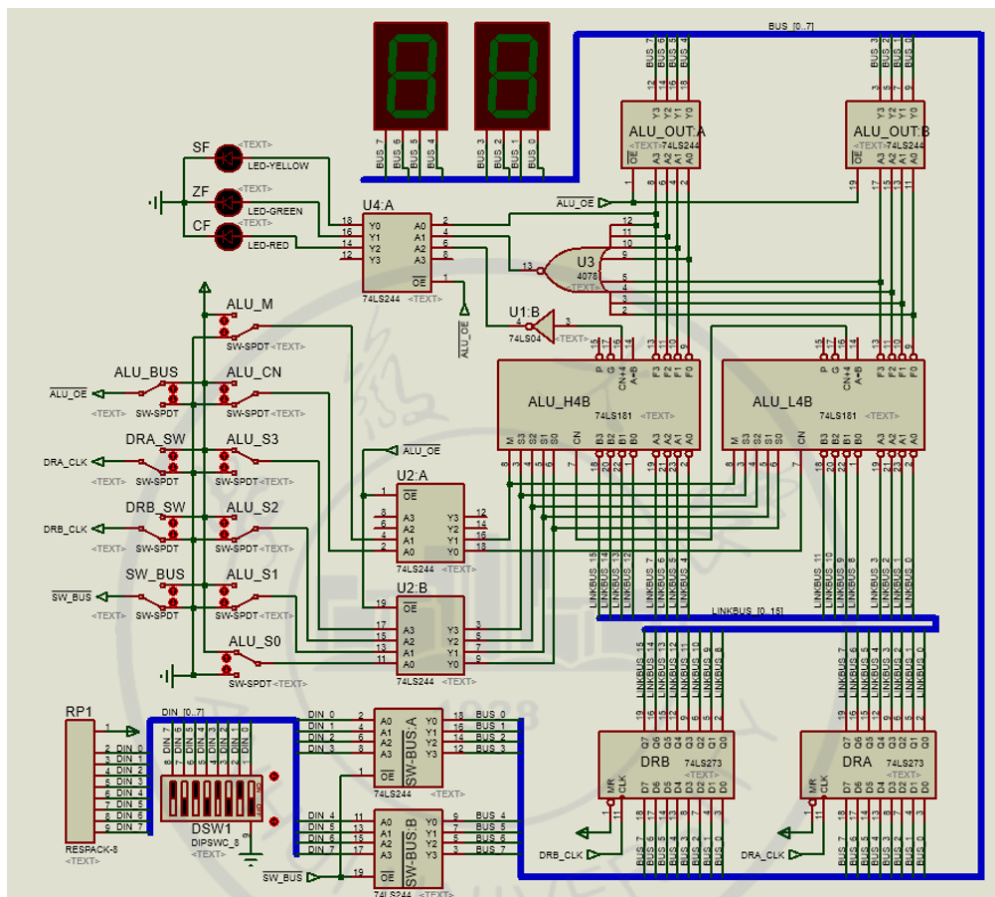


图 6: 实验电路原理图

控制信号的不同组合对应不同的运算功能：

- 算术运算模式（**M=0**）：主要执行加法、减法、加 1、减 1 等算术操作
- 逻辑运算模式（**M=1**）：主要执行与、或、非、异或等逻辑操作
- 功能选择（**S3-S0**）：具体定义运算的类型和特性

3 实验内容与数据处理

3.1 实验步骤详述

1. DRA_CLK=0, DRB_CLK=0 等待数据所存, SW_BUS=0 等待数据输入显示
ALU_OE=1 关闭运算器输出

2. 通过波拨码开关输入数据，此时两位十六进制数码管显示输入数据，输入后通过将 $\overline{\text{DRA_CLK}}$ 置为 1 将当前 2 位十六进制数所存到 DRA, 重复上述步骤，输入数据后将 $\overline{\text{DRB_CLK}}$ 置为 1 将当前 2 位十六进制数所存到 DRB
3. 通过设置 $\overline{\text{ALU_S0}}$ 、 $\overline{\text{ALU_S1}}$ 、 $\overline{\text{ALU_S2}}$ 、 $\overline{\text{ALU_S3}}$ 选择运算功能, 通过设置 $\overline{\text{ALU_M}}$ 选择算术运算或逻辑运算，通过设置 $\overline{\text{ALU_CN}}$ 选择有进位或无进位运算
4. 将 $\overline{\text{SW_BUS}}=1$ ，失能数据输入显示, 将 $\overline{\text{ALU_OE}}=0$ ，使能运算器输出观察并记录运算结果及标志位状态

3.2 实验一：无符号数运算验证

令电路原理图中各个开关的初始状态为: $\overline{\text{DRA_CLK}}=\overline{\text{DRB_CLK}}=0, \overline{\text{SW_BUS}}=\overline{\text{SW_BUS}}=1$, $(\text{S3}, \text{S2}, \text{S1}, \text{S0}, \text{M}, \text{CN})=(1, 1, 1, 1, 1, 1)$ 。操作拨码开关，向数据暂存器 DRA 写入 AAH, DRB 写入 55H (即 $\text{A}=0\text{xAAH}$, $\text{B}=0\text{x55H}$)。改变运算器的控制信号 $(\text{S3}, \text{S2}, \text{S1}, \text{S0}, \text{M}, \text{CN})$ 的组合，运算器使能 ($\overline{\text{SW_BUS}}=0$)，观察运算器的输出和标志位，并填入下表中，与理论值比较，验证 74LS181 的功能。

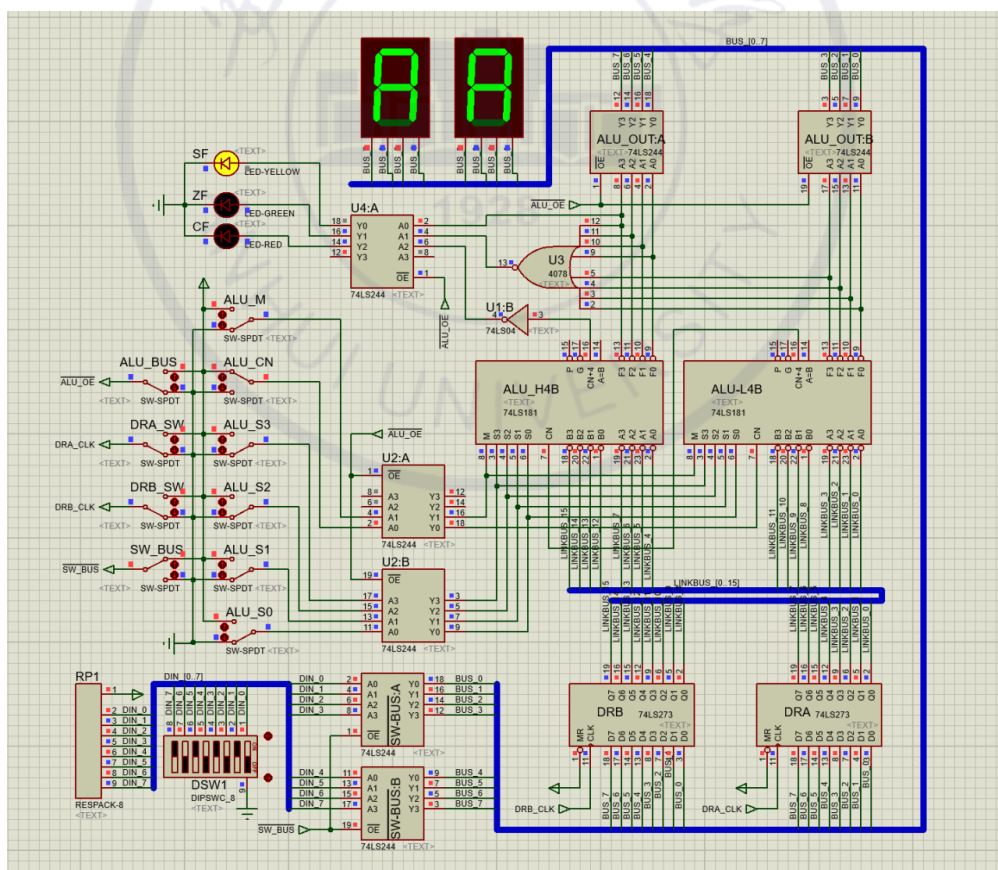


图 7: 实验一截图示例

例：先将 AAH 锁存至 DRA，再将 55H 锁存至 DRB。设置 S3S2S1S0 为 0000， $\overline{\text{ALU_CN}}$ 设置为 1 无进位，M 设置为 0，此时 ALU 执行加法运算，输出结果为 AAH。

例：先将 7AH 锁存至 DRA，再将 8BH 锁存至 DRB。设置 S3S2S1S0 为 0011，CN 设置为 0，M 设置为 0，此时 ALU 执行加法运算，输出结果为 -5H（FBH）。标志位 CF/ZF/SF = 110。实验结果截图如图 9 所示。

实验结果记录如下图表 10。

DRA	DRB	S3	S2	S1	S0	M=0 (算术运算)		标志位		M=1
						CN=1 无进位	CN=0 有进位	CF/ZF/SF	逻辑运算	
7AH	8BH	0	0	0	0	F= 7A	F= 7B	000 000	F= 85	
		0	0	0	1	F= FB	F= EC	001 001	F= 04	
		0	0	1	0	F= 7E	F= 7F	000 000	F= 81	
		0	0	1	1	F= FE	F= 00	001 110	F= 00	
		0	1	0	0	F= FA	F= EB	001 001	F= F5	
		0	1	0	1	F= 6B	F= 6C	100 100	F= 74	
		0	1	1	0	F= EE	F= EF	001 001	F= F1	
		0	1	1	1	F= 6F	F= 70	100 100	F= 70	
		1	0	0	0	F= 84	F= 85	001 001	F= 8F	
		1	0	0	1	F= 85	F= 86	100 100	F= 0E	
		1	0	1	0	F= 88	F= 89	001 001	F= 8B	
		1	0	1	1	F= 09	F= 0A	100 100	F= 0A	
		1	1	0	0	F= F4	F= F5	001 001	F= FF	
		1	1	0	1	F= 75	F= 76	100 100	F= 7E	
		1	1	1	0	F= F8	F= F9	001 001	F= FB	
		1	1	1	1	F= 79	F= 7A	100 100	F= 7A	

图 10: 实验二结果记录表格示例

4 思考与讨论

4.1 思考题

1. 74LS181 组成的运算器通路，能否区分有符号数运算和无符号数运算？两者的运算过程有何不同？各自的数值表示范围是多少？

74LS181 本身为组合逻辑 ALU，不区分数据的语义（有符号/无符号）——它只按二进制规则进行位运算与进位处理；两者的运算“过程”在硬件层面相同，但解释结果时不同。数值范围（以 8 位为例）：无符号数 0..255 ($2^8 - 1$)，有符号（补码）为 -128..+127。

2. 在 74LS181 组成的运算器通路中，输入锁存器 DRA、DRB 的作用是什么？运算结果输出端连接的 74LS244 缓冲器的作用是什么？若去掉其中一个输入锁存器（使 74LS181 输入直连总线），运算器还能正常工作吗？若去掉输出端 74LS244（使 74LS181 输出直连总线），运算器还能正常工作吗？

DRA/DRB 作为输入锁存器的作用是：数据同步与保持、时序隔离、避免在运算中总线变化导致读入不稳定数据。74LS244 作为输出缓冲器的作用是：提高驱动能力、实现三态输出以避免总线冲突、提供输出使能控制和信号整形。若去掉输入锁存器，74LS181 在理论上仍能工作（单片、总线上无其它驱动时），但在有总线共享或时序变化时会出现数据不稳定与竞态，实验系统将变得不可靠；若去掉输出缓冲器使输出直接连总线，若总线上只有该器件且不存在其他驱动，短时间内可能观察到结果，但在多设备系统中会产生总线冲突、驱动能力不足或损坏器件，故不可行。

3. 在 74LS181 进行无符号数运算时，标志位 SF 是否有意义？在有符号数运算时，CF 的含义是否与无符号运算时一致？举例说明：做有符号数的减法（例如“A 减 0”）时为何 CF 可能置位；做“A 加 0”时 CF 会置位吗？在什么情况下有符号数的加法会出现 CF 置位？

在无符号运算中，SF（最高位）仅表示最高位的电平，对数值正负无意义，因此不能当作无符号溢出判据；有符号运算的溢出应由溢出标志（OF）判断，而不是单纯看 CF。CF 表示无符号运算的进位/借位（对有符号意义不同）。例如“A 减 0”：减法通常由加上被减数与被减数的补码实现（ $A + (-0)$ ），在进位链上可能产生进位/借位，从而使 CF 置位；“A 加 0”通常不会产生向更高位的进位，因此 CF 一般不置位。对于有符号数，加法发生溢出（OF 置位）的典型情况为：两个正数相加得到负数，或两个负数相加得到正数；此时 CF 可能为 1（若有进位穿出最高位），但判断有符号溢出应以 OF 为准，CF 与 OF 的意义并不完全一致。

4.2 有符号数与无符号数运算的本质区别

通过对实验结果的深入分析，我们可以得出以下重要结论：

运算过程的同一性：74LS181 在硬件层面执行运算时，并不区分有符号数和无符号数。无论是哪种数据类型，运算器都按照相同的二进制规则进行处理。这种设计体现了计算机体系结构的一个重要原则：数据的解释权在于程序员或编译器，硬件只负责执行基本的二进制操作。

结果解释的差异性：

- 无符号数：结果直接解释为二进制数值，溢出通过进位标志 CF 判断
- 有符号数：结果采用补码解释，溢出通过符号标志 SF 和溢出标志判断

数值范围对比：

- 8 位无符号数范围：0 到 255 ($2^8 - 1$)
- 8 位有符号数范围：-128 到 +127（采用二进制补码表示）

这种范围差异直接影响了运算结果的正确性判断标准。

4.3 标志位系统的深入理解

无符号数运算中的 SF 标志：在无符号数运算中，符号标志 SF 实际上没有实际意义。因为无符号数的最高位是数值位而非符号位，SF 反映的只是最高位的状态，不能作为数值正负的判断依据。

有符号数运算中的 CF 标志：

- CF 标志在有符号数运算中的含义与无符号数不同

- 对于有符号数，CF 不能直接作为溢出判断依据
- 需要结合溢出标志 OF 进行综合判断

特殊运算情况分析：

1. **“A 减 0”操作：**在补码运算中，减法通过加补码实现。A 减 0 等价于 A 加 0 的补码，由于 0 的补码是全 0，但进位处理机制可能导致 CF 置位，这反映了补码运算的特殊性。
2. **“A 加 0”操作：**通常不会使 CF 置位，因为不会产生向更高位的进位。
3. **有符号数加法的 CF 置位条件：**当两个正数相加得到负数，或两个负数相加得到正数时，表明发生了溢出，此时 CF 可能置位，但更重要的是要检查 OF 标志。

本次实验为我们后续学习计算机组成原理和体系结构奠定了重要的实践基础，对理解现代处理器中 ALU 的设计思想具有重要启发意义。

