

长安大学

# 学生实验报告

实验课名称：计算机组成原理

实验项目名称：基本运算器实验

专业名称：人工智能

班级：2022240401

学号：2022905226

学生姓名：

同组人员姓名：

教师姓名：

2024 年 10 月 17 日

## 一、实验名称：

基本运算器实验

## 二、实验目的：

- (1) 了解运算器的组成结构。
- (2) 基于数据通路图，观测并分析运算器的工作原理。
- (3) 基于信号时序图，观测并分析运算器的工作原理。

## 三、实验内容及要求：

- (1) 本机运行实现基本运算器的各种逻辑运算和算数运算，并通过指示灯观察验证结果。
- (2) 联机运行实现基本运算器的各种逻辑运算和算数运算，并通过上位机观察验证运算结果的信号时序图。
- (3) 基于信号时序图，观测并分析运算器的工作原理。

## 四、实验原理简述：

运算器内部含有三个独立运算部件，分别为**算术、逻辑和移位**运算部件。

要处理的数据存于暂存器 A 和暂存器 B，三个部件同时接受来自 A 和 B 的数据。

各部件对操作数进行何种运算由**控制信号**  $S_3 \cdots S_0$  和 CN 来决定。

任何时候，多路选择开关只选择三部件中一个部件的结果作为 ALU 的输出。如果是影响进位的运算，还将置进位标志 FC，在运算结果输出前，置 ALU 零标志。

ALU 中所有模块集成在一片 CPLD 中。

数据流动过程：

### a. 输入阶段：

- 操作数准备：要处理的数据从 CPU 寄存器或内存中加载到暂存器 A 和暂存器 B 中。
- 数据分发：暂存器 A 和 B 中的数据同时被发送到三个独立的运算部件(算术、逻辑和移位运算部件)。
- 控制信号设置：控制单元根据当前指令设置控制信号  $S_3 \cdots S_0$  和 CN，为下一阶段的运算做准备。

### b. 运算阶段：

- 并行计算：三个运算部件同时开始各自的运算过程。
  - \* 算术部件：执行加法、减法等算术运算。

- \* 逻辑部件：执行与、或、异或等逻辑运算。
- \* 移位部件：执行左移、右移等移位操作。
- 中间结果生成：每个部件根据输入的操作数和控制信号生成各自的运算结果。
- 标志位计算：在运算过程中，各部件同时计算可能影响的标志位（进位、溢出等）。

c. 选择阶段：

- 结果选择：多路选择开关根据控制信号（可能是  $S3...S0$  的某些位）从三个运算部件的结果中选择一个。
- 数据通路建立：选中的结果被导向 ALU 的输出端口。
- 非选中结果处理：未被选中的运算结果被丢弃或保存，视具体实现而定。

d. 输出阶段：

- 结果输出：选中的运算结果被发送到 ALU 的输出端口，准备被写入目标寄存器或内存。
- 标志位设置：
  - \* 如果是影响进位的运算（如加法），则设置进位标志 FC。
  - \* 检查运算结果，如果为零则设置零标志。
  - \* 其他相关标志（溢出标志、符号标志等）也在此时被更新。
- 状态更新：ALU 将新的标志位状态通知给控制单元，为后续的条件判断和分支指令提供依据。

数据处理流程如图 1 所示：

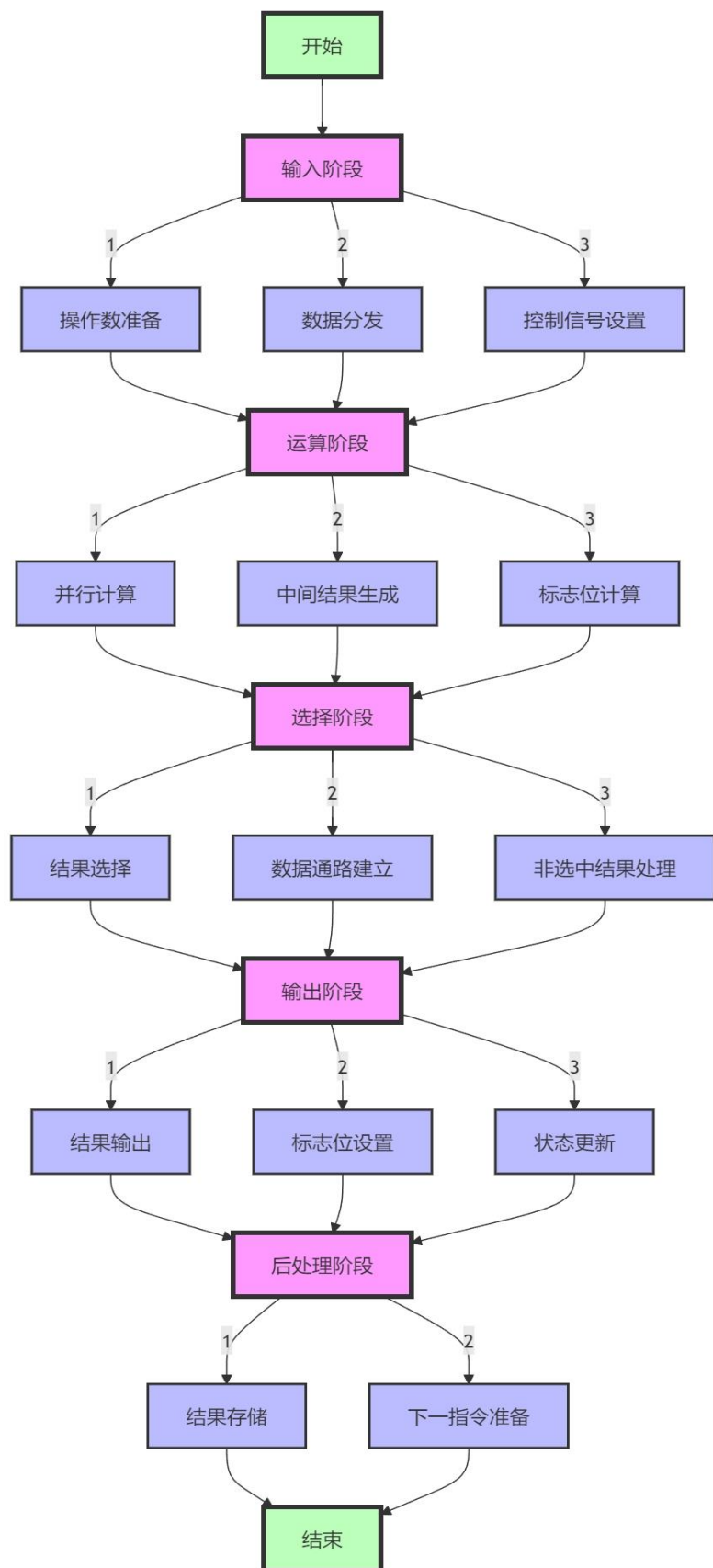


图 1：数据处理流程图

运算器原理图如图 2 所示：

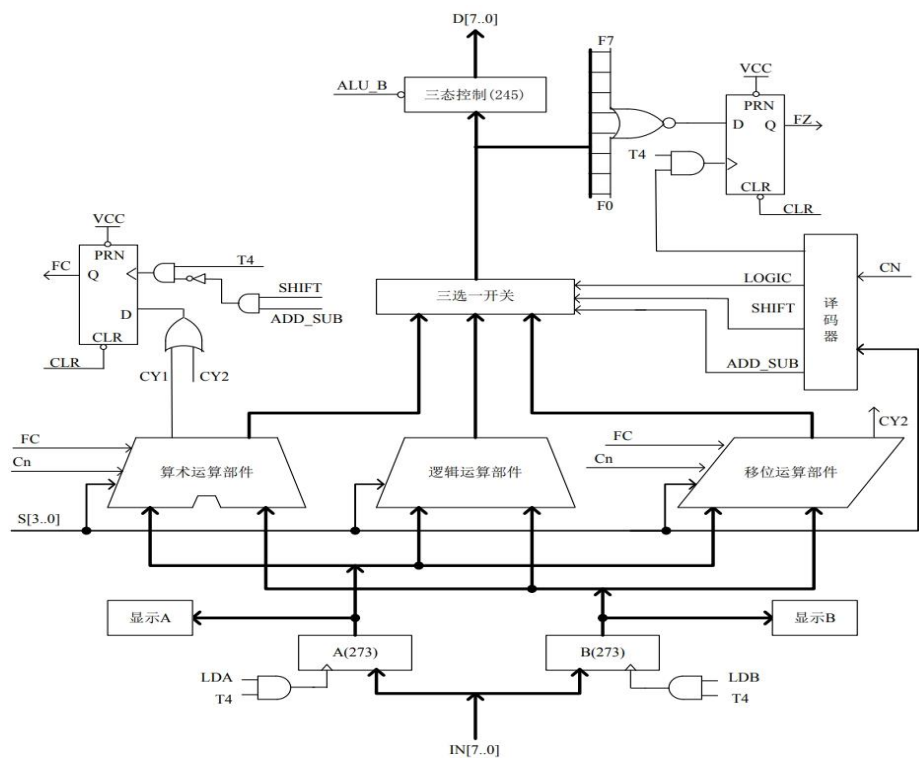


图 2：运算器原理图

控制信号  $S_3 \sim S_0$  控制的运算对应表如下：

运算类型	$S_3 S_2 S_1 S_0$	CN	功 能
逻辑运算	0000	X	$F=A$ （直通）
	0001	X	$F=B$ （直通）
	0010	X	$F=AB$ (FZ)
	0011	X	$F=A+B$ (FZ)
	0100	X	$F=\neg A$ (FZ)
移位运算	0101	X	$F=A$ 不带进位循环右移 B（取低 3 位）位 (FZ)
	0110	0	$F=A$ 逻辑右移一位 (FZ)
		1	$F=A$ 带进位循环右移一位 (FC, FZ)
	0111	0	$F=A$ 逻辑左移一位 (FZ)
		1	$F=A$ 带进位循环左移一位 (FC, FZ)
算术运算	1000	X	置 $FC=CN$ (FC)
	1001	X	$F=A$ 加 B (FC, FZ)
	1010	X	$F=A$ 加 B 加 FC (FC, FZ)
	1011	X	$F=A$ 减 B (FC, FZ)
	1100	X	$F=A$ 减 1 (FC, FZ)
	1101	X	$F=A$ 加 1 (FC, FZ)
	1110	X	（保留）
	1111	X	（保留）

图 3：控制信号  $S_3 \sim S_0$  控制的运算对应表

五、实验设备及工具：

TDX-CMX 系统  
上位机电脑端

六、实验过程详述：

一. 本机运行

1. 接线。把时序与操作台单元的“MODE”用短路块短接，使系统工作在四节拍模式，JP1 用短路块将 1、2 短接，按图4连接实验电路。

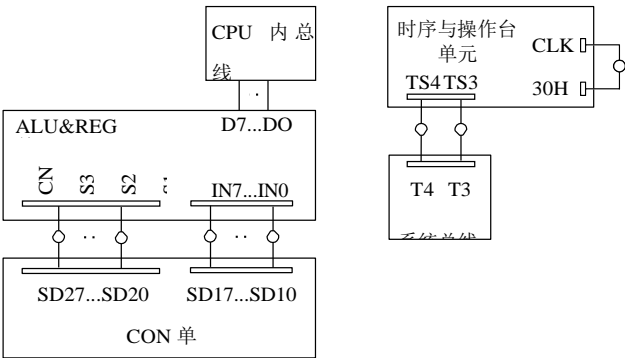


图 4：系统接线图

2. 将时序与操作台单元的开关 KK2 置为‘单拍’档，开关 KK1、KK3 置为‘运行’档。置 ALU\_B=1，关闭运算器输出。

3. 打开电源，如果听到有‘嘀’报警声，说明有总线竞争，应立即关闭电源，重新检查接线，直到错误排除。然后按动 CON 单元的 CLR 按钮，将运算器的 A、B 和 FC、FZ 清零。

4. 用输入开关向暂寄存器 A 置数。

按动 2 次时序单元的 ST 按钮，产生 T1、T2 节拍后，拨动 CON 单元的 SD17...SD10 数据开关，形成二进制数 01100101

置 LDA=1，LDB=0。按动 2 次 ST 按钮产生 T3、T4 节拍，则将二进制数 01100101 置入暂寄存器 A 中，暂寄存器 A 的值通过 ALU 单元的 A7...A0 八位 LED 灯显示。

5. 用输入开关向暂寄存器 B 置数。

按动 2 次时序单元的 ST 按钮，产生 T1、T2 节拍后，拨动 CON 单元的 SD17...SD10 数据开关，形成二进制数 10100111（或其它数值）

置 LDA=0，LDB=1，按动 2 次 ST 按钮产生 T3、T4 节拍，则将二进制数 10100111 置入暂寄存器 B 中，暂寄存器 B 的值通过 ALU 单元的 B7...B0 八位 LED 灯显示。

6. 改变运算器的功能设置，观察运算器的输出。

按动 2 次时序单元的 ST 按钮，产生 T1、T2 节拍后，置 ALU\_B=0、LDA=0、

LDB=0, 然后按表3置S3、S2、S1、S0和Cn的数值,并观察数据总线LED显示灯显示的结果。

实验过程中将输入置为  $A = 01100101$ ,  $B = 10100111$ , 置控制信号为 0001。结果如下图,可见控制信号为 0001 时,直接将 B 输出,无进位。

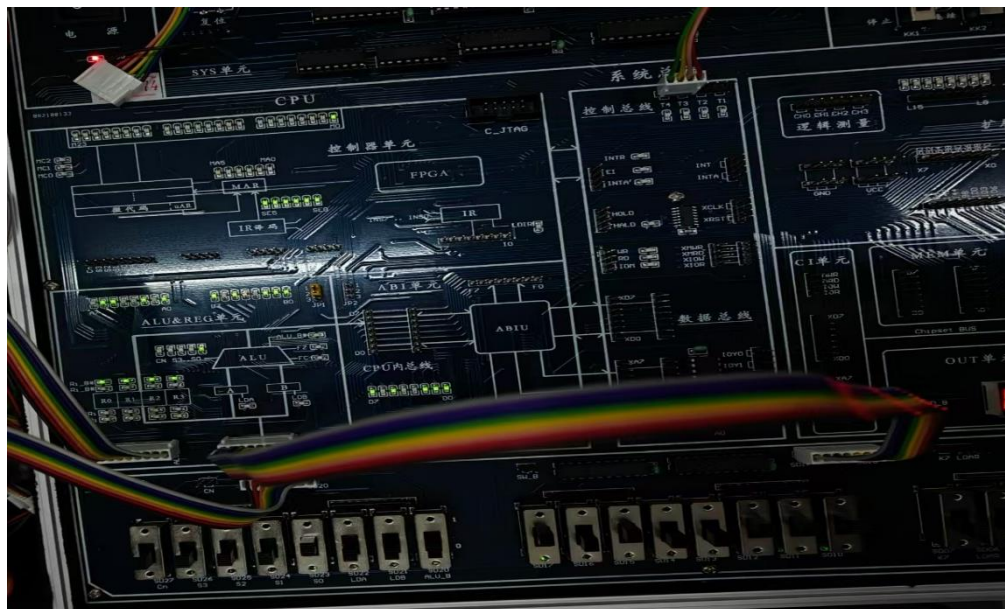


图 5: 本机实验过程示例

## 二. 联机运行

实验箱和 PC 联机操作,可通过软件中的数据通路图来观测实验结果,也可通过软件中的信号时序图来观测实验结果。

### (1) 观测数据通路图

打开 TDX-CMX 软件,选择联机软件的“【实验】—【运算器实验】”,打开运算器实验的数据通路图,如图6所示。

操作方法同本机运行,每按动一次 ST 按钮,数据通路图会有数据的流动,反映当前运算器所做的操作,或在软件中选择“【调试】—【单节拍】”,其作用相当于将时序单元的状态开关 KK2 置为‘单拍’档后按动了一次 ST 按钮,数据通路图也会反映当前运算器所做的操作。

重复上述操作, 改变 A、B 的值, 可以得出各种运算结果并验证 FC、FZ 的锁存功能。

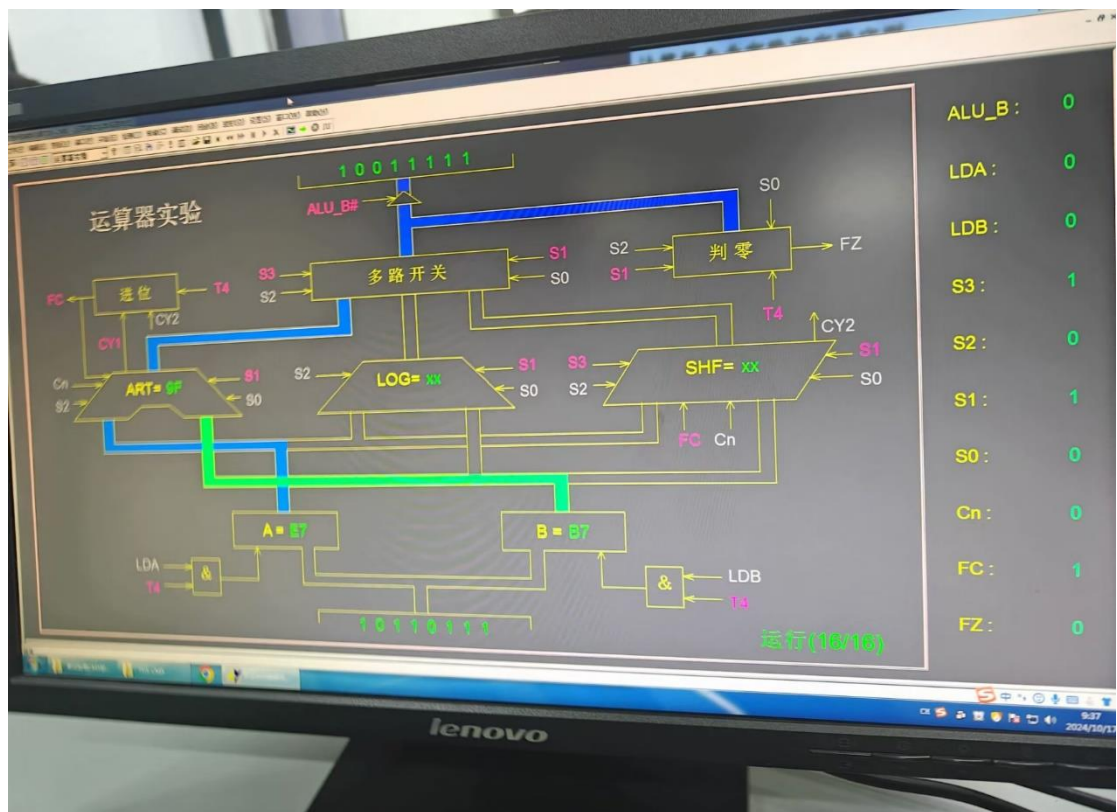



图6: 数据通路图

## (2) 观测信号时序图

打开通路图。再点击  打开选择观察信号窗口, 选择想要观察的信号, 即可观察时序图。

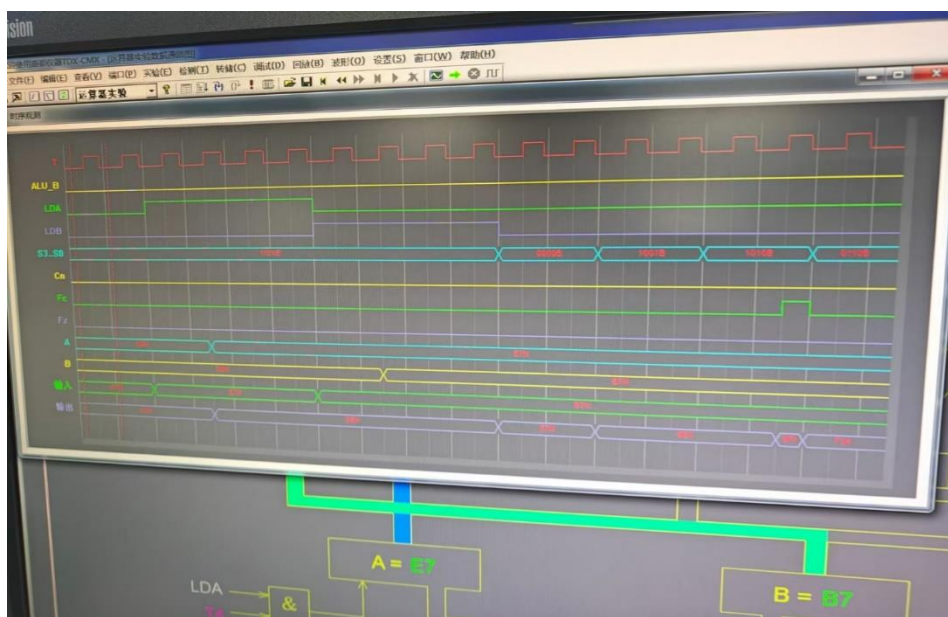


图7: 观察信号时序



输入A = E7, B = B7。观察上图，可知暂存器 A 的控制信号 LDA 在第一个机器周期的 T2 节拍后已经有效，但是暂存器 A 的数据在 T4 节拍上升沿才改变为 E7H，说明暂存器 A 的输入是时序逻辑，受 T4 节拍控制。

暂存器 B 同理，其数据在相应的 T4 节拍上升沿改变为 B7H。运算方式选择 S3..S0 在第三个机器周期 T2 节拍后被设置为**算术加法**运算，进位标志 FC 在 T4 节拍上升沿改变，说明进位标志 FC 是时序逻辑，受 T4 节拍影响。

由于  $E7H + B7H = 19EH$ ，此运算会产生进位，因此 FC 标志应在此时被置位。运算器的输出在第三个机器周期 T2 节拍结束后 T3 节拍来之前 ALU\_B 变有效后直接输出结果 9EH（因为进位被舍弃），可知运算器的输出是组合逻辑，只受 ALU\_B 影响。

## 七、实验结果与分析

运算类型	A	B	S3	S2	S1	S0	CN	结果		
逻辑运算	E7	B7	0	0	0	0	X	F=(E7)	FC=(0)	FZ=(0)
	E7	B7	0	0	0	1	X	F=(B7)	FC=(0)	FZ=(0)
	E7	B7	0	0	1	0	X	F=(A7)	FC=(0)	FZ=(0)
	E7	B7	0	0	1	1	X	F=(F7)	FC=(0)	FZ=(0)
	E7	B7	0	1	0	0	X	F=(18)	FC=(0)	FZ=(0)
	E7	B7	0	1	0	1	X	F=(73)	FC=(0)	FZ=(0)
移位运算	E7	B7	0	1	1	0	0	F=(73)	FC=(0)	FZ=(0)
	E7	B7	0	1	1	0	1	F=(F3)	FC=(1)	FZ=(0)
	E7	B7	0	1	1	1	0	F=(CE)	FC=(0)	FZ=(0)
	E7	B7	0	1	1	1	1	F=(CF)	FC=(1)	FZ=(0)
算术运算	E7	B7	1	0	0	0	0	F=(E7)	FC=(0)	FZ=(0)
	E7	B7	1	0	0	0	1	F=(E7)	FC=(1)	FZ=(0)
	E7	B7	1	0	0	1	X	F=(9E)	FC=(1)	FZ=(0)
	E7	B7	1	0	1	0 (FC=0)	X	F=(9E)	FC=(1)	FZ=(0)
	E7	B7	1	0	1	0 (FC=1)	X	F=(9F)	FC=(1)	FZ=(0)
	E7	B7	1	0	1	1	X	F=(30)	FC=(1)	FZ=(0)
	E7	B7	1	1	0	0	X	F=(E6)	FC=(0)	FZ=(0)
	E7	B7	1	1	0	1	X	F=(E8)	FC=(1)	FZ=(0)
逻辑运算	00	FF	0	0	1	0	X	F=(00)	FC=(0)	FZ=(1)
算术运算	FF	01	1	0	0	1	X	F=(00)	FC=(1)	FZ=(1)
	80	80	1	0	0	1	X	F=(00)	FC=(1)	FZ=(1)
	7F	01	1	1	0	1	X	F=(80)	FC=(0)	FZ=(0)

表 1: 运算器结果

主要以  $A = E7_H, B = B7_H$  输入完成大部分的运算，另外也取了一些特殊数据来验证 FZ、FC 标志位，记录数据如表 1 所示。

运算分析：

逻辑运算 ( $A=E7_H, B=B7_H$ )

- a)  $S_3S_2S_1S_0 = 0000$ :  $F=E7_H, FC=0, FZ=0$ 
  - 操作: A 的直通
  - 结果: 正确输出 A 的值
  - 标志位: FC 和 FZ 为 0 (无进位, 结果非零)
- b)  $S_3S_2S_1S_0 = 0001$ :  $F=B7_H, FC=0, FZ=0$ 
  - 操作: B 的直通
  - 结果: 正确输出 B 的值
  - 标志位: FC 和 FZ 保持为 0
- c)  $S_3S_2S_1S_0 = 0010$ :  $F=A7_H, FC=0, FZ=0$ 
  - 操作: A 和 B 的按位与
  - 计算:  $E7_H (11100111) \& B7_H (10110111) = A7_H (10100111)$
- d)  $S_3S_2S_1S_0 = 0011$ :  $F=F7_H, FC=0, FZ=0$ 
  - 操作: A 和 B 的按位或
  - 计算:  $E7_H (11100111) | B7_H (10110111) = F7_H (11110111)$
- e)  $S_3S_2S_1S_0 = 0100$ :  $F=18_H, FC=0, FZ=0$ 
  - 操作: A 的按位取反
  - 计算:  $\sim E7_H (11100111) = 18_H (00011000)$
- f)  $S_3S_2S_1S_0 = 0101$ :  $F=73_H, FC=0, FZ=0$ 
  - 操作: A 不带进位循环右移 B (取低 3 位) 位
  - 计算:  $B7_H$  的低 3 位是 111 (二进制), 即 7  $E7_H (11100111)$  右移 7 位 =  $73_H (01110011)$

移位运算 ( $A=E7_H, B=B7_H$ )

- a)  $S_3S_2S_1S_0 = 0110, CN=0$ :  $F=73_H, FC=0, FZ=0$ 
  - 操作: A 的逻辑右移一位
  - 计算:  $E7_H (11100111) \gg 1 = 73_H (01110011)$
- b)  $S_3S_2S_1S_0 = 0110, CN=1$ :  $F=F3_H, FC=1, FZ=0$ 
  - 操作: A 的带进位循环右移一位
  - 计算:  $E7_H (11100111)$  循环右移 1 位, 最低位 1 移到最高位
  - 结果:  $F3_H (11110011)$
  - 标志位:  $FC=1$  (移出的位是 1)
- c)  $S_3S_2S_1S_0 = 0111, CN=0$ :  $F=CE_H, FC=0, FZ=0$ 
  - 操作: A 的逻辑左移一位
  - 计算:  $E7_H (11100111) \ll 1 = CE_H (11001110)$
- d)  $S_3S_2S_1S_0 = 0111, CN=1$ :  $F=CF_H, FC=1, FZ=0$ 
  - 操作: A 的带进位循环左移一位
  - 计算:  $E7_H (11100111)$  循环左移 1 位, 最高位 1 移到最低位
  - 结果:  $CF_H (11001111)$

- 标志位:  $FC=1$  (移出的位是 1)

算术运算 ( $A=E7_H$ ,  $B=B7_H$ )

- $S_3S_2S_1S_0 = 1000$ ,  $CN=0$ :  $F=E7_H$ ,  $FC=0$ ,  $FZ=0$ 
  - 操作: 置  $FC=CN$
  - 结果:  $FC=0$ ,  $F$  保持  $A$  的值不变
- $S_3S_2S_1S_0 = 1000$ ,  $CN=1$ :  $F=E7_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作: 置  $FC=CN$
  - 结果:  $FC=1$ ,  $F$  保持  $A$  的值不变
- $S_3S_2S_1S_0 = 1001$ :  $F=9E_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作:  $A$  加  $B$
  - 计算:  $E7_H + B7_H = 19E_H$
  - 结果:  $F=9E_H$  (结果超过 8 位)
  - 标志位:  $FC=1$  (有进位)
- $S_3S_2S_1S_0 = 1010$ ,  $FC=0$ :  $F=9E_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作:  $A$  加  $B$  加  $FC$  (初始  $FC=0$ )
  - 结果: 同上, 因为  $FC$  初始为 0
- $S_3S_2S_1S_0 = 1010$ ,  $FC=1$ :  $F=9F_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作:  $A$  加  $B$  加  $FC$  (初始  $FC=1$ )
  - 计算:  $E7_H + B7_H + 1 = 19F_H$
  - 结果:  $F=9F_H$
  - 标志位:  $FC=1$  (有进位)
- $S_3S_2S_1S_0 = 1011$ :  $F=30_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作:  $A$  减  $B$
  - 计算:  $E7_H - B7_H = 30_H$
  - 标志位:  $FC=1$  (无借位, 结果为正)
- $S_3S_2S_1S_0 = 1100$ :  $F=E6_H$ ,  $FC=0$ ,  $FZ=0$ 
  - 操作:  $A$  减 1
  - 计算:  $E7_H - 1 = E6_H$
  - 标志位:  $FC=0$  (有借位)
- $S_3S_2S_1S_0 = 1101$ :  $F=E8_H$ ,  $FC=1$ ,  $FZ=0$ 
  - 操作:  $A$  加 1
  - 计算:  $E7_H + 1 = E8_H$
  - 标志位:  $FC=1$  (无进位)

特殊情况

- $A=00_H$ ,  $B=FF_H$ ,  $S_3S_2S_1S_0 = 0010$ :  $F=00_H$ ,  $FC=0$ ,  $FZ=1$ 
  - 操作: 按位与
  - 结果: 0
  - 标志位:  $FZ=1$  (结果为零)
- $A=FF_H$ ,  $B=01_H$ ,  $S_3S_2S_1S_0 = 1001$ :  $F=00_H$ ,  $FC=1$ ,  $FZ=1$ 
  - 操作: 加法
  - 计算:  $FF_H + 01_H = 100_H$
  - 结果: 溢出,  $F=00_H$
  - 标志位:  $FC=1$  (进位),  $FZ=1$  (结果为零)
- $A=80_H$ ,  $B=80_H$ ,  $S_3S_2S_1S_0 = 1001$ :  $F=00_H$ ,  $FC=1$ ,  $FZ=1$

- 操作：加法
  - 计算： $80_H + 80_H = 100_H$
  - 结果：溢出， $F=00_H$
  - 标志位： $FC=1$ （进位）， $FZ=1$ （结果为零）
- d)  $A=7F_H$ ,  $B=01_H$ ,  $S_3S_2S_1S_0 = 1101$ :  $F=80_H$ ,  $FC=0$ ,  $FZ=0$
- 操作：A 加 1
  - 计算： $7F_H + 1 = 80_H$
  - 结果：变为负数（最高位为 1）
  - 标志位： $FC=0$ （无进位）

通过上述数据测试结果验证 ALU 能够正确执行各种操作，包括逻辑运算、移位运算和算术运算。ALU 在处理边界情况（溢出、零结果等）时也表现出正确的行为，证明其设计和实现符合预期，能够准确执行各种基本运算并正确设置相关标志位。

## 八、思考题回答及实验心得

思考题：

将  $A=01H$  和  $B=02H$  进行逻辑与运算，观察运算器零标志  $FZ$  的时序，是否和进位标志  $FC$  一致？

回答：

零标志  $FZ$  的时序与进位标志  $FC$  的时序不一致。分析如下：

### 1. 运算过程

- $A = 01H = 00000001$  (B)
- $B = 02H = 00000010$  (B)
- $A \& B = 00000000 = 00H$

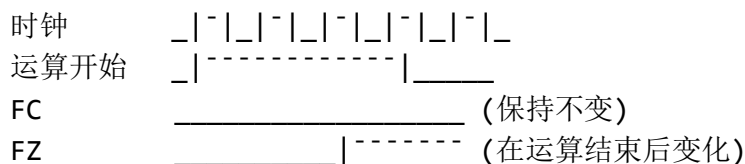
### 2. 结果

- 运算结果为  $00H$ ，即全 0。
- 逻辑与运算不涉及进位， $FC$  保持不变（0）。
- 结果为 0，因此  $FZ$  会被置为 1。

### 3. 时序行为

- a)  $FC$ （进位标志）：
  - 在逻辑运算中不受影响。
  - 整个运算过程中保持稳定状态。
- b)  $FZ$ （零标志）：
  - 运算完成后根据结果更新。
  - 从初始状态（0）变为 1。

### 4. 时序图示意



所以在这个逻辑与运算中，FC 和 FZ 的时序行为不一致。FC 在整个过程中保持不变，而 FZ 会在运算后更新为 1。FC 主要用于算术运算，在逻辑运算中通常不变。FZ 需要在每次运算后根据结果进行更新，因此会有一个明显的状态变化。

心得体会:

通过完成这次基本运算器实验，我深入理解了计算机底层运算的基本原理和实现方式。还有很多的心得体会。

首先是我真正体会到了课本上学到的知识在实际中的应用。比如做  $E7_H + B7_H$  的加法运算时, 我看到了如何通过控制信号  $S3 \sim S0 = 1001$  来触发加法操作, 并观察到结果  $9E_H$  以及进位标志  $FC$  的变化。这个过程使抽象的加法器概念变得具体和生动。

其次是通过观察 FC（进位标志）和 FZ（零标志）的变化，我理解了这两个标志位的重要作用。做  $FF_H + 01_H = 00_H$  时，FC 被置为 1，指示发生了溢出。所以 FC 在多精度运算中实际上是很重要的。做  $00_H \& FF_H = 00_H$  时，FZ 被置为 1，所以 FZ 用来做条件判断，检测运算结果是否为零。

然后还有 ALU 基本运算单元的强大和灵活性，不管是算术运算、逻辑运算还是移位运算都能轻松完成。

最后我尝试了在实验中处理特殊情况，处理特殊情况时实际上就要考虑边界情况。比如做  $80_{\text{H}} + 80_{\text{H}} = 00_{\text{H}}$  运算，此时溢出会使标志位 FC 置 1，发生溢出。做  $\text{FF}_{\text{H}} + 01_{\text{H}} = 00_{\text{H}}$  的运算会产生零结果，导致 FZ 置 1。做  $7\text{F}_{\text{H}} + 01_{\text{H}} = 80_{\text{H}}$  运算会使正数变为负数，此时符号位改变。