

第 1 章 运算器

计算机的一个最主要的功能就是处理各种算术和逻辑运算，这个功能要由 CPU 中的运算器来完成，运算器也称作算术逻辑部件 ALU。本章首先安排一个基本的运算器实验，了解运算器的基本结构，然后再设计一个加法器和一个乘法器。

1.1 基本运算器实验

1.1.1 实验目的

- (1) 了解运算器的组成结构。
- (2) 掌握运算器的工作原理。

1.1.2 实验设备

PC 机一台，TD-CMA 实验系统一套。

1.1.3 实验原理

本实验的原理如图 1-1-1 所示。

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（有些处理器体系结构把移位运算器放于算术和逻辑运算部件之前，如 ARM），各部件对操作数进行何种运算由控制信号 S3...S0 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。

逻辑运算部件由逻辑门构成，较为简单，而后面又有专门的算术运算部件设计实验，在此对这两个部件不再赘述。移位运算采用的是桶形移位器，一般采用交叉开关矩阵来实现，交叉开关的原理如图 1-1-2 所示。图中显示的是一个 4X4 的矩阵（系统中是一个 8X8 的矩阵）。每一个输入都通过开关与一个输出相连，把沿对角线的开关导通，就可实现移位功能，即：

(1) 对于逻辑左移或逻辑右移功能，将一条对角线的开关导通，这将所有的输入位与所使用的输出分别相连，而没有同任何输入相连的则输出连接 0。

(2) 对于循环右移功能，右移对角线同互补的左移对角线一起激活。例如，在 4 位矩阵中使用‘右 1’和‘左 3’对角线来实现右循环 1 位。

(3) 对于未连接的输出位，移位时使用符号扩展或是 0 填充，具体由相应的指令控制。使用另外的逻辑进行移位总量译码和符号判别。

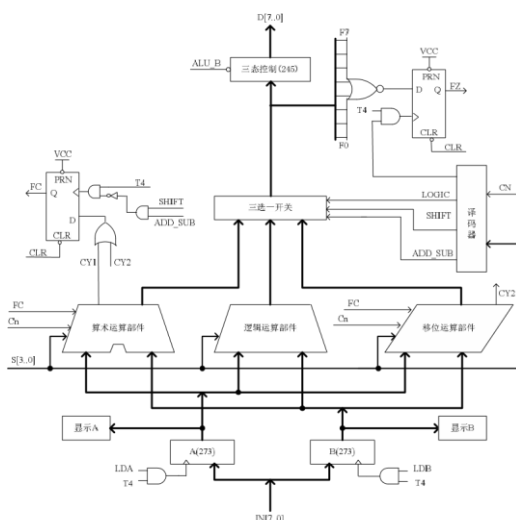


图 1-1-1 运算器原理图

运算器部件由一片 CPLD 实现。ALU 的输入和输出通过三态门 74LS245 连到 CPU 内总线上，另外还有指示灯标明进位标志 FC 和零标志 FZ。请注意：实验箱上凡丝印标注有马蹄形标记‘ \sqcap ’，表示这两根排针之间是连通的。图中除 T4 和 CLR，其余信号均来自于 ALU 单元的排线座，实验箱中所有单元的 T1、T2、T3、T4 都连接至控制总线单元的 T1、T2、T3、T4，CLR 都连接至 CON 单元的 CLR 按钮。T4 由时序单元的 TS4 提供（时序单元的介绍见附录二），其余控制信号均由 CON 单元的二进制数据开关模拟给出。控制信号中除 T4 为脉冲信号外，其余均为电平信号，其中 ALU_B 为低有效，其余为高有效。

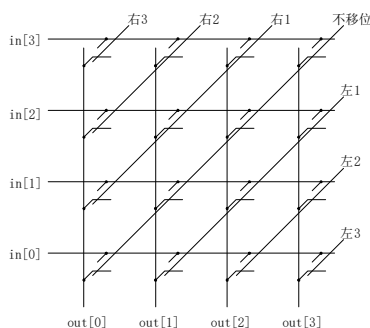


图 1-1-2 交叉开关桶形移位器原理图

暂存器 A 和暂存器 B 的数据能在 LED 灯上实时显示，原理如图 1-1-3 所示（以 A0 为例，其它相同）。进位标志 FC、零标志 FZ 和数据总线 D7...D0 的显示原理也是如此。



1-1-3 A0 显示原理图

ALU 和外围电路的连接如图 1-1-4 所示，图中的小方框代表排针座。

运算器的逻辑功能表如表 1-1-1 所示，其中 S3 S2 S1 S0 CN 为控制信号，FC 为进位标志，FZ 为运算器零标志，表中功能栏内的 FC、FZ 表示当前运算会影响到该标志。

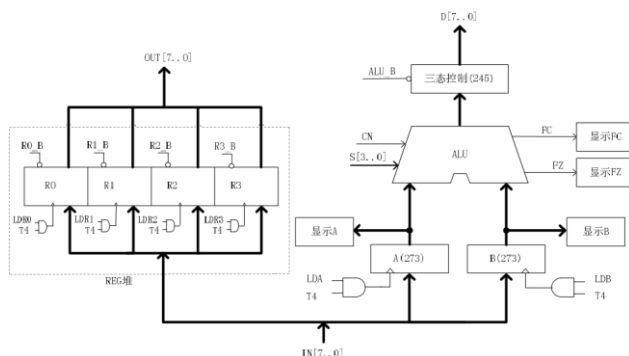


图 1-1-4 ALU 和外围电路连接原理图

表 1-1-1 运算器逻辑功能表

运算类型	S3 S2 S1 S0	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=/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)	

*表中“X”为任意态，下同

1.1.4 实验步骤

(1) 按图 1-1-5 连接实验电路，并检查无误。图中将用户需要连接的信号用圆圈标明（其它实验相同）。

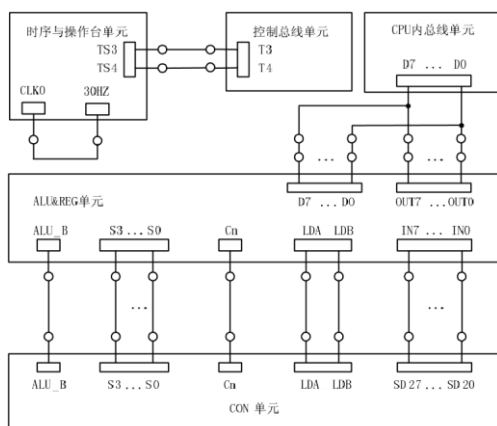


图 1-1-5 实验接线图

(2) 将时序与操作台单元的开关 KK2 置为‘单步’档,开关 KK1、KK3 置为‘运行’档。(3) 打开电源开关，如果听到有‘嘀’报警声，说明有总线竞争现象，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) 用输入开关向暂存器 A 置数。

① 拨动 CON 单元的 SD27...SD20 数据开关，形成二进制数 01100101（或其它数值），数据显示亮为‘1’，灭为‘0’。

② 置 LDA=1，LDB=0，连续按动时序单元的 ST 按钮，产生一个 T4 上沿，则将二进制数 01100101 置入暂存器 A 中，暂存器 A 的值通过 ALU 单元的 A7...A0 八位 LED 灯显示。

(5) 用输入开关向暂存器 B 置数。

① 拨动 CON 单元的 SD27...SD20 数据开关，形成二进制数 10100111（或其它数值）。

- ② 置 LDA=0, LDB=1, 连续按动时序单元的 ST 按钮, 产生一个 T4 上沿, 则将二进制数

10100111

置入暂存器 B 中, 暂存器 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。置 ALU_B=0、LDA=0、LDB=0, 然后按表 1-1-1 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。如置 S3、S2、S1、

S0 为 0010, 运算器作逻辑与运算, 置 S3、S2、S1、S0 为 1001, 运算器作加法运算。

如果实验箱和 PC 联机操作, 则可通过软件中的数据通路图来观测实验结果 (软件使用说明请看附录一), 方法是: 打开软件, 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 1-1-6 所示。进行上面的手动操作, 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】—【单周期】”, 其作用相当于将时序单元的状态开关 KK2 置为‘单步’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, 并完成表 1-1-2。然后改变 A、B 的值, 验证 FC、FZ 的锁存功能。

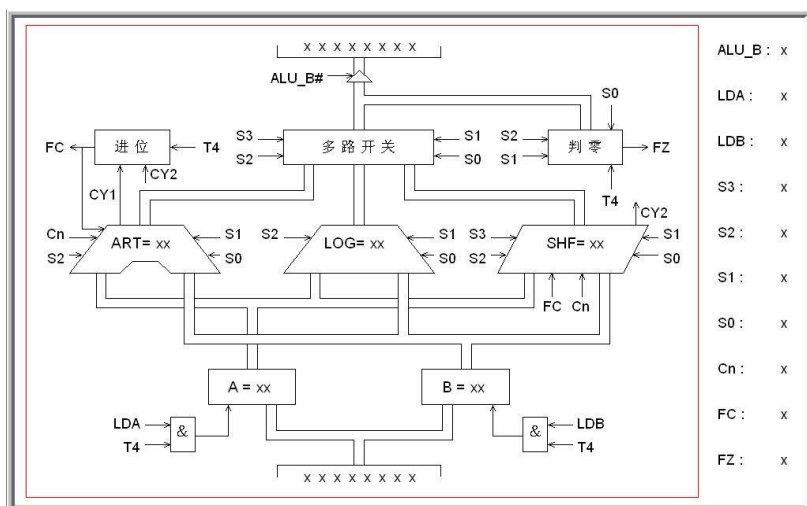


图 1-1-6 数据通路图表

1-1-2 运算结果表

运算类型	A	B	S3 S2 S1 S0	CN	结果
逻辑运算	65	A7	0 0 0 0	X	F=(65) FC=() FZ=()
	65	A7	0 0 0 1	X	F=(A7) FC=() FZ=()
			0 0 1 0	X	F=() FC=() FZ=()
			0 0 1 1	X	F=() FC=() FZ=()
			0 1 0 0	X	F=() FC=() FZ=()

移位运算			0 1 0 1	X	F=() FC=() FZ=()
			0 1 1 0	0	F=() FC=() FZ=()
				1	F=() FC=() FZ=()
			0 1 1 1	0	F=() FC=() FZ=()
				1	F=() FC=() FZ=()
算术运算			1 0 0 0	X	F=() FC=() FZ=()
			1 0 0 1	X	F=() FC=() FZ=()
			1 0 1 0 (FC=0)	X	F=() FC=() FZ=()
			1 0 1 0 (FC=1)	X	F=() FC=() FZ=()
			1 0 1 1	X	F=() FC=() FZ=()
			1 1 0 0	X	F=() FC=() FZ=()
			1 1 0 1	X	F=() FC=() FZ=()

1.2 超前进位加法器设计实验

1.2.1 实验目的

- (1) 掌握超前进位加法器的原理及其设计方法。
- (2) 熟悉 CPLD 应用设计及 EDA 软件的使用。

1.2.2 实验设备

PC 机一台，TD-CMA 实验系统一套。

1.2.3 实验原理

加法器是执行二进制加法运算的逻辑部件，也是 CPU 运算器的基本逻辑部件（减法可以通过补码相加来实现）。加法器又分为半加器和全加器（FA），不考虑低位的进位，只考虑两个二进制数相加，得到和以及向高位进位的加法器为半加器，而全加器是在半加器的基础上又考虑了低位过来的进位信号。

表 1-2-1 1 位全加器真值表

输 入			输 出	
A	B	Ci	S	Co