

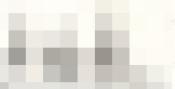
北京邮电大学

实验报告

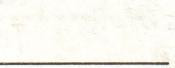
课程名称 计算机组成原理

实验名称 运算器组成实验

计算机系301班

姓名 

教师 

成绩 

2024年3月28日

一、实验类型

本实验类型为原理型十分析型。

二、实验目的

- (1) 熟悉逻辑测试笔的使用方法；
- (2) 熟悉TEC-8模型计算机的节拍脉冲 T_1 、 T_2 、 T_3 ；
- (3) 熟悉双端口寄存器组的读写操作；
- (4) 熟悉运算器的数据传递通路；
- (5) 验证74181的加、减、与或功能；
- (6) 完成几种指定的算术、逻辑运算。

三、实验设备

TEC-8A超强型计算机组成原理与系统结构实验室、逻辑笔一支、导线若干。

四、实验原理

1. TEC-8模型计算机的基本时序

在TEC-8中，执行一条微指令（或在硬布线控制器中完成一个机器周期）需要连续的3个节拍脉冲 T_1 、 T_2 和 T_3 。它们的时序关系如图1所示。

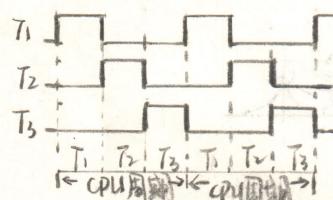


图1. 机器周期与 T_1 、 T_2 、 T_3 时序关系图

对于运算器操作， T_1 期间产生2个8位参与运算的数A和B，A是被加数，B是加数；产生控制运算类型的信号M、 S_3 、 S_2 、 S_1 、 S_0 和 C_{in} ；产生控制写入标志寄存器的信号 LD_Z 和控制写入C标志寄存器的信号 LD_C ，产生将运算器的数据结果送至数据总线DBUS的控制信号ABUS。这些控制信号保持到 T_3 结束； T_2 期间，根据控制信号，完成某种运算功能； T_3 上升沿，保存运算的数据到一个8位寄存器中，同时保存进位标志C和结果为0标志Z。

2. 运算器组成实验电路原理

电路原理图见图2。

双端口寄存器组由1片EPROM7064(U40)组成（见图2中虚线框内），内含4个8位寄存器 R_0 、 R_1 、 R_2 、 R_3 ，2个4选1译码器根据 RD_{0-1} 、 RS_{0-1} 决定A、B两端口输出，2-4译码器对 RD_{0-1} 译码，决定被写入的寄存器，在 T_3 上升沿，数据将被写入。

ALU由2片74181、1片7474、1片74244、1片74245和1片7430构成。加法和减法运算同时影响

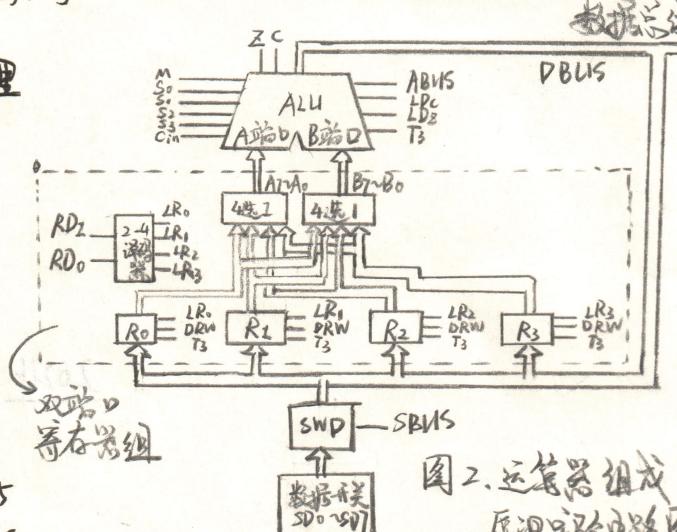


图2. 运算器组成
原理实验电路图

(按本人实际接线情况设计)

C标志和Z标志，与操作码操作只影响Z标志。

注意：数据总线 DBUS 有 4 个信号来源，运算器、存储器、数据开关和中断地址寄存器，在任一时刻只允许其中一个信号源送数据总线。

五、实验步骤

1. 打开实验箱，确定设备已断电后，将控制信号切换器的开关拨到中间位置（“独立”），将编程开关拨到“正常”位置。

2. 按照表 1. 接线参照表接线，接通电源。表 1. 接线参照表

3. 合上电源，按下 CLR 按钮，使得

~~TEC-8 实验系统~~ 将 K_4, K_{12}, K_{13} 置为低电平，合上电源，按下 CLR 按钮，使得 TEC-8 实验系统处于初始状态。

4. 将数据传入寄存器。

① ~~将 K₄, K₆ 调为 L~~，将 K_4 调为 H，将 K_{13} 调为 H，

② 将 K_4, K_6 调为 LL，在数据开关 SD[0~SD7] 处拨出 $SD[7:0] = FF_{16}$ ，按下 DP，将 FF_{16} 传入寄存器 R_0 。

③ 将 K_4, K_6 调为 LH，在数据开关处调出 $SD[7:0] = 55_{16}$ ，按下 DP，将 55_{16} 传入寄存器 R_1 。

④ 将 K_4, K_6 调为 HL，在数据开关处调出 $SD[7:0] = AA_{16}$ ，按下 DP，将 AA_{16} 传入寄存器 R_2 。

⑤ 将 K_4, K_6 调为 HH，在数据开关处调出 $SD[7:0] = 10_{16}$ ，按下 DP，将 10_{16} 传入寄存器 R_3 。

⑥ 将 K_4 调为 L，将 K_{13} 调为 L，防止后续实验中寄存器的值被修改。

5. 验证 ALU 的运算功能。将 K_{12} 调为 H，即可在 DBUS 中观察到运算结果，将 K_5, K_6 拨为 H。
 (加) ① 将 $K[10:7]$ 拨到 HLLH，将 K_{11} 拨为 L，将 K_4 拨为 H，依次拨动 $K[3:2]$ 和 $K[1:0]$ ，每次拨动后按下 DP，记录运算结果和 C, Z。

(减) ② 将 $K[10:7]$ 拨到 LHHL，将 K_{11} 拨为 L，将 K_4 拨为 L，依次拨动 $K[3:2]$ 和 $K[1:0]$ ，每次拨动后按下 DP，记录运算结果和 C, Z。

(与) ③ 将 $K[10:7]$ 拨到 HLHH，将 K_{11} 拨为 H，~~将 K4 拨为 H~~ 依次拨动 $K[3:2]$ 和 $K[1:0]$ ，每次拨动后按下 DP，记录运算结果和 C, Z。

(或) ④ 将 $K[10:7]$ 拨到 HHHL，将 K_{11} 拨为 H，依次拨动 $K[3:2]$ 和 $K[1:0]$ ，每次拨动后按下 DP，记录运算结果和 C, Z。

6. 断电，整理实验仪器，实验结束。

电平/开关	输入信号	接线颜色
K_0	RD_0	黄
K_1	RD_1	
K_2	RS_0	绿
K_3	RS_1	
K_4	DRW	蓝
K_5	LD_C	黄
K_6	LD_Z	
K_7	S_0	
K_8	S_1	蓝
K_9	S_2	
K_{10}	S_3	
K_{11}	M	红
K_{12}	A_{BUS}	黄
K_{13}	S_{BUS}	绿
K_{14}	C_{IN}	黄
GND	M_{BUS}	黑

六、实验数据整理。

见表2- 加、与、或表与表3- 减表。

表2. 加、与、或表。

A	B	加法			与			或		
		结果	C	Z	结果	C	Z	结果	C	Z
FF	FF	FE	1	0	FF	1	0	FF	1	0
FF	55	54	1	0	55	1	0	FF	1	0
FF	AA	A9	1	0	AA	1	0	FF	1	0
FF	10	0F	1	0	10	1	0	FF	1	0
55	55	AA	0	0	55	1	0	55	1	0
55	AA	FF	0	0	00	1	1	FF	0	0
55	10	65	0	0	10	1	0	55	1	0
AA	AA	54	1	0	AA	1	0	AA	1	0
AA	10	BA	0	0	00	1	1	BA	1	0
10	10	20	0	0	10	1	0	10	1	0

← C标志很特殊?

表3. 减表

七、实验结果分析

经分析，实验结果与理论知识相符，可以很简单地验证加法、减法的结果与C、Z标志，同时与、或运算结果与Z标志的正确性也容易验证，C标志位的解析见八、实验思考部分。第六个思考题。

A	B	减法		
		结果	C	Z
FF	FF	00	1	1
FF	55	AA	1	0
FF	AA	55	1	0
FF	10	EF	1	0
55	FF	56	0	0
55	55	00	1	1
55	AA	AB	0	0
55	10	45	1	0
AA	FF	AB	0	0
AA	55	55	1	0
AA	AA	00	1	1
AA	10	9A	1	0
10	FF	11	0	0
10	55	BB	0	0
10	AA	66	0	0
10	10	00	1	1

八、实验思考

1. 计算机为何能够快速地进行运算？程序为何执行效率不一样？

计算机之所以能快速运算，是因为电子元件的高速开关使得其具备了高速的信息传输和处理能力。

程序执行效率则受多个层次差异的影响，例如：系统调度效率，数据结构与算法的选择，编程语言和编译器的差异，CPU的性能指标等。

2. 本次实验时 MBUS 为何接地？

在 TEC-8 实验模型中，MBUS 是双端口 RAM 的控制信号，当 MBUS 为高电平时，双端口 RAM 将左端口的值传递到数据总线上，这将干扰本次实验的操作，故应将 MBUS 接地。

3. 结合实验现象，解释各信号作用，说明每一实验步骤中，有哪些信号是（或不是）必需的。

各信号作用见实验原理部分，实验步骤为本人构造，所列出的需要操作的电平开关为必需的，否则为不必要的。

4. ALU 具有记忆功能吗？如果有，如何设计？

ALU 为组合逻辑电路，不具有记忆功能，可见图 3.74181 原理图。

5. 为什么 ALU 的端口确定后，DBUS 上能直接观测到运算结果，而标志位却在下一步才能观测到？

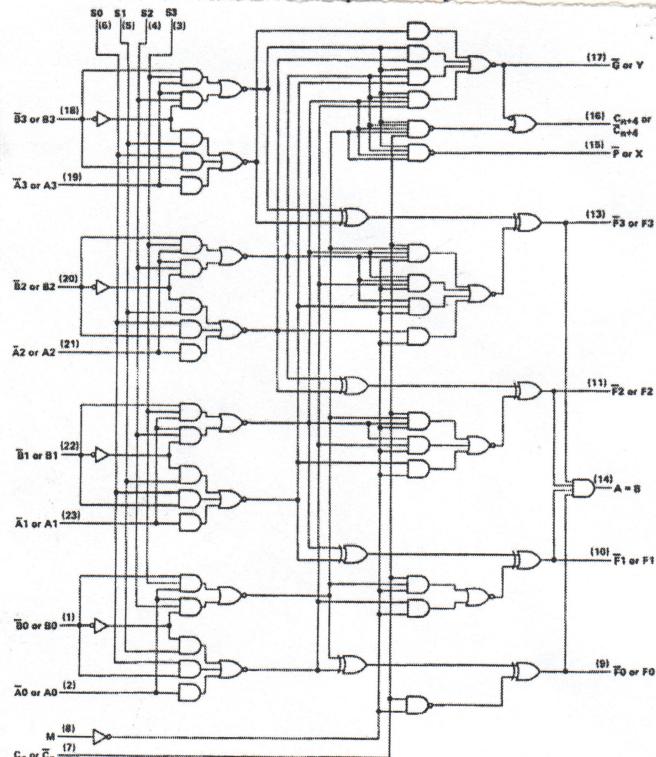
DBUS 上能观测到结果，是因为 ABUS 为高电平后，运算结果将传递到 DBUS 上。

标志位在下一步才能观测到，是因为 C、Z 有储存在触发器中，需在 T₃ 上升沿才能更新其存储的值。

6. 逻辑运算中 C 标志位怎么产生的？

74181 是函数发生器式运算器，即输入 A、B 经函数发生器转化为 X、Y 后传入受控全加器得到结果，M 为全加器控制信号，M=1 时将阻断进位信号，对应为逻辑运算，但 M 并未阻断 C_{n+1} 向集成片的传递，故 C 标志位仍会产生。

图 3.74181 原理图



九、实验总结

在本次实验中，我使用 TEC-8 计算模型，对运算器的逻辑运算功能和算术运算功能进行了验证。在实验后，我对实验数据进行了整理，并进行了深入的分析和思考。

经过本次实验，我了解了 TEC-8 模型计算机的基本时序，熟悉了双端口寄存器组的读写操作，熟悉了运算器的组成和操作，熟悉了运算器的数据传递。可以说，我较好地达成了本次实验的目的。

