

计算机组织与结构实验指导

2024

第一章 TEC-9 计算机组成和数字逻辑实验系统介绍

TEC-9 它适用于《计算机组成原理》、《计算机组织和结构》和《数字逻辑和数字系统》三门课程的实验教学，是一种多用仪器。该仪器将提高学生的动手能力，提高学生对计算机整体和各组成部分的理解，提高学生对数字系统和计算机系统的综合设计能力。

安全提示：

请仔细阅读安全提示部分内容

- 1、CPLD（EPM3128）在做数字电路实验或非组成逻辑控制器实验时，特别在芯片中有其它内容时，请将选择开关拨到“脱机”或“微程序”位置。以避免 CPLD 输出的信号与数据通路和时序电路的部份信号冲突。使 CPLD 发热以至烧坏该芯片或烧坏数据通路部分器件（数据通路部分的总线允许信号在同一时间只允许一个信号有效）。
- 2、数据通路实验时，请将开关拨到“脱机”位置，数据通路信号不能同时有两种输入方式，否则易损坏其它芯片。
- 3、为了防止 CPLD 被损坏，在小板上设有隔离芯片，因此请将信号管脚设定为实验台上标有的方式（输入或者输出），输入管脚禁止设为输出管脚，输出管脚禁止设为输入管脚。否则 CPLD 芯片会发烫乃至损坏芯片。
- 4、每次数字电路/EDA 实验后或用户自定义控制实验后，为了确保实验台安全，可将其内容恢复成出厂时的硬布线控制器内容（光盘中的硬布线控制器写入内容）。

一、TEC-9 实验系统的特点

1. 采用单板式结构、计算机模型采用 8 位。简单、实用。计算机模型分为数据通路、控制器、时序电路、控制台、数字逻辑实验区五部分。各部分之间采用可插、拔的导线连接或进行开关直接选通。
2. 指令系统采用 4 位操作码，容纳 16 条指令，已实现了加、减、逻辑与、存数、取数、条件转移、无条件转移、开中断、关中断、中断返回、停机 11 条指令，指令功能非常典型。其它指令备用。
3. 数据通路采用双端口存储器作为主存，实现了数据总线和指令总线双总线体制，体现了当代 CPU 的设计思想。
4. 运算器中 ALU 由 2 片 74181 实现。4 个通用寄存器由 1 片 ispLSI1016 组成，设计新颖。
5. 控制器采用微过程控制器和硬连线控制器两种类型，体现了当代计算机控制器设计技术的完备性。
6. 控制存储器中的微代码可以通过 PC 机下载或由实验台 PS2 键盘修改，省去插、拔 EEPROM 芯片。
7. 实验台上提供了一片在系统编程器件 CPLD,学生在 PC 机设计好组合逻辑控制器方案后下载到 CPLD 中，CPLD 就构成了新的控制器。控制器的设计并实现对提高计算机综合设计能力会有很大帮助。CPLD 也可用于数字逻辑和数字系统的设计。
8. 控制台包含 8 个数据开关，用于置数功能；16 个逻辑电平开关，用于置信号电平；控制台有复位和启动二个单脉冲发生器，一个单独的单脉冲，用于数字电路实验和中断。控制台有 5 种操作：写存储器，读存储器，读寄存器，写寄存器，启动程序运行。
9. 微程序控制器中的微代码输出、微地址总线、程序地址总线、数据总线、存储器地址总线、进位、双端口存储器的读、写冲突位 BUSYL#和 BUSYR#等都有指示灯，便于查看指令的执行过程。
10. 数字逻辑和数字系统实验部分除上述可用的一片 CPLD、16 个电平开关和 3 个单脉冲按钮（复位和启动）外，还有 8 个指示灯，多个双列直插插座，4 个 8432 编码驱动的数码管，4 个直接驱动的数码管，1 个喇叭。

11. 实验台采用监控电路，监控电路采用了 128X64 字符图形液晶，可以方便查看和修改寄存器、存储器、微程序控制器中的内容。

12. 实验台采用了一键恢复功能，在微程序控制器代码被破坏时，通过一键恢复即可恢复到出厂状态。在恢复时需输入修改密码。可以防止被恶意恢复。

13. 实验台配套软件可以方便查看、修改寄存器、双端口存储器、微程序控制器的内容。可以通过上端运行功能，非常方便用户查看实验运行状况。

14. 实验台控制器有三种模式：

1) 微程序：控制器为微程序控制器，微程序控制器的信号通过隔离芯片直接与数据通路信号相连。避免了综合实验需接很多导线的问题。

2) 脱机：微程序控制器、数据通路、硬布线控制器三部分信号完全独立。由用户通过导线连接。

3) 硬布线：控制器为硬布线控制，硬布线控制器的信号通过隔离芯片直接与数据通路信号相连。避免了综合实验需接很多导线的问题。

15. 电源部分具有抗电源对地短路能力。

16. 微程序/硬布线控制器与数据通路直接选通信号（通过控制开关选择微程序/脱机/硬布线，直接选通信号，不需连接实验导线）：

微程序/硬布线	TJ	CN#	M	S3	S2	S1	S0	M1	LDDR1	WRD	LRW
数据通路	TJ	CN#	M	S3	S2	S1	S0	M1	LDDR1	WRD	LRW

微程序/硬布线	CEL	ALU_BUS	RS_BUS	SW_BUS	IAR_BUS	LDER	M3
数据通路	CEL	ALU_BUS	RS_BUS	SW_BUS	IAR_BUS	LDER	M3

微程序/硬布线	AR1_INC	LDAR1	LDIAR	M4	PC_INC	PD_ADD	LDPC
数据通路	AR1_INC	LDAR1	LDIAR	M4	PC_INC	PD_ADD	LDPC

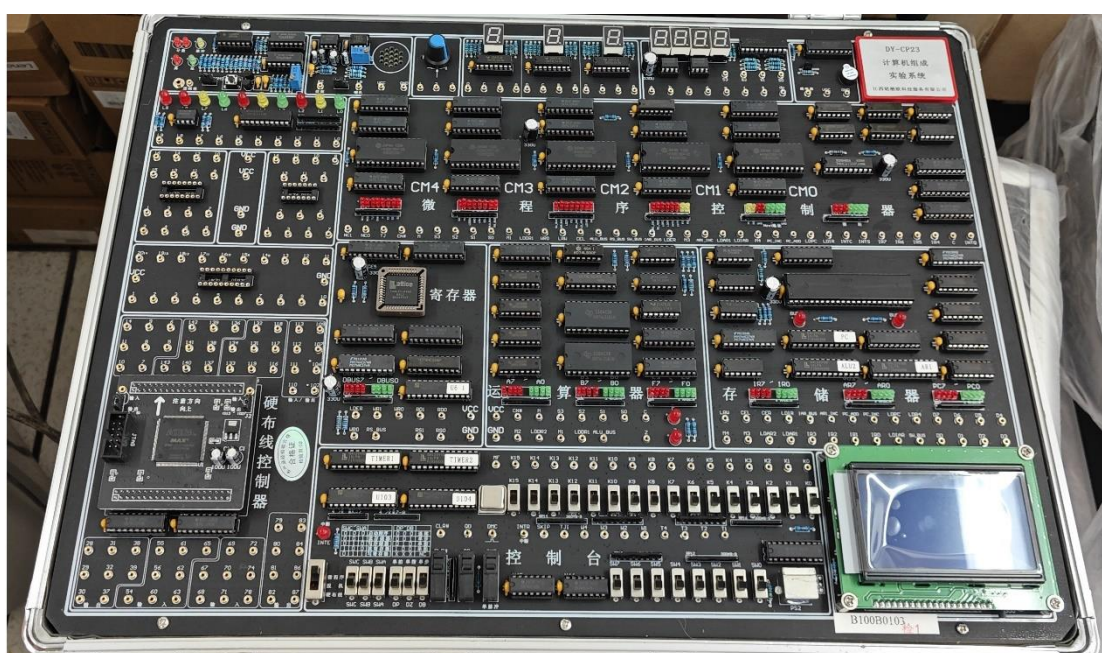
微程序/硬布线	INTC	INTS	IR7	IR6	IR5	IR4	C	INTQ
数据通路	INTC	INTS	IR7	IR6	IR5	IR4	C	INTQ

二、TEC-9 实验系统的组成

TEC-9 实验系统由以下几个部分组成：

- 1) 控制台
- 2) 数据通路
- 3) 控制器
- 4) 时序电路
- 5) 数字逻辑实验区
- 6) 电源模块

下面分别对各组成部分予以介绍。



1、电源

电源部分由一个电源、一个电源插座、一个电源开关和一个红色电源指示灯组成。电源通过四个螺钉安装在实验箱底部，它输出+5V 电压，最大负载电流 3A，具有抗+5V 对地短路功能。电源插座用于接交流 220V，插座内装有保险丝。电源开关接通时，模块电源输出 +5V，红色指示灯点亮。

2、时序发生器

时序发生器产生计算机模型所需的时序和数字逻辑实验所需的时钟。时序电路由一个 500KHz 晶振、2 片 GAL22V10 组成。根据本机设计，执行一条微指令需要 4 个节拍脉冲 T1、T2、T3、T4，执行一条机器指令需要三个节拍电位 W1、W2、W3，因此本机的基本时序如下：

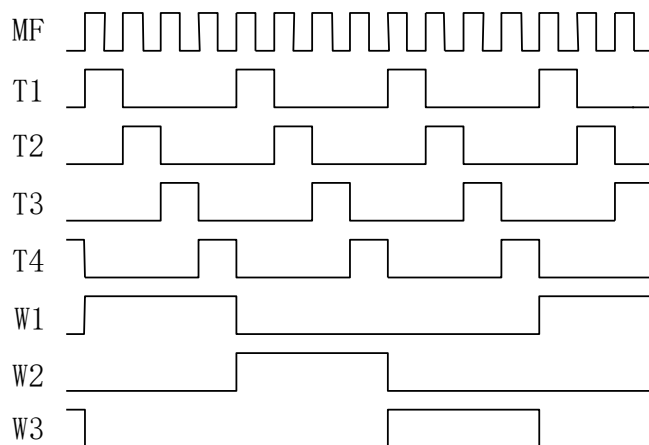


图1.1 基本时序图

图中，MF 是晶振产生的 500KHz 基本时钟，T1、T2、T3、T4 是数据通路和控制器中各寄存器的节拍脉冲信号，印制板上已将它们和相关的寄存器相连。T1、T2、T3、T4 既供微过程控制器使用，也供硬连线控制器使用。W1、W2、W3，只供硬连线控制器做节拍电位信号使用。

3、数据通路

TEC-9 的数据通路采用了数据总线和指令总线双总线形式。它还使用了大规模在系统编程器件作为寄存器堆，使得设计简单明了，可修改性强。

图 1.2 是数据通路总体图，下面介绍图中各个主要部件的作用。

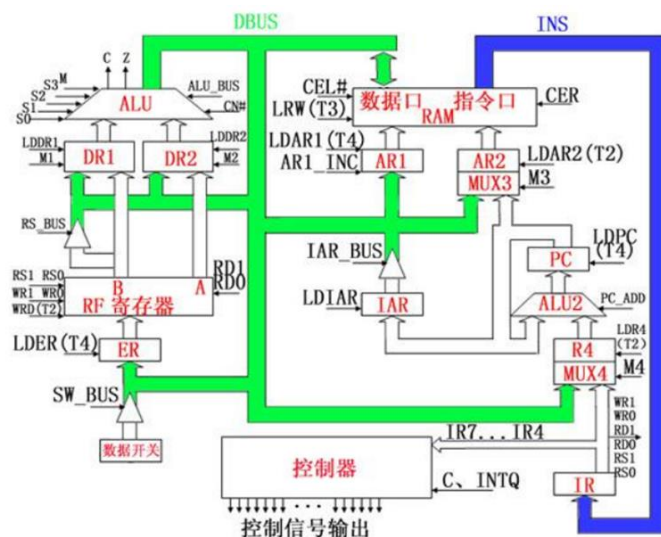


图 1.2 总体数据通路图

1) 运算器 ALU

运算器 ALU 由两片 74LS181 组成，在选择端 M 和 S0-S3 控制下，ALU 对数据 A、B 进行各种算术、逻辑运算。有关 74181 运算的具体操作，请看 74181 的资料和教科书。当 LDRi=1 时，在 T4 的上升沿寄存器 C 保存运算产生的进位标志信号。

2) 运算操作数寄存器 DR1 和 DR2

DR1 和 DR2 是运算操作数寄存器，DR1 和 ALU 的 B 口相连，DR2 和 ALU 的 A 口相连。DR1 和 DR2 各由 2 片 74LS298 构成。当 M1=0 且 LDDR1=1 时，在 T3 的下降沿，DR1 接收来寄存器堆 B 端口的数据，当 M1=1 且 LDDR1=1 时，在 T3 的下降沿，DR1 接收来自数据总线 DBUS 的数据。当 M2=0 且 LDDR2=1 时，在 T3 的下降沿，DR2 接收来自通用寄存器堆 A 端口的数据。当 M2=1 且 LDDR2=1 时，在 T3 的下降沿，DR2 接收来自数据总线 DBUS 的数据。

3) 多端口通用寄存器堆 RF

双端口通用寄存器堆 RF 由一片 ispLSI1016 构成，其中包含 4 个 8 位寄存器 (R0、R1、R2、R3)，有三个控制端口：两个控制读操作，一个控制写操作，三个端口可以同时操作。由 RD1、RD0 选中的寄存器的数据从 A 端口读出，由 RS1、RS0 选中的寄存器的数据从 B 端口读出；WR1、WR0 选择要写入的寄存器。WRD 控制写操作，当 WRD=1 时，在 T2 上升沿将来自 ER 寄存器的数据写入由 WR1、WR0 选中的寄存器。

从 RF 的 A 端口读出的数据直接送 DR2。由 B 端口读出的数据直接送 DR1 之外，还可以送数据总线 DBUS。当 RS_BUS=1 时，允许 B 端口数据送 DBUS。

4) 暂存寄存器 ER

暂存寄存器 ER 是一片 74LS374，主要用于暂时保存运算器的结果。当 LDER=1 时，在 T4 的上升沿，将数据总线 DBUS 上的数据打入暂存寄存器 ER。ER 的输出送往多端口通用寄存器堆 RF，作为写入数据使用。

5) 开关寄存器 SW_BUS

开关寄存器 SW_BUS 是一片 74LS244，用于将控制台数据开关 SW7~SW0 的数据送往数据总线 DBUS。当 SW_BUS=1 时，允许开关 SW7~SW0 的数据送往总线 DBUS。

6) 双端口存储器 RAM

双端口存储器 RAM 由一片 IDT7132 及少量控制电路构成。IDT7132 是 2048 字节的双端口静态随机存储器，本实验系统实际使用 256 字节。IDT7132 的两个端口可以同时进行读、写操作，在本实验系统中，RAM 左端口连接数据总线 DBUS，可进行读、写操作；右端口连接指令总线 IBUS，输出到指令寄存器 IR，作为只读端口使用。IDT7132 有 6 个控制引脚，CEL、LRW、OEL 控制左端口读、写操作；CER、RRW、OER 控制右端口的读写操作。CEL 为左端口选择引脚，高电平有效；当 CEL=0 时，禁止对左端口的读、写操作。LRW 控制对左端口的读写。当 CEL=1 且 LRW=1 时，左端口进行读操作；当 CEL=1 且 LRW=0 在 T3 的上升沿左端口进行写操作。OEL 的作用等同于三态门，当 CEL=1 且 OEL=1 时，允许左端口读出的数据送到数据总线 DBUS 上；当 OEL=1 时，禁止左端口的数据放到 DBUS。本实验系统中左端口 OEL 由 LRW 经反相产生，不需单独控制。控制右端口的三个引脚与左端口的三个完全类似，不过只使用了读操作，在实验板上已将 RRW 固定接高电平，OER 固定接地。当 CER=1 时，右端口读出的数据（更确切的说法是指令）放到指令总线 IBUS 上，然后当 LDIR=1 时在 T3 的上升沿打入指令寄存器 IR。所有数据/指令的写入都使用左端口，右端口作为指令端口，不需要进行数据的写入。

左端口读出的数据放在数据总线 DBUS 上, 由数据总线指示灯 DBUS7-DBUS0 显示。右端口读出的指令放在指令总线 IBUS 上, 由指令总线指示灯 IBUS7-IBUS0 显示。

7) 地址寄存器 AR1 和 AR2

地址寄存器 AR1 和 AR2 提供双端口存储器的地址。AR1 是 1 片 GAL22V10, 具有加 1 功能, 提供双端口存储器左端口地址, AR1 从数据总线 DBUS 接收数据。AR1 的控制信号是 LDAR1 和 AR1_INC。当 AR_INC=1 时, 在 T4 的上升沿, AR1 的值加 1; 当 LDAR1=1 时, 在 T4 的上升沿, 将数据总线 DBUS 的数据打入地址寄存器 AR1。AR2 由 2 片 74LS298 组成, 有两个数据输入端, 一个来自程序计数器 PC, 另一个来自数据总线 DBUS。AR2 的控制信号是 LDAR2 和 M3。M3 选择数据来源, 当 M3=1 时, 选中数据总线 DBUS; 当 M3=0 时, 选中程序计数器 PC。LDAR2 控制何时接收地址, 当 LDAR2=1 时, 在 T2 的下降沿将选中的数据源上数据打入 AR2。

8) 程序计数器 PC、地址加法器 ALU2、地址缓存器 R4

程序计数器 PC、地址加法器 ALU2、地址缓存器 R4 联合完成三种操作: PC 加载 PC+1、PC+D。

R4 是由 2 片 74LS298 构成的具有存储功能的两路选择器, 当 M4=1 时, 选中数据总线 DBUS; 当 M4=0 时, 从指令寄存器 IR 的低 4 位 IR3~IR0 接收数据。当 LDR4=1 时, 在 T2 的下降沿将选中的数据打入 R4。

ALU2 是由一片 GAL22V10 构成, 当 PC_ADD=1 时, 完成 PC 和 IR 低 4 位的相加, 即 PC+D。

程序计数器 PC 是由一片 GAL22V10 构成, 当 PC_INC=1 时, 完成 PC+1; 当 PC_ADD=1 时, 与 ALU2 一起完成 PC+D 的功能; 当 LDPC=1 时, 接收从 ALU2 和 R4 来的地址, 实验是接收来数据总线 DBUS 的地址, 这些新的程序地址在 T4 的上升沿打入 PC 寄存器。

9) 指令寄存器 IR

指令寄存器 IR 是一片 74LS374。当 LDIR=1 时, 在 T4 的上升沿, 它从双端口存储器的右端口接收指令。指令的操作码部分 IR7—IR4 送往控制器译码, 产生数据通路的控制信号。指令的操作数部分送往寄存器堆 RF, 选择参与运算的寄存器。IR1、IR0 与 RD1、RD0 连接, 选择目标操作数寄存器; IR3、IR2 与 RS1、RS0 连接, 选择源操作数寄存器。IR1、IR0 也与 WR1、WR0 连接, 以便将运算结果送往目标操作数寄存器。

10) 机器指令

本实验系统设计了 11 条基本的机器指令, 均为单字长 (8 位) 指令, 指令功能及格式如下表所示。另一点需说明的是: 为了简化运算, 指令 JC D 中的 D 是一个 4 位的正数, 用 D3~D0 表示。

4、微程序控制器

控制器用来产生数据通路操作所需的控制信号。TEC-9 提供了一个微程序控制器, 以便能进行计算机组成原理基本实验。在进行课程设计时, 学生可设计自己的控制器。图 1.3 是控制器框图。

表 1 机器指令系统

名称	助记符	功能	指令格式							
			IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
加法	ADD Rd, Rs	Rd + Rs→Rd	0	0	0	0	Rs1	Rs0	Rd1	Rd0
减法	SUB Rd, Rs	Rd - Rs→Rd	0	0	0	1	Rs1	Rs0	Rd1	Rd0
逻辑与	AND Rd, Rs	Rd & Rs→Rd	0	0	1	1	Rs1	Rs0	Rd1	Rd0
存数	STA Rd, [Rs]	Rd→[Rs]	0	1	0	0	Rs1	Rs0	Rd1	Rd0
取数	LDA Rd, [Rs]	[Rs]→Rd	0	1	0	1	Rs1	Rs0	Rd1	Rd0
无条件转移	JMP [RS]	[RS] →PC	1	0	0	0	Rs1	Rs0	X	X
条件转移	JC D	若 C = 1 则 PC=D→PC	1	0	0	1	D3	D2	D1	D0
停机	STP	暂停执行	0	1	1	0	X	X	X	X
中断返回	IRET	返回断点	1	0	1	0	X	X	X	X
开中断	INTS	允许中断	1	0	1	1	X	X	X	X
关中断	INTC	禁止中断	1	1	0	0	X	X	X	X

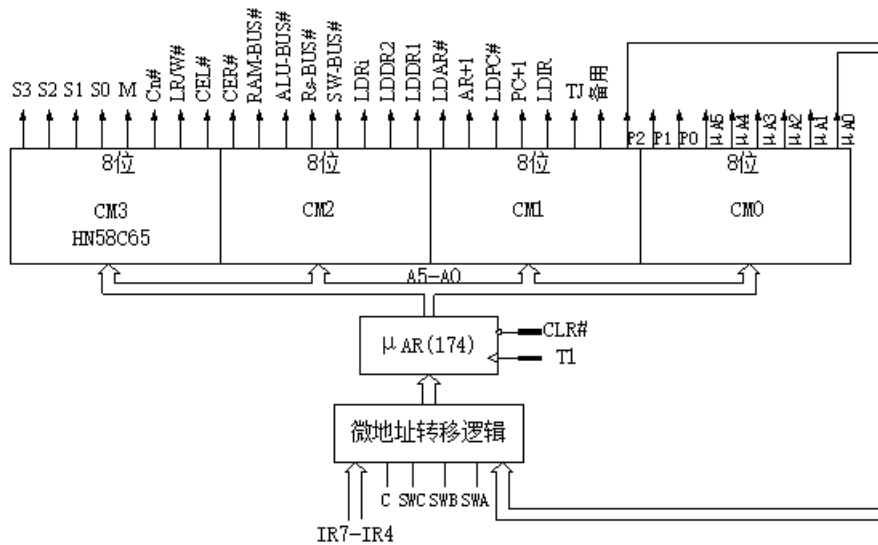


图 1.3 控制器框图

1) 控制存储器

控制存储器由 5 片 HN58C65 构成。HN58C65 是 E²PROM，存储容量为 8K 字节，本实验系统只使用了 128 字节。微指令格式采用水平型，微指令字长 35 位，其中顺序控制部分 10 位：判别字段 4 位，后继微地址 6 位。操作控制字段 25 位，各位进行直接控制。

判断标志位 P3 和控制台开关 SWB、SWA 结合在一起确定微程序的分支，完成不同的控制台操作。

判断标志位 P2 与指令操作码（IR 的高 4 位 IR7~IR4）结合确定微程序的分支。转向各种指令的不同微程序流程。

判断标志位 P1 标志一条指令的结束，与中断请求信号 INTQ 结合，实验对程序的中断

处理。

判断标志位 P0 与进位标志 C 结合确定微程序的分支，实验条件转指令。

操作控制字段 25 位，全部采用直接表示法，控制数据通路的操作。在设计过程中，根据微程序流程图对控制信号进行了适当的综合与归并，把某些在微程序流程图中作用相同或者类似的信号归并为一个信号。下面列出微程序提供的控制信号。

表 1.2 控制信号表

S3, S2, S1, S0	选择运算器的运算类型。
M	选择运算器的运算模式：M=0，算术运算；M=1，逻辑运算。
Cn#	运算器最低位的 +1 信号。为 0 时，运算器最低位有进位。
LRW	当 LRW=1 且 CEL=1 时，对双端口存储器左端口进行读操作；当 LRW=0 且 CEL=1 时，对左端口进行写操作。
CEL	双端口存储器左端口使能信号。为 1 时允许对左端口读、写。
CER	双端口存储器右端口使能信号。为 1 时将指令送往指令总线 IBUS。
M1 (M2)	当 M1=1 时，操作数寄存器 DR1 从数据总线 DBUS 接收数据，当 M1=0 时，操作数寄存器 DR1 从寄存器堆 RF 接收数据。此信号也用于作为操作数寄存器 DR2 的数据来源选择信号。
ALU_BUS	ALU 输出三态门使能信号，为 1 时将 ALU 运算结果送 DBUS。
RS_BUS	通用寄存器右端口三态门使能信号，为 1 时将 RF 的 B 端口数据送 DBUS。
SW_BUS	控制台输出三态门使能信号，为 1 时将控制台开关 SW7-SW0 数据送 DBUS。
WRD	双端口寄存器堆写入信号，为 1 时将数据总线上的数据在 T2 的上升沿写入由 WR1、WR0 指定的个寄存器。
LDDR2	对操作数寄存器 DR2 进行加载的控制信号，为 1 时在 T3 的下降沿将由 RS1、RS0 指定的寄存器中的数据打入 DR2。
LDDR1	对操作数寄存器 DR1 进行加载的控制信号，为 1 时在 T3 的下降沿将由 RD1、RD0 指定的寄存器中的数据打入 DR1。
LDAR1(LDAR2)	对地址寄存器 AR1 进行加载的控制信号。此信号也可用于作为允许对地址寄存器 AR2 加载。
AR1_INC	对 AR1 进行加 1 操作的电位控制信号。
LDPC(LDR4)	为 1 时，对程序计数器 PC 进行加载。此信号也用于作为 R4 的加载允许信号 LDR4
PC_INC	为 1 时，对 PC 进行加 1 操作的电位控制信号。
LDIR(CER)	为 1 时，对指令寄存器进行加载的控制信号。
INTS	置中断允许标志 INTE 为 1

INTC	清除中断允许标志 INTE
M4	当 M4=1 时，R4 从数据总线 DBUS 接收数据。 当 M4=0 时，R4 从指令寄存器 IR 接收数据。
M3	当 M3=1 时，AR2 从数据总线 DBUS 接收数据。 当 M3=0 时，AR2 从程序计数器 PC 接收数据
LDER	为 1 时，允许对暂存寄存器 ER 加载
LDIAR	为 1 时，对中断寄存器 IAR 加载。
TJ	停机指令，暂停微程序运行。

2) 微地址寄存器 μ AR

微地址寄存器 μ AR 是 1 片 74LS273，对控制存储器提供微程序地址。当 CLR#=0 时，将异步清零，使微程序从 000000B 开始执行。在每一个 T1 的上升沿，新的微指令地址打入微地址寄存器中。微地址由指示灯 uA5-uA0 显示。控制台信号 SWC 直接连接 74LS273，作为 uD6，用于实验读寄存器操作。

3) 微程序地址译码电路

微程序地址译码电路产生后继微程序地址，它由 2 片 74LS32 和 2 片 74LS08 构成。微程序地址译码电路数据来源是：控制存储器产生的后继微程序地址 uA5~uA0,控制存储器产生的标志位 P3~P0，指令操作码 IR7~IR4，进位标志位 C，中断请求标志 INTQ，控制台方式标志位 SWB、SWA。

5、控存 EEPROM 的改写

TEC-9 中的 5 片 EEPROM (CM4-CM0) 是控存，里面装有微程序的微代码。由于它是电可擦除和编程的 EEPROM，因此可以实现不用将 CM4-CM0 从插座上取出就能实现对其编程的目的。

6、控制台

控制台由若干拨动开关和指示灯组成，用于设置控制台指令、人工控制数据通路、设置数据代码信号和显示相关数据组成等。

1) 数据开关 SW7-SW0

八位数据开关，通过 74LS244 接到数据通路部分的数据总线 DBUS 上，用于向数据通路中的寄存器和存储器置数。当 SW_BUS=1 时，SW7—SW0 的数据送往数据总线 DBUS。开关拨到上面位置时输出 1，开关拨到下面位置时输出 0。SW7 对应 DBUS 最高位，SW0 对应 DBUS 最低位。

2) 模拟数据通路控制信号开关 K15-K0

拨动开关，拨到上面位置输出 1，拨到下面位置输出 0。实验中用于模拟数据通路部分所需的电平控制信号。例如，将 K1 与 LDDR1 相连，则 K1 拨到上面位置时，表示 LDDR1 为 1。这些开关在数字逻辑与数字系统实验时也作为电平输入开关。

3) 数据总线指示灯 DBUS

八个发光二极管（高四位为红，低四位为绿），指示 DBUS 上数据。灯亮表示 1。

4) 指令总线指示灯 IBUS

八个发光二极管（高四位为红，低四位为绿），指示 IBUS 上数据。灯亮表示 1。

5) 地址指示灯 AR

八个发光二极管（高四位为红，低四位为绿），指示双端口存储器的左端口地址寄存器内容。灯亮表示 1。

6) 程序计数器指示灯 PC

八个发光二极管（高四位为红，低四位为绿），指示双端口存储器右端口地址。灯亮表示 1。

7) 40 位微命令指示灯 CM3-CM0

40 个红色发光二极管，显示从控制存储器读出的微命令的内容。

8) 其它指示灯 C、BUSYL、BUSYR

C 是进位标志指示灯。BUSYL、BUSYR 分别是 RAM 左右端口忙指示灯。

9) 运算器指令灯

A7~A0，运算器 A 口数据指示灯；B7~B0，运算器 B 口数据指示灯；F7~F0，运算器运算结果输出数据指示灯

10) 微动开关 CLR#、QD

按一次 CLR# 开关，产生一个负的单脉冲 CLR#，正的单脉冲 CLR。CLR# 对全机进行复位。CLR# 到时序和控制器的连接已经在印制板上实现，控制存储器和数据通路部分不使用 CLR#。按一次 QD 按钮，产生一个正的启动脉冲 QD 和负的单脉冲 QD#。QD 使机器运行。

11) 工作方式选择开关

a、工作模式设置开关

SWC	SWB	SWA	功能
0	0	0	启动程序:程序从指定的地址开始运行
0	0	1	读双端口存储器
0	1	0	写双端口存储器
0	1	1	写寄存器堆
1	0	0	读寄存器堆

启动程序 (PR): 按下复位按钮 CLR# 后，微地址寄存器清零。这时，SWC=0、SWB=0、SWA=0，用数据开关 SW7—SW0 设置 RAM 中的程序首地址，按 QD 按钮后，启动程序执行。

写存储器 (WRM): 按下复位按钮 CLR#，置 SWC=0、SWB=1、SWA=0。①在 SW7—SW0 中置好存储器地址，按 QD 按钮将此地址打入 AR1。②在 SW7—SW0 置好数据，按

QD，将数据写入 AR1 指定的存储器单元，这时 AR 加 1。③返回②。依次进行下去，直到按复位键 CLR#为止。这样就实现了对 RAM 的连续手动写入。这个控制台操作的主要作用是**向 RAM 中写入自己编写的程序和数据**。

读存储器 (RRM)：按下复位按钮 CLR#，置 SWC=0、SWB=0、SWA=1。①在 SW7—SW0 中置好存储器地址，按 QD 按钮将此地址打入 AR1，RAM 此地址单元的内容读至 DBUS 显示。②按 QD 按钮，这时 AR1 加 1，RAM 新地址单元的内容读至 DBUS 显示。③返回②。依次进行下去，直到按复位键 CLR#为止。这样就实现了对 RAM 的连续读出显示。这个控制台操作的主要作用是检查写入 RAM 的程序和数据是否正确。在程序执行后检查程序执行的结果（在存储器中的部分）是否正确。

寄存器写操作 (WRF)：按下复位按钮 CLR#，置 SWC=0、SWB=1、SWA=1。①首先在 SW7—SW0 置好存储器地址，按 QD 按钮，则将此地址打入 AR1 寄存器和 AR2 寄存器。②在 SW1、SW0 置好寄存器选择信号 WR1、WR0，按 QD 按钮，通过双端口存储器的右端口将 WR1、WR0（即 SW1、SW0）送到指令寄存器 IR 的低 2 位。③在 SW7—SW0 中置好要写入寄存器的数据；按 QD 按钮，将数据写入由 WR1、WR0 指定的寄存器。④返回②继续执行，直到按复位按钮 CLR#。这个控制台操作主要在程序运行前，向相关的通用寄存器中置入初始数据。

寄存器读操作 (RRF)：按下复位按钮 CLR#，置 SWC=1、SWB=0、SWA=0。①首先在 SW7—SW0 置好存储器地址，按 QD 按钮，则将此地址打入 AR1 寄存器和 AR2 寄存器。②在 SW3、SW2 置好寄存器选择信号 RS1、RS0，按 QD 按钮，通过双端口存储器的右端口将 RS1、RS0（即 SW3、SW2）送到指令寄存器 IR 的第 3、2 位。RS1、RS0 选中的寄存器的数据读出到 DBUS 上显示出来。③返回②继续下来，直到按复位键 CLR#为止。这个控制台操作的主要作用是在程序执行前检查写入寄存器堆中的数据是否正确，在程序执行后检查程序执行的结果（在寄存器堆中的部分）是否正确。

b、控制器选择开关

微程序：选择控制器为微过程控制器，将自动一一对应连接好微程序信号与数据通路信号。

脱机：微过程控制器、数据通路、硬布线控制器三部分信号完全独立。

硬布线：选择控制器为硬布线控制器，将自动一一对应连接好硬布线控制器与数据通路间的信号。

c、DP、DZ、DB 三个开关只能有一个为高有效。

DP	DZ	DB	功能
0	0	0	连续运行
0	0	1	单步工作方式。硬布线
0	1	0	单指：运行一条指令。
1	0	0	单拍

DP（单拍）、DB（单步）是两种特殊的非连续工作方式。当 DP=1 时，计算机处于单拍工作方式，按一次 QD 按钮，只发送一组时序信号 T1-T4，执行一条微指令。

DB 方式只对硬连线控制器适用，当 DB=1 时，按一次 QD 按钮，发送一组 W1-W3，执行一条机器指令。当 DP=0 且 DB=0 时，DZ=0 时，TEC-9 处于连续工作方式，按 QD 按钮，连续执行双端口 RAM 中存储的程序。

7、硬连线控制器 EPM3128

EPM3128 是 Altera 公司的 1 个在系统可编程器件，包含有 2500 门，适用于设计大规模的数字逻辑与数字系统电路。在计算机模型实验中，它用作设计并实现硬连线控制器，代替出厂时提供的微过程控制器。它有 1 个下载插座，下载时下载电缆的一端插在下载插座上，另一端插在 PC 机并行口上，下载电缆将 PC 机和 EMP3128 连在一起。在 PC 机上运行 QUARTUS II 工具软件，输入控制器的设计方案，进行编译、连接和适配，然后下载到 EPM3128 中去，就构成了 1 个硬连线控制器。硬连线控制器和数据通路部分采用可插、拔的导线连接或通过控制器选择开关选通。

三、实验台监控使用说明

1、介绍

a、为了方便用户使用，实验台设有 LCD128X64 字符图形液晶显示部份，用户可以通过 PS2 键盘，非常方便的查看和修改微程序控制器内容、寄存器堆的内容、双端口存储器的内容。在用户实验时，可以时时的显示寄存器堆中 4 个寄存器的内容，和当前 PC 地址和 PC 地址后二个地址。及该地址中的内容。

b、实验台监控部份设有“恢复到出厂状态”，为了防止被学生恶意恢复，在恢复到出厂状态时需提供密码。

c、为了方便用户记忆密码，系统可以重新设置 10 位及 10 位以下密码。密码有效字符为 0~9 和字母 a~z。

d、查看和修改微程序控器和双端口存储器内容时，可能通过 PS2 键盘中部的方向键盘（上、下、左、右）和翻页（Page up、Page Down）进行查看或移动光标。

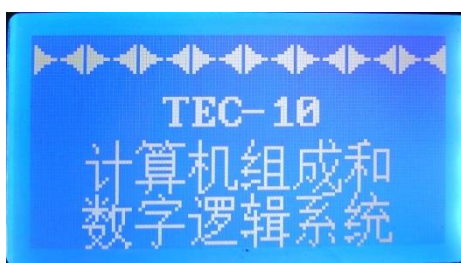
e、在修改微程序控制器和存储器时，只能输入有效的十六进制数字符。输入字符只能在 PS2 键盘主输入区，系统不能识别 PS2 辅助键盘。

2、使用说明

1) 打开实验台电源，监控系统显示如下图：

注意：接入 PS2 键盘时，必须在关电的情况下插入。不能带电拔插 PS2 键盘。

密码输入有效字符为 0~9 和 a~z 字母。以回车结束密码输入。



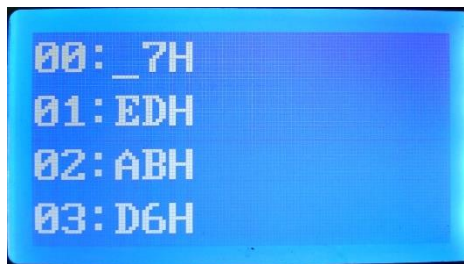
2) 按 F10 键，进入主菜单状态如下图，进入主菜单后，实验台上控制台信号断开，如果需进实验操作，须退出主菜单状态，回到初始菜单（上图）。



主菜单功能按键介绍：

a、F2：写双端口存储器

按 F2 进入写双端口存储器。如下图：



左边显示为存储器地址，右边为该地址内容。通过方向键移动光标，将不修改该内容，如果输入

有效的十六进制字符，自动修改该位置处内容。

b、F3：读双端口存储器

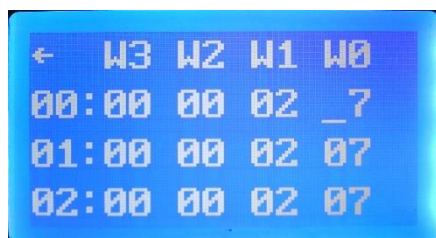
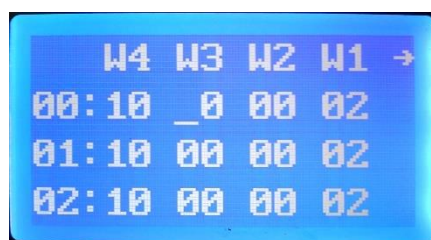
按 F3 进入读双端口存储器。如下图：



左边显示为存储器地址，右边显示该地址内容。可以通过上下方向键和翻页键查看其它地址的内容。

c、F4：写微程序控制器

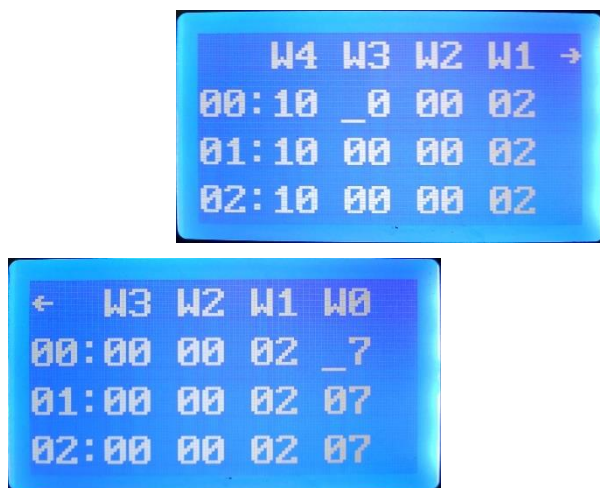
按 F4 进入微程序控制器内容。为防止学生误操作，在进入该功能时需输入修改密码。密码正确后进入修改状态。如下图：



左边显示为存储器地址，右边为该地址内容。通过方向键移动光标，将不修改该内容，如果输入有效的十六进制字符，自动修改该位置处内容。LCD 一屏只能显示 4 个控存内容，当光标移动到一行的最后一列或第一列时，继续移动光标将自动切换显示控存内容。

d、F5：读微程序控制器

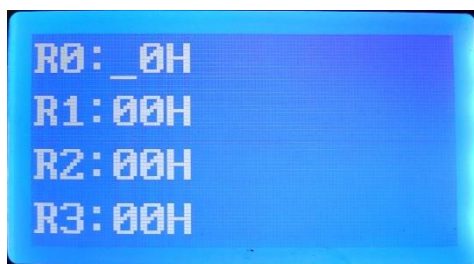
按 F5 键，进入读微程序控制器内容。如下图：



左边显示为存储器地址，右边为该地址内容。LCD 一屏只能显示 4 个控存内容，当光标移动到一行的最后一列或第一列时，继续移动光标将自动切换显示控存内容。

e、F6: 写寄存器

按 F6 键，进入写寄存器。如下图：



左边显示寄存器号，右边显示对应的寄存器号中的内容。通过方向键移动光标，将不修改该内容，如果输入有效的十六进制字符，自动修改该位置处内容。

四、HQFC-B1 软件使用说明

1、介绍

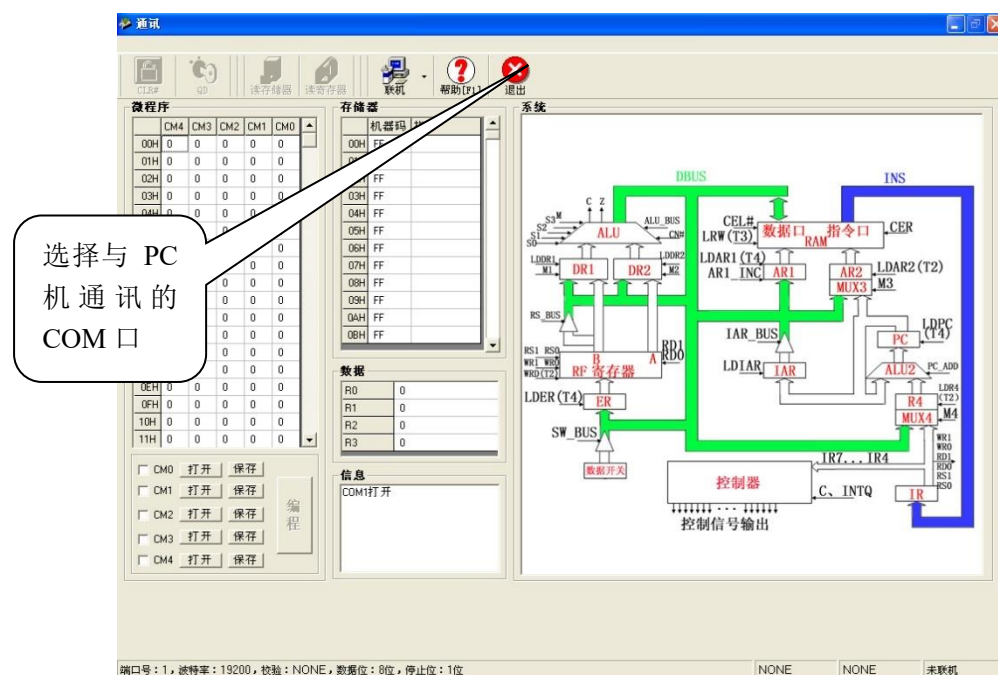
HQFC-B1 软件为 TEC 系列部分实验箱共有软件。为方便用户使用，实验台提供配件软件，非常方便的对实验台进行操作。可以通过软件非常方便的修改和查看实验台上控存、存储器、寄存器内容。

2、使用说明

- 安装完成软件后，在 PC 机操作系统/所有程序/HQFC-B1 计算机组成原理.EXE
- 打开软件如下图：(如下图)
- 点击“联机”。如下图：
- 联机成功后，点击存储器和寄存器区时，非常方便修改存储器和寄存器内容。修改内容以回车结束输入并修改。
- 通过软件的“CLR#”和“QD”键对实验台操作，上端软件非常方便的查看实验台运行情况。如下图：
- 如果微程序代码为出厂代码，右边将跟踪显示运行状态。如果微程序代码为用户代码时，右边将不跟踪显示运行状态。
- 为了避免微程序代码被学生恶意破坏，在对微程序重新编程时，需在“输入密码：”框中，输入有效的密码。如果需更改密码，需用实验台监控系统修改密码。

3、软件运行要求

- 显示分辨率至少应为 1024*768，或更高。
- PC 机串口一个或 USB 口一个（使用 USB 口时需选购 USB-COM 通信线一条）。
- 软件只能识别 COM1~COM4，如果使用 USB-COM 通信线时，虚拟 COM 端口号须在 COM1~COM4 之间的端口号。



打开时软件界面

The screenshot shows the J-EC software interface with the following components and callouts:

- 修改缓冲区内容** (Modify buffer content): Points to the CM (Control Memory) table on the left.
- 打开文件的内容或，联机后为当前控存内容** (Open file content or, after online connection, current control memory content): Points to the '打开' (Open) button in the CM table.
- 点击该列时，修改存储器内容。** (Click this column to modify memory content): Points to the '15H' address in the CM table.
- 点击该列时，修改寄存器内容** (Click this column to modify register content): Points to the '15H' address in the CM table.
- 选择需要写入的CM** (Select the CM to be written): Points to the 'CM0' checkbox in the CM table.
- 打开二进制文件或以二进制格式保存缓冲区的内容** (Open binary file or save buffer content in binary format): Points to the '保存' (Save) button in the CM table.
- 存储器/寄存器的内容，蓝底/红底表示当前运行位置。** (Memory/register content, blue/red background indicates current running position): Points to the '15H' address in the CM table.
- 如果微代码为出厂代码。控制器选择为“微程序”时，将时时跟踪显示运行状态。** (If microcode is factory code. When controller is selected as 'Microprogram', display running status in real-time tracking): Points to the '微程序' (Microprogram) checkbox in the CM table.

The interface includes a menu bar (文件, 读存储器, 读寄存器, 联机, 帮助[F1], 退出), a toolbar, and a main window displaying a flowchart and a register table.

寄存器堆 (Register Stack) table:

寄存器	内容
R0	80
R1	58
R2	00
R3	00

信息 (Information) table:

信息	内容
2	

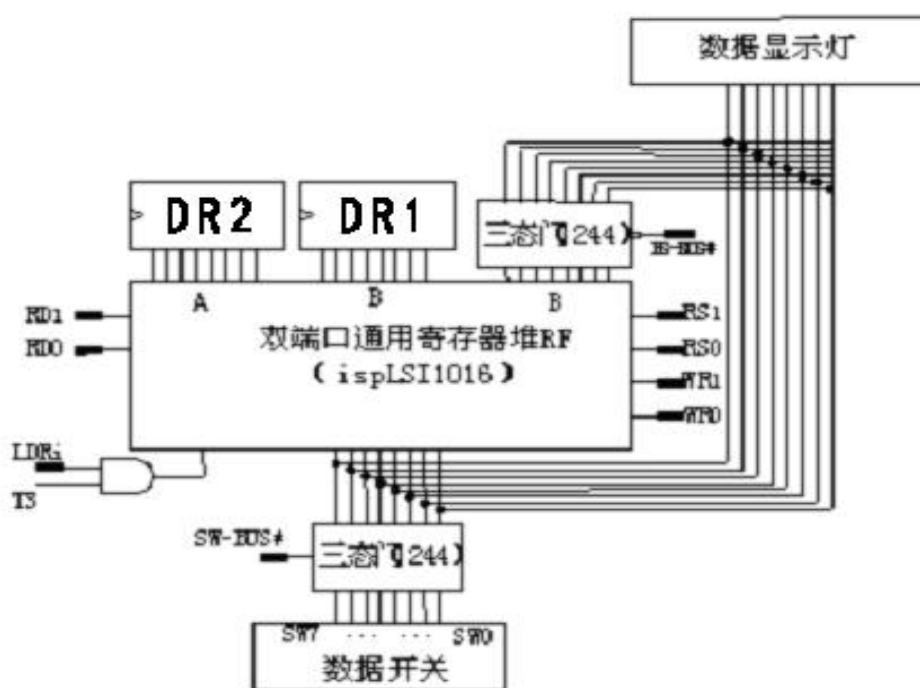
第二章 计算机组成原理实验

实验一 寄存器实验

一、实验目的

- (1) 熟悉寄存器堆的工作方法。
- (2) 掌握寄存器读写方法。

二、实验电路及原理



1、双端口通用寄存器堆 RF:

由一片 ispLSI1016 构成，其中包含 4 个 8 位寄存器 (R0、R1、R2、R3)，有三个控制端口：两个控制读操作，一个控制写操作，三个端口可以同时操作。

由 RD1、RD0 选中的寄存器的数据从 A 端口读出，00 为 R0，依次类推；

由 RS1、RS0 选中的寄存器的数据从 B 端口读出，00 为 R0，依次类推；

WR1、WR0 选择要写入的寄存器，00 为 R0，依次类推；

WRD 控制写操作：当 WRD=1 时，在 T2 上升沿将来自 ER 寄存器的数据写入由 WR1、WR0 选中的寄存器。

从 RF 的 A 端口读出的数据直接送 DR2。

由 B 端口读出的数据直接送 DR1 之外，还可以送数据总线 DBUS。当 RS_BUS=1 时，允许 B 端口数据送 DBUS；当 RS_BUS=0 时，禁止送往数据总线。

2、暂存寄存器 ER

暂存寄存器 ER 是一片 74LS374，主要用于暂时保存运算器的结果。当 LDER=1 时，在 T4 的上升沿，将数据总线 DBUS 上的数据打入暂存寄存器 ER。ER 的输出送往多端口通用寄存器堆 RF，作为写入数据使用。

3、开关寄存器 SW_BUS

开关寄存器 SW_BUS 是一片 74LS244，用于将控制台数据开关 SW7~SW0 的数据送往数据总线 DBUS。当 SW_BUS=1 时，允许开关 SW7~SW0 的数据送往总线 DBUS。

三、实验设备

(1) TEC-9 计算机组成原理实验系统 1 台

四、实验接线

按表所示，正确连接寄存器堆与实验台上的电平开关 K0—K8。

信号	IAR_BUS	ALU_BUS	LRW	CEL	LDER
电平开关	GND	GND	GND	GND	VCC

信号	RD1	RD0	RS1	RS0	WR1	WR0	WRD	RS_BUS	SW_BUS
开关	K8	K7	K6	K5	K4	K3	K2	K1	K0

接线后，将所有开关拨到“0”的状态。

说明：ALU_BUS=0，关闭 ALU 向数据总线 DBUS 的输出。ALU_BUS=1，ALU 运算结果输出到 DBUS。SW_BUS = 1，开启数据开关 SW0 — SW7 向数据总线 DBUS 的输出。**注意，对于数据总线 DBUS（或者其它任何总线），在任一时刻，只能有一个数据源向它输出。观察数据指示灯状态是否与数据开关状态一致。**

五、实验任务及步骤

1、向寄存器堆中四个寄存器分别写入数据，使 R0=55H、R1=AAH、R2=0FH、R3=F0H。

- 置 DB=0、DZ=0、DP=1；脱机状态；
- 按复位按钮 CLR#，使实验系统处于初始状态

(1) 写寄存器 R0=55H

- 将所有开关置为 0；
- 将 SW7-0 置为 01010101（55H）；
- WR1 和 WR0 置为 00，选择 R0 寄存器；
- WRD 置为 1，允许写寄存器；
- SW_BUS 置为 1，将 SW7-0 的数据送往 DBUS；
- 按一次 QD 按钮，将 55H 写入 ER 寄存器；
- 再按一次 QD 按钮，将 ER 寄存器（55H）写入 R0。

(2) 参考 (1)，分别完成写寄存器 R1=AAH，R2=0FH，R3=F0H 的任务，并写出实验

操作过程。

2、读寄存器堆，验证写入 R0、R1、R2、R3 的值。

(1) 读寄存器 R0

- 将所有开关置为 0；
- RS1 和 RS0 置为 00，选择 R0 寄存器送往 B 端口；RS_BUS 置为 1，允许 B 端口的数据送往 DBUS；
- 按一次 QD 按钮，在 DBUS 的数据指示灯上可以看到 R0 寄存器存放的数据 55H。

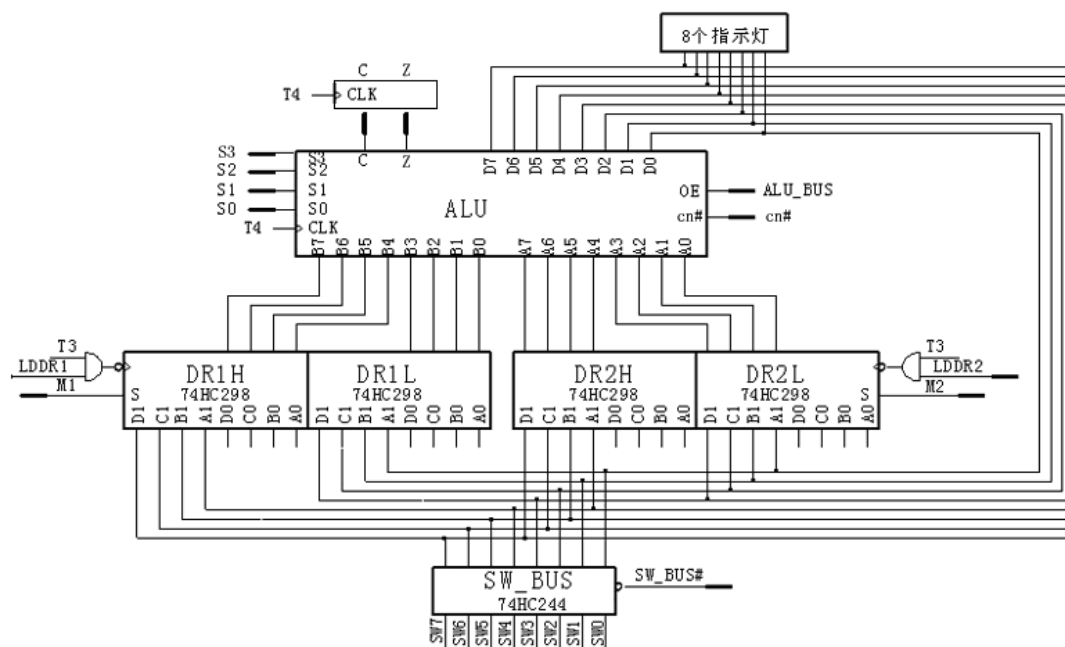
(2) 参考 (1)，分别完成读寄存器 R1，R2，R3 的任务，并写出实验操作过程，记录实验结果。

实验二 运算器组成实验

一、实验目的

- (1) 掌握算术逻辑运算加、减、与等的工作原理。
- (2) 熟悉简单运算器的数据传送通路。
- (3) 验证实验台运算器的 8 位加、减、与、直通功能。
- (4) 按给定数据，完成几种指定的算术和逻辑运算。

二、实验电路



图示出了本实验所用的运算器数据通路图。ALU 由 2 片 74LS181 构成，在选择端 M 和 S0-S3 控制下，ALU 对数据 A、B 进行各种算术、逻辑运算。当 LDRi=1 时，在 T4 的上升沿寄存器 C 保存运算产生的进位标志信号。

DR1 和 DR2 是运算操作数寄存器，DR1 和 ALU 的 B 口相连，DR2 和 ALU 的 A 口相

连。DR1 和 DR2 各由 2 片 74LS298 构成。ALU 的输出通过三态门发送到数据总线 DBUS7—DBUS0 上，进位信号 C、Z 保存在寄存器中。另有二个指示灯 C、Z 显示运算器进位信号状态。

当 M1=0 且 LDDR1=1 时，在 T3 的下降沿，DR1 接收来寄存器堆 B 端口的数据，当 M1=1 且 LDDR1=1 时，在 T3 的下降沿，DR1 接收来自数据总线 DBUS 的数据。

当 M2=0 且 LDDR2=1 时，在 T3 的下降沿，DR2 接收来自通用寄存器堆 A 端口的数据。当 M2=1 且 LDDR2=1 时，在 T3 的下降沿，DR2 接收来自数据总线 DBUS 的数据。

相关控制信号说明：

S3, S2, S1, S0	选择运算器的运算类型。
M	选择运算器的运算模式：M=0，算术运算；M=1，逻辑运算。
Cn#	运算器最低位的 +1 信号。为 0 时，运算器最低位有进位。
LRW	当 LRW=1 且 CEL=1 时，对双端口存储器左端口进行读操作；当 LRW=0 且 CEL=1 时，对左端口进行写操作。
M1 (M2)	当 M1=1 时，操作数寄存器 DR1 从数据总线 DBUS 接收数据，当 M1=0 时，操作数寄存器 DR1 从寄存器堆 RF 接收数据。此信号也用于作为操作数寄存器 DR2 的数据来源选择信号。
ALU_BUS	ALU 输出三态门使能信号，为 1 时将 ALU 运算结果送 DBUS。
RS_BUS	通用寄存器右端口三态门使能信号，为 1 时将 RF 的 B 端口数据送 DBUS。
SW_BUS	控制台输出三态门使能信号，为 1 时将控制台开关 SW7-SW0 数据送 DBUS。
WRD	双端口寄存器堆写入信号，为 1 时将数据总线上的数据在 T2 的上升沿写入由 WR1、WR0 指定的个寄存器。
LDDR2	对操作数寄存器 DR2 进行加载的控制信号，为 1 时在 T3 的下降沿将由 RS1、RS0 指定的寄存器中的数据打入 DR2。
LDDR1	对操作数寄存器 DR1 进行加载的控制信号，为 1 时在 T3 的下降沿将由 RD1、RD0 指定的寄存器中的数据打入 DR1。
LDAR1(LDAR2)	对地址寄存器 AR1 进行加载的控制信号。此信号也可用于作为允许对地址寄存器 AR2 加载。

由 74LS181 构成的 8 位运算器的运算类型由选择端 S3、S2、S1、S0 选择，功能如下表所示：

运算器运算类型选择表（*：表示移每一位到下更高有效位）

选择	高电平作用数据	
	M=1	M=0 算术运算

				逻辑功能	CN=1 无进位	CN=0 有进位
S3	S2	S1	S0			
0	0	0	0	$F = /A$	$F = A$	$F = A$ 加 1
0	0	0	1	$F = /(A+B)$	$F = A+B$	$F = (A+B)$ 加 1
0	0	1	0	$F = (/A)B$	$F = A+/B$	$F = (A+(/B))$ 加 1
0	0	1	1	$F = 0$	$F = \text{减 } 1$	$F = 0$
0	1	0	0	$F = /(AB)$	$F = A$ 加 $A(/B)$	$F = A$ 加 (A/B) 加 1
0	1	0	1	$F = /B$	$F = (A+B)$ 加 $(A(/B))$	$F = A$ 减 B
0	1	1	0	$F = A \oplus B$	$F = A$ 减 B 减 1	$F = A$ 减 B
0	1	1	1	$F = A(/B)$	$F = A(/B)$ 减 1	$F = A(/B)$
1	0	0	0	$F = /A+B$	$F = A$ 加 AB	$F = A$ 加 (AB) 加 1
1	0	0	1	$F = /(A \oplus B)$	$F = A$ 加 B	$F = A$ 加 B 加 1
1	0	1	0	$F = B$	$F = (A+(/B))$ 加 AB	$F = (A+/B)$ 加 (AB) 加 1
1	0	1	1	$F = AB$	$F = AB$ 减 1	$F = AB$
1	1	0	0	$F = 1$	$F = A$ 加 A^*	$F = A$ 加 A 加 1
1	1	0	1	$F = A+/B$	$F = (A+B)$ 加 A	$F = (A+B)$ 加 A 加 1
1	1	1	0	$F = A+B$	$F = (A+(/B))$ 加 A	$F = (A+(/B))$ 加 A 加 1
1	1	1	1	$F = A$	$F = A$ 减 1	$F = A$

注意：表中 $A+B$ 表示逻辑运算， A 加 B 是算数运算。

选择工作方式后，C、Z 需按一次 QD 保存于寄存器。

进位 C 只在加法运算和减法运算时产生，与、直通操作不影响进位 C 的状态，即进位 C 保持不变。减法运算采用加减数的反码再加以 1 实现。在加法运算中，C 代表进位；在减法运算中，C 代表借位。运算产生的进位在 T4 的上升沿送入 C 寄存器保存。

在 SW_BUS 信号为 1 时，参与运算的数据通过一个三态门 74HC244 (SW_BUS) 送到 DBUS 总线上，进而送至 DR1 或 DR2 操作数寄存器。输入数据可由实验台上的 8 个二进制数据开关 SW0—SW7 来设置，其中 SW0 是最低位，SW7 是最高位。开关向上时为 1，开关向下时为 0。

控制信号均为电位信号。T3、T4 是脉冲信号，印制板上已连接到实验台的时序电路产生的 T3、T4 信号上。S3、S2、S1、S0、ALU_BUS、LD_DR2、LDDR1、M1、M2、SW_BUS 各电位控制信号用电平开关 K0—K15 来模拟。

K0—K15 是一组用于模拟各控制电平信号的开关，开关向上时为 1，开关向下时为 0，每个开关无固定用途，可根据实验具体情况选用。S3、S2、S1、S0、ALU_BUS、LDDR2、LDDR1 为高电平有效，SW_BUS 为高电平有效。M1=1 时，DR1 选择数据总线作为数据输入端；M1=0 时，DR1 选择寄存器 B 口作为数据输入端。当 LDDR1=1 时，在 T3 的下降沿，选中的数据被打入 DR1 寄存器。M2=1 时，DR2 选择数据总线作为数据输入端；M2=0 时，DR2 选择寄存器 A 口作为数据输入端。当 LD_DR2=1 时，在 T3 的下降沿，选中的数据被

打入 DR2 寄存器。

数据总线 DBUS 有 5 个数据来源: 运算器 ALU, 寄存器堆 RF, 控制台开关 SW0—SW7, 双端口存储器 IDT7132 和中断地址寄存器 IAR。在任何时刻, 都不允许 2 个或者 2 个以上的数据源同时向数据总线 DBUS 输送数据, 只允许 1 个(或者没有)数据源向数据总线 DBUS 输送数据。在本实验中, 为了保证数据的正确设置和观察, 请令 RS_BUS = 0; LRW = 0; IAR_BUS = 0。

为了在实验中, 每次只产生一组 T1、T2、T3、T4 脉冲, 需将实验台上的选择开关设置为单拍, 每按一次 QD 按钮, 则顺序产生 T1、T2、T3、T4 各一个单脉冲。本实验中采用单脉冲输出。

三、实验设备

TEC-9 计算机组成原理实验系统 1 台

四、接线及初步验证

正确连接运算器模块与实验台上的电平开关 K0—K15。由于运算器的 C 和 C 指示灯、8 位数据开关 SW0—SW7、T3、T4 的连线已由印制电路板连好, 故接线任务仅仅是完成有关控制信号与电平开关 K0—K15 的连线。正确选择开关为单拍。用数据开关 SW0—SW7 向 DR1 和 DR2 寄存器置数。禁止中断寄存器、寄存器堆、双端口存储器向总线送数。M1、M2=1, 使 DR1、DR2 从数据总线接收数据。

数据通路	IAR_BUS	RS_BUS	LRW	CEL	M1	M2
电平开关	GND	GND	GND	GND	VCC	VCC

信号	S3	S2	S1	S0	M	CN#	LDDR2	LDDR1	ALU_BUS	SW_BUS
开关	K9	K8	K7	K6	K5	K4	K3	K2	K1	K0

五、实验任务及步骤

1、验证运算器的算术运算和逻辑运算功能。

(1) 实验要求: 令 DR1=01100011B, DR2=10110100B, 正确选择 S3、S2、S1、S0, 进行加、减、与、直通, 实验, 记下实验结果(数据和进位)并对结果进行分析。

(2) 实验步骤:

- 置 DB=0、DZ=0、DP=1; 脱机状态;
- 按复位按钮 CLR#, 使实验系统处于初始状态;
- 将所有开关置为 0;
- 将 SW7-SW0 置为 01010101 (55H);
- 将 LDDR1 置为 1, SW_BUS 置为 1;
- 按一次 QD, 将 55H 写入 DR1;
- 将 LDDR1 置为 0;

- 将 SW7-SW0 置为 10101010 (AAH);
- 将 LDDR2 置为 1;
- 按一次 QD, 将 AAH 写入 DR2;
- 将 S3-S0、M、CN 置为 1001 0 1, 设置 ALU 工作状态为 A 加 B, 不带进位, 完成 DR1 和 DR2 内容相加。
- 将 SW_BUS 置为 0, ALU_BUS 置为 1, 观察 DBUS 指示灯, 显示为 FFH。
- 置 S3-S0、M、CN 为 0110 0 0, 置 ALU 工作状态为 A 减 B, DBUS 指示灯 ()。
- 置 S3-S0、M、CN 为 0000 0 1, 置 ALU 工作状态为 A 直通, DBUS 指示灯 ()。
- 置 S3-S0、M、CN 为 1011 1 0, 置 ALU 工作状态为 A 与 B, DBUS 指示灯 ()。

2、M1、M2 控制信号的作用是什么? 改变 M1、M2 的高低电平, 重复 2 中一组数据的运算, 观察出现什么问题?

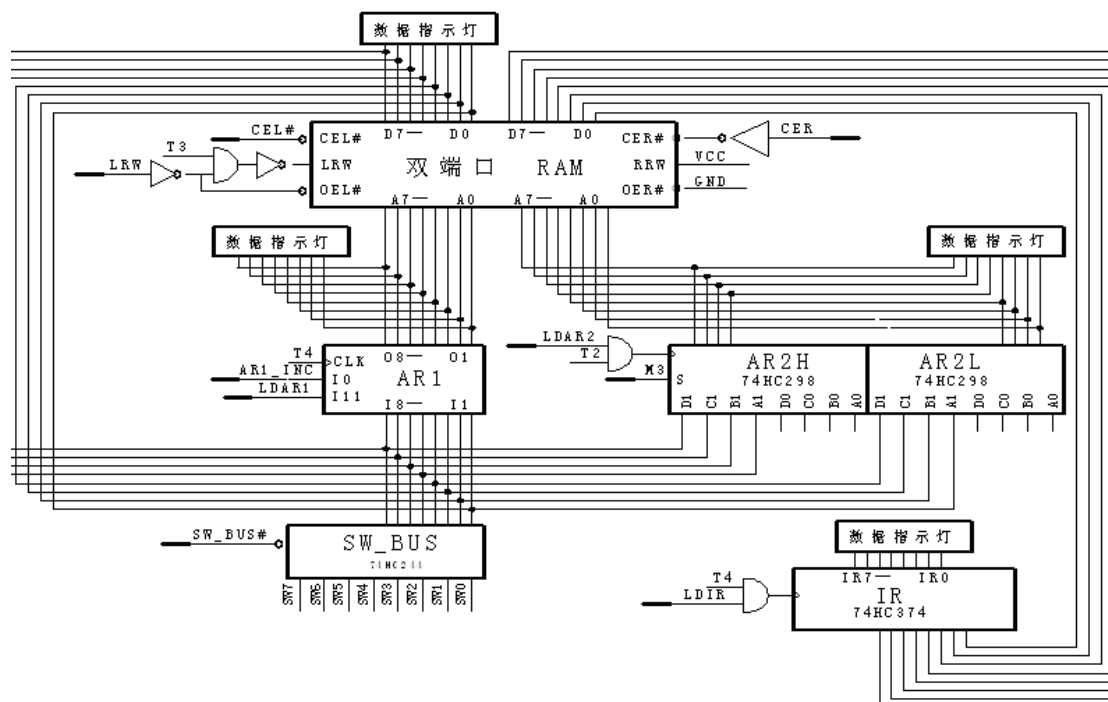
3、结合实验一设计实验, 写出将 01100011B 写入 R1, 将 01100011B 写入 R3, 再利用 ALU 完成 R1 加 R3 和 R1 减 R3。写出实验步骤, 并记录结果。

实验三 双端口存储器原理实验

一、实验目的

1. 了解双端口静态存储器 IDT7132 的工作特性及其使用方法
2. 了解半导体存储器怎样存储和读取数据。
3. 了解双端口存储器怎样并行读写, 并分析冲突产生的情况。

二、实验电路及原理



上图示出了双端口存储器的实验电路图。

使用一片 IDT7132 (U36) (2048*8 位) 静态随机存储器, 两个端口的地址输入 A8—A10 引脚接地, 因此实际使用存储容量为 256 字节。IDT7132 的两个端口可以同时进行读、写操作, 在本实验系统中, RAM 左端口连接数据总线 DBUS, 可进行读、写操作; 右端口连接指令总线 IBUS, 输出到指令寄存器 IR, 作为只读端口使用。左端口的数据可以在 DBUS7—DBUS0 数据指示灯上观察, 右端口的数据可在指令总线 IR7—IR0 的数据指示灯观察。

一片 GAL22V10 (U37) 作为左端口的地址寄存器 (AR1), 内部具有地址递增的功能。AR1 从数据总线 DBUS 接收数据。两片 4 位的 74HC298 (U28、U27) 作为右端口的地址寄存器 (AR2H、AR2L), 带有选择输入地址源的功能, 有两个数据输入端, 一个来自程序计数器 PC, 另一个来自数据总线 DBUS。

LRW	当 LRW=1 且 CEL=1 时, 对双端口存储器左端口进行读操作; 当 LRW=0 且 CEL=1 时, 对左端口进行写操作。
CEL	双端口存储器左端口使能信号。为 1 时允许对左端口读、写。
CER	双端口存储器右端口使能信号。为 1 时将指令送往指令总线 IBUS。
AR1_INC	对 AR1 进行加 1 操作的电位控制信号。
LDPC(LDR4)	为 1 时, 对程序计数器 PC 进行加载。此信号也用于作为 R4 的加载允许信号 LDR4
PC_INC	为 1 时, 对 PC 进行加 1 操作的电位控制信号。
LDIR(CER)	为 1 时, 对指令寄存器进行加载的控制信号。
M4	当 M4=1 时, R4 从数据总线 DBUS 接收数据。 当 M4=0 时, R4 从指令寄存器 IR 接收数据。
M3	当 M3=1 时, AR2 从数据总线 DBUS 接收数据。 当 M3=0 时, AR2 从程序计数器 PC 接收数据

存储器 IDT7132 有 6 个控制引脚: CEL、LRW、OEL、CER、RRW、OER。CEL、LRW、OEL 控制左端口读、写操作, CER、RRW、OER 控制右端口读、写操作。

(1) 左端口操作

CEL 为左端口选择引脚, 高有效。当 CEL=0 时, 禁止左端口读、写操作; 当 CEL=1 时, 允许左端口读、写操作。当 LRW 为高时, 左端口进行读操作; 当 LRW 为低时, 左端口进行写操作。当 OEL 为低时, 将左端口读出的数据放到数据总线 DBUS 上; 当 OEL 为高时, 禁止左端口读出的数据放到数据总线 DBUS 上。OEL 为 LRW 反向产生。因此本实验台 OEL 信号不需单独控制。实验台上已连接 T3 到时序发生器的 T3 输出。

- 当 CEL=1 且 LRW=1 时, 左端口进行读操作, 同时将读出的数据放到数据总线 DBUS 上。

- 当 CEL=1 且 LRW=0 时, 在 T3 的上升沿开始进行写操作, 将数据总线上的数据写入

存储器。

(2) 右端口操作

CER、RRW、OER 控制右端口读、写操作的方式与 CEL、LRW、OER 控制左端口读、写操作的方式类似，不过右端口读出的数据放到指令总线上而不是数据总线上。实验台上已连接 T3 到时序发生器的 T3 输出。实验台上 OER 已固定接地，RRW 固定接高电平，CER# 由 CER 反相产生。

- 当 CER=1 且 LDIR=1 时,右端口读出的指令在 T4 的上升沿打入 IR 寄存器。

(3) 存储器地址操作

存储器的地址由地址寄存器 AR1、AR2 提供，而 AR1 和 AR2 的内容根据数码开关 SW0—SW7 设置产生，并经三态门 SW_BUS 发送到数据总线时被 AR1 或 AR2 接收，三态门的控制信号 SW_BUS 是低电平有效。数据总线 DBUS 有 5 个数据来源：运算器 ALU，寄存器堆 RF，控制台开关 SW0—SW7，双端口存储器 IDT7132 和中断地址寄存器 IAR。在任何时刻，都不允许 2 个或者 2 个以上的数据源同时向数据总线 DBUS 输送数据,只允许 1 个(或者没有)数据源向数据总线 DBUS 输送数据。

在本实验中，为了保证数据的正确设置和观察，请置 RS_BUS = 0，ALU_BUS = 0,IAR_BUS=0。AR1 的控制信号是 LDAR1 和 AR1_INC。当 LDAR1=1 时，AR1 从 DBUS 接收地址；当 AR1_INC =1 时，使 AR1 中的存储器地址增加 1；在 T4 的上升沿,产生新的地址；LDAR1 和 AR1_INC 两者不可同时为 1。AR2 的控制信号是 LDAR2 和 M3。当 M3=1 时，AR2 从数据总线 DBUS 接收数据；当 M3=0 时，AR2 以 PC 总线 PC0—PC7 作为数据来源。当 LDAR2=1 时，在 T2 的下降沿，将新的 PC 值打入 AR2。

三、实验设备

TEC-9 计算机组成原理实验系统 1 台

四、实验接线

按表所示，将有关控制信号和和二进制开关对应接好，仔细复查一遍，然后接通电源。

说明：IAR_BUS#接 VCC，ALU_BUS 接 GND，RS_BUS#接 VCC，禁止中断地址寄存器、运算器、多端口寄存器堆 RF 向数据总线 DBUS 送数据。AR1_INC 接 GND,M3 接 VCC，使地址寄存器 AR1 和 AR2 从数据总线 DBUS 取得地址数据。

信号	IAR_BUS	ALU_BUS	RS_BUS	AR1_INC	M3	PC_ADD	PC_INC	LDPC
开关	GND	GND	GND	K7	VCC	GND	GND	GND

信号	LDIR	SW_BUS	LDAR2	LDAR1	CER	LRW	CEL
开关	K6	K5	K4	K3	K2	K1	K0

五、实验任务及步骤

1、写存储器，向 00H、10H、20H、30H、40H 单元写入数据 11H、22H、33H、44H、55H。

(1) 将 11H 写入存储器的 00H 单元：

- 按实验连线将所有控制线和开关连接好。
- 置 DP = 1, DB = 0, DZ = 0, 使实验台处于单拍状态。
- 打开电源。按复位按钮 CLR#, 使实验系统处于初始状态。
- 将所有开关置为 0;
- 将数码开关 SW7—SW0 设置为 00H, 将此数据作为地址置入 AR1;
- 将 SW_BUS 置为 1, LDAR1 置为 1, 准备将 00H 写入 AR1;
- 按动 QD, 将 00H 打入 AR1;
- 将 LDAR1 置为 0, 将 LRW 和 CEL 置为 01;
- 重新设置 SW0—SW7, 置为 11H,
- 按动 QD, 将 11H 打入 AR1 指定的存储单元。

(2) 参考 (1) 的步骤, 依次向 10H、20H、30H、40H 单元写入数据 22H、33H、44H、55H (提示, 可以按照 (1) 的方法写入地址, 也可以利用 AR1_INC 让地址递增), 并记录实验结果。

2、使用双端口存储器的左端口, 依次读出存储器第 00H、10H、20H、30H、40H 单元中的内容, 观察上述各单元中的内容是否与该单元的地址号相同, 并记录数据。

(1) 读 00H 单元：

- 置 DP = 1, DB = 0, DZ = 0, 使实验台处于单拍状态。
- 打开电源。按复位按钮 CLR#, 使实验系统处于初始状态。
- 将所有开关置为 0
- 将数码开关 SW7—SW0 设置为 00H, 将此数据作为地址置入 AR1;
- 将 SW_BUS 置为 1, LDAR1 置为 1, 准备将 00H 写入 AR1;
- 按动 QD, 将 00H 打入 AR1;
- 将 LDAR1 和 SW_BUS 置为 0, 将 LRW 和 CEL 置为 11;
- 按动 QD, 在 DBUS 观察到 00H 单元中存储的数据 11H。

(2) 参考 (1) 的步骤, 依次读出 10H、20H、30H、40H 单元的数据, 写出并记录实验结果。

3、通过双端口存储器右端口 (指令端口), 依次把存储器第 00H、10H、20H、30H、40H 单元中的内容置入指令寄存器 IR, 观察结果是否与 2 相同。

(1) 通过有端口读 00H 单元内容, 并写入 IR

- 置 DP = 1, DB = 0, DZ = 0, 使实验台处于单拍状态。
- 打开电源。按复位按钮 CLR#, 使实验系统处于初始状态。
- 将所有开关置为 0
- 将数码开关 SW7—SW0 设置为 00H;

- 将 SW_BUS 置为 1，LDAR2 置为 1，准备将 00H 写入 AR2；
 - 按动 QD，将 00H 打入 AR2；
 - 将 LDAR2（避免修改 AR2）和 SW_BUS 置为 0（避免数据冲突），LDIR 置 1（允许写 IR 寄存器），将 LRW 和 CER 置为 11（数据从右端口输出）；
 - 按动 QD，在 IR 寄存器指示灯观察到 00H 单元中存储的数据 11H。
- （2）参考（1）的步骤，依次将 10H、20H、30H、40H 单元的内容写入 IR 寄存器，并记录实验结果。

4、双端口存储器的并行读写和访问冲突测试。

置 CEL=1 且 CER=1，使存储器左、右端口同时被选中。当 AR1 和 AR2 的地址不相同，没有访问冲突；地址相同时，则发生冲突。要检测冲突，可以观察 BUSYL 和 BUSYR 指示灯（分别是两个端口的“忙”信号输出）。BUSYL/R 亮时不一定发生冲突，但发生冲突时，BUSYL/R 一定亮。当某一个端口（无论是左端口还是右端口）的 BUSYL/R 亮时，对该端口的写操作被 IDT7132 忽略掉。

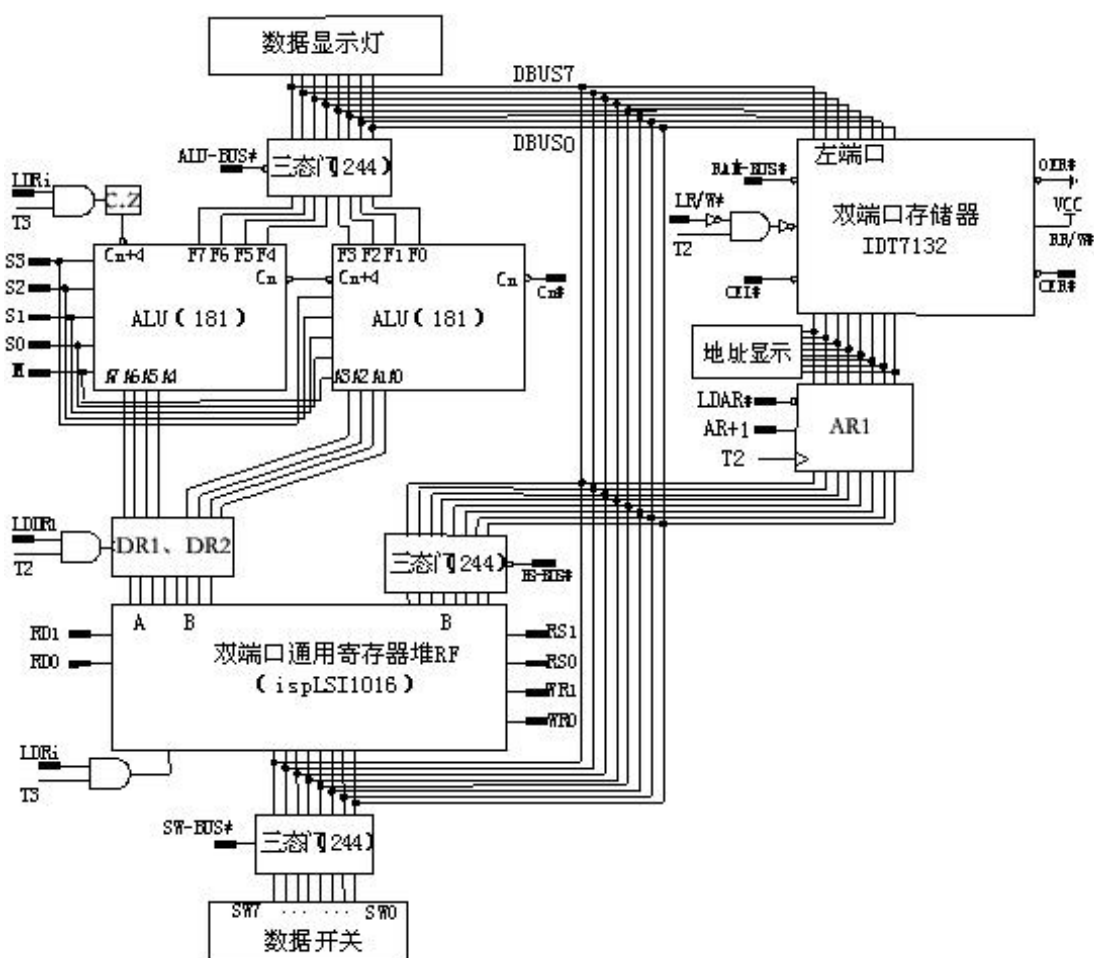
- 参考 1 和 3 的部分步骤，向 AR1 和 AR2 分别写入 38H 单元地址；
- 将 SW_BUS 置为 0，LDAR1 和 LDAR2 分别置为 0；
- 将 LRW、CER、CEL 置为 110，观察并记录实验现象；
- 将 LRW、CER、CEL 置为 010，观察并记录实验现象；
- 将 LRW、CER、CEL 置为 011，观察并记录实验现象；
- 将 LRW、CER、CEL 置为 001，观察并记录实验现象；
- 将 LRW、CER、CEL 置为 111，观察并记录实验现象；

第四节 数据通路组成实验

一、实验目的

1. 进一步熟悉计算机的数据通路。
2. 将双端口通用寄存器堆和双端口存储器模块连接，构成新的数据通路。
3. 掌握数字逻辑电路中的一般规律，以及排除故障的一般原则和方法。
4. 锻炼分析问题和解决问题的能力，在出现故障的情况下，独立分析故障现象，并排除故障。

二、实验电路



数据通路实验电路图如上图所示。它是将双端口存储器模块和双端口通用寄存器堆模块连接在一起形成的。存储器的指令端口(右端口)不参与本次实验。通用寄存器堆连接运算器模块，本次实验涉及其中的 DR1。

由于双端口存储器 RAM 是三态输出，因而可以将它直接连接到数据总线 DBUS 上。此外，DBUS 上还连接着双端口通用寄存器堆。这样，写入存储器的数据可由通用寄存器堆提供，而从存储器 RAM 读出的数据也可送到通用寄存器堆保存。

双端口存储器 RAM 已在第三节做过介绍，DR2 在第三节的实验中使用过。通用寄存器堆 RF 由一个 ISP1016 实现，功能上与两个 4 位的 MC14580 并联构成的寄存器堆类似。

RF 内含四个 8 位的通用寄存器 R0、R1、R2、R3，带有一个写入端口和两个输出端口，从而可以同时写入一路数据，读出两路数据。写入端口取名为 WR 端口，连接一个 8 位的暂存寄存器 ER，这是一个 74HC374。输出端口取名为 RS 端口（B 端口）、RD 端口（A 端口），连接运算器模块的两个操作数寄存器 DR1、DR2。RS 端口（B 端口）的数据输出还可通过一个 8 位的三态门 RSO 直接向 DBUS 输出。

双端口通用寄存器堆模块的控制信号中，RS1、RS0 用于选择从 RS 端口（B 端口）读出的通用寄存器，RD1、RD0 用于选择从 RD 端口（A 端口）读出的通用寄存器。而 WR1、WR0 则用于选择从 WR 端口写入的通用寄存器。WRD 是写入控制信号，当 WRD=1 时，在 T2 上升沿时刻，将暂存寄存器 ER 中的数据写入通用寄存器堆中由 WR1、WR0 选中的寄存器；当 WRD=0 时，ER 中的数据不写入通用寄存器中。LDER 信号控制 ER 从 DBUS 写入数据，当 LDER=1 时，在 T4 的上升沿，DBUS 上的数据写入 ER。RS_BUS 信号则控制 RS 端口到 DBUS 的输出三态门，是一个高有效信号。以上控制信号各自连接一个二进制开关 K0—K15。

三、实验设备

TEC-9 计算机组成原理实验系统 1 台

四、实验接线

信号	IAR_BUS	CER	AR1_INC	S3	S2	S1	S0	M	CN#	M2
开关	GND	GND	GND	VCC	GND	GND	VCC	GND	VCC	GND

信号	WRD	WR1	WR0	RD1	RD0	RS1	RS0	LDER
开关	K14	K13	K12	GND	GND	K11	K10	VCC

LDDR2	M1	LDDR1	LDAR1	LRW	CEL	ALU_BUS	RS_BUS	SW_BUS
K8	K7	K6	K5	K4	K3	K2	K1	K0

M3	PC_ADD	PC_INC	LDPC
VCC	GND	GND	GND

五、实验任务及步骤

1、用 8 位数据开关向 RF 中的 R0 写入 0FH。并将 R0 通过存储器左端口写入存储单元的 11H，并将 11H 单元中的数写入到 R1 中。

（1）将 0FH 写入 R0，再写入存储器的 11H 单元；

- 置 DB=0、DZ=0、DP=1；脱机状态；
- 按复位按钮 CLR#，使实验系统处于初始状态；
- 将所有开关置为 0；

写 R0:

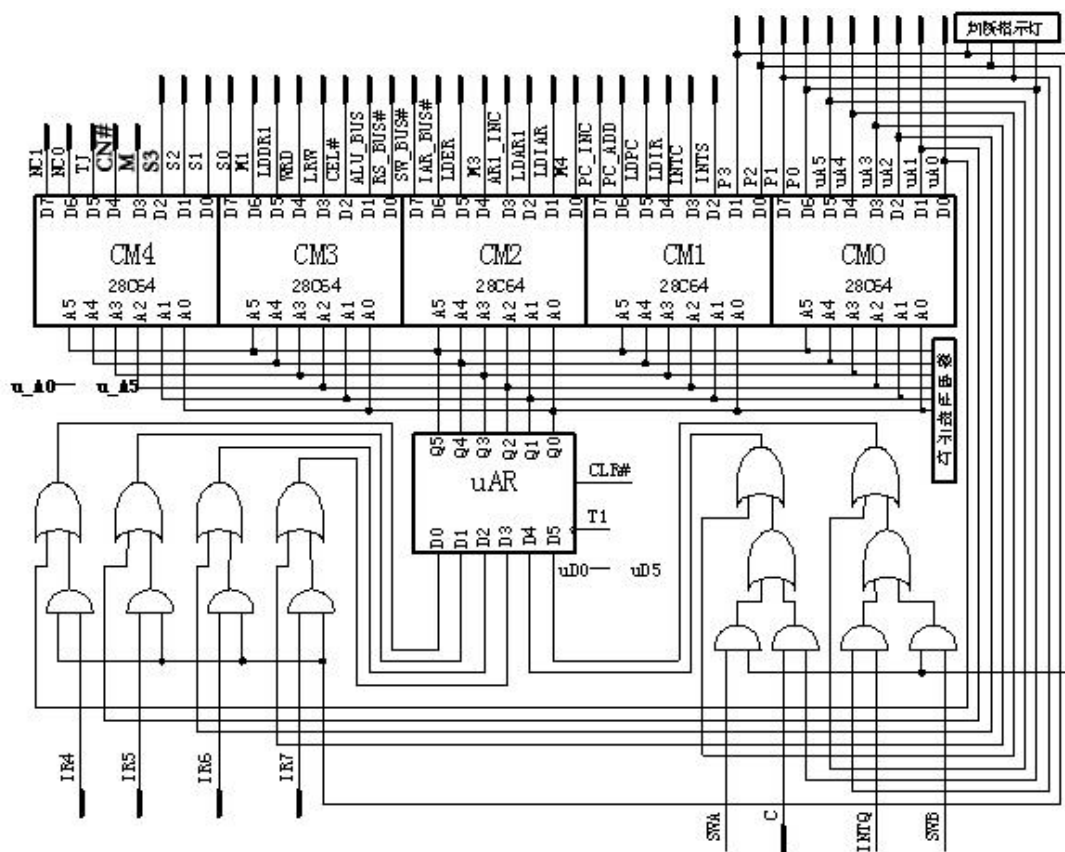
- 将 SW7-0 置为 00001111 (0FH)；
- WR1 和 WR0 置为 00，选择 R0 寄存器；
- WRD 置为 1，允许写寄存器；
- SW_BUS 置为 1，LDER 置为 1，将 SW7-0 的数据送往 DBUS；
- 按一次 QD 按钮，将 0FH 写入 ER 寄存器；
- 再按一次 QD 按钮，将 ER 寄存器 (0FH) 写入 R0。
- 将所有开关置为 0；

写存储器 11H 单元:

- 将 SW7-0 置为 00010001 (11H)，将此数据作为地址置入 AR1；
- 将 SW_BUS 置为 1，LDAR1 置为 1，准备将 11H 写入 AR1；
- 按动 QD，将 11H 打入 AR1；
- 将所有开关置为 0；
- RS1 和 RS0 置为 00，选择 R0 寄存器送往 B 端口；
- RS_BUS 置为 1，允许 B 端口的数据送往 DBUS；
- 将 LRW 和 CEL 置为 01，准备写存储单元；
- 按一次 QD 按钮，在 DBUS 的数据指示灯上可以看到 R0 寄存器存放的数据 0FH，同时将 0FH 打入 AR1 指定的存储单元；
- 将 RS_BUS 置为 0，LRW 和 CEL 置为 11，可以在 DBUS 上读到 AR1 指定单元的数据。

(2) 参考 (1)，将存储器 11H 单元中的数写入到 R1 中。设计实验步骤，并记录实验结果。

(3) 设计实验，在 R0 中写入 55H，存储单元 11H 中写入 AAH，完成“R0 加[11H]”的计算。设计实验步骤，并记录实验结果。(该实验已经将 S3-S0、M 和 CN 置为加的状态；)



微程序控制器的组成

对应微指令格式，微程序控制器的组成见上图。控制存储器采用 5 片 EEPROM 28C64(U8、U9、U10、U11、U12)。28C64 的输出是 D0—D7，分别与引脚 11、12、13、15、16、17、18、19 相对应，CM0 是最低字节，CM4 是最高字节。微地址寄存器 6 位，用一片 8D 触发器 74LS273 组成，带有清零端。两级与门、或门构成微地址转移逻辑，用于产生下一微指令的地址。在每个 T1 上升沿时刻，新的微指令地址会打入微地址寄存器中，控制存储器随即输出相应的微命令代码。微地址转移逻辑生成下一地址，等下一个 T1 上升沿时打入微地址寄存器。5 片 EEPROM 的地址 A6（引脚 4）直接与控制台开关 SWC 连接，当 SWC =1 时，微地址大于或者等于 40H，当 SWC=0 时，微地址的范围为 00H—3FH。SWC 主要用于实现读寄存器堆的功能。

微地址转移逻辑的多个输入信号中，INTQ 是中断请求，本实验中可以不理睬。SWA、SWB 是控制台的两个二进制开关信号，实验台上线已接好接。C 是进位信号，IR7—IR4 是机器指令代码，由于本次实验不连接数据通路，这些信号都接到二进制开关 K0—K15 上。

本实验仪设置了 11 条机器指令，均为单字长（8 位）指令。指令功能及格式如下表所示。指令的高 4 位提供给微程序控制器，低 4 位提供给数据通路。

指令功能与格式表

名称	助记符	功能	指令格式					
			R7 R6 R5 R4	R3 R2	R1 R0			
加法	ADD Rd,Rs	Rd+Rs->Rd	0 0 0 0	RS1 RS0	RD1 RD0			
减法	SUB Rd,Rs	Rd-Rs->Rd	0 0 0 1	RS1 RS0	RD1 RD0			
逻辑与	AND Rd,Rs	Rd&Rs->Rd	0 0 1 1	RS1 RS0	RD1 RD0			
存数	STA Rd,[Rs]	Rd->[Rs]	0 1 0 0	RS1 RS0	RD1 RD0			
取数	LDA Rd,[Rs]	[Rs]->Rd	0 1 0 1	RS1 RS0	RD1 RD0			
无条件转移	JMP [Rs]	[Rs]->PC	1 0 0 0	RS1 RS0	X X			
条件转移	JC D	若 C=1 则 PC+D->PC	1 0 0 1	D3 D2	D1 D0			
停机	STP	暂停运行	0 1 1 0	X X	X X			
中断返回	IRET	返回断点	1 0 1 0	X X	X X			
开中断	INTS	允许中断	1 0 1 1	X X	X X			
关中断	INTC	禁止中断	1 1 0 0	X X	X X			

为了向 RAM 中装入程序和数据，检查写入是否正确，并能启动程序执行，还设计了以下五个控制台操作微程序：

开关状态			功能
SWC	SWB	SWA	
0	0	0	从指定地址启动程序
0	0	1	读存储器
0	1	0	写存储器
0	1	1	写寄存器堆
1	0	0	读寄存器堆

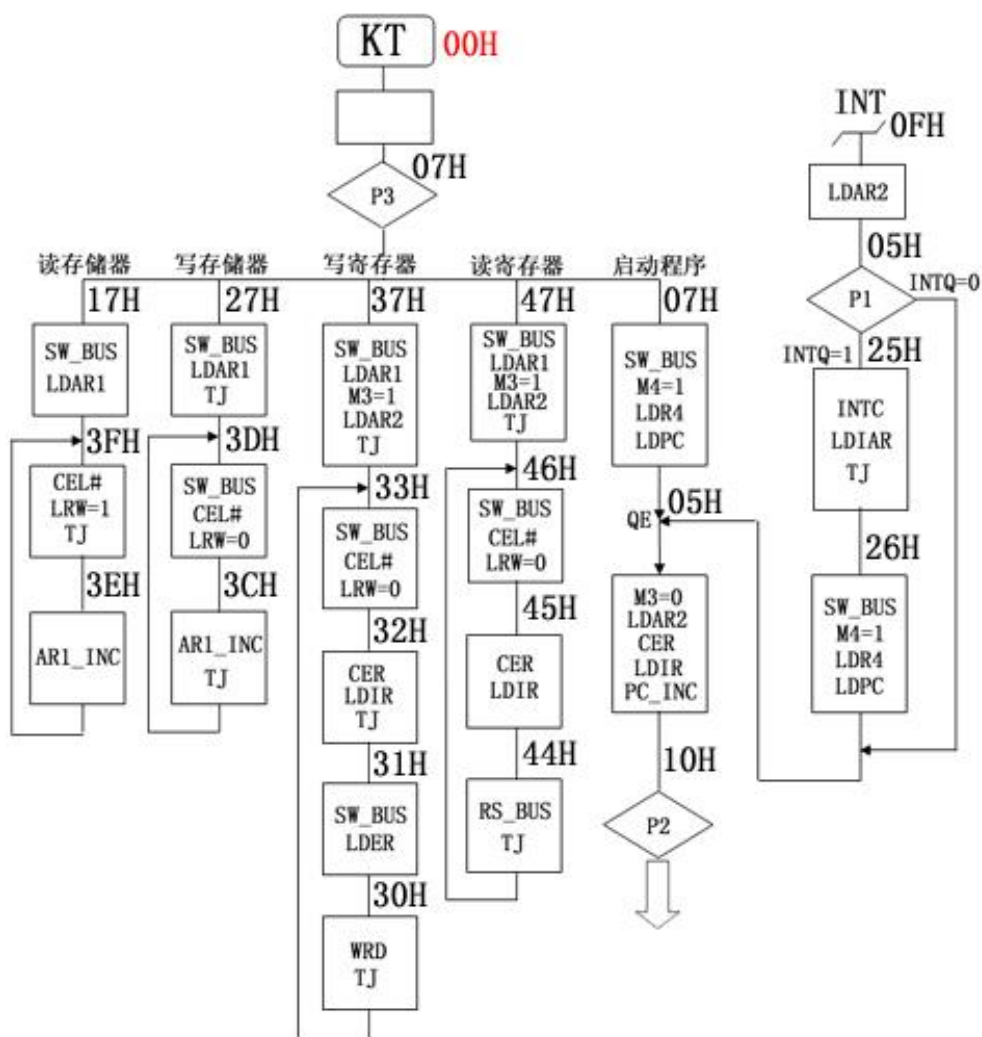
存储器写操作（KWE）：按下复位按钮 CLR#后，微地址寄存器状态为全零。此时置 SWC = 0、SWB = 1，SWA = 0，按启动按钮后微指令地址转入 27H 的微指令，从而可对 RAM 连续进行手动写入。

存储器读操作（KRD）：按下复位按钮 CLR#后，置 SWC = 0，SWB = 0，SWA = 1，按启动按钮后微指令地址转入 17H，从而可对 RAM 连续进行读操作。

写寄存器操作（KLD）：按下复位按钮 CLR#后，置 SWC = 0，SWB = 1，SWA = 1，按启动按钮后微指令地址转入 37H，从而可对寄存器堆中的寄存器连续进行写操作。

读寄存器操作（KRR）：按下复位按钮 CLR#后，置 SWC = 1，SWB = 0，SWA = 0，按启动按钮后微指令地址转入 47H，从而可对寄存器堆中的寄存器连续进行读操作。

启动程序（PR）：按下复位按钮 CLR#后，置 SWC = 0，SWB = 0，SWA = 0，用数据开关 SW7—SW0 设置内存中程序的首地址，按启动按钮后微指令地址转入 07H，然后转到“取指”微指令。



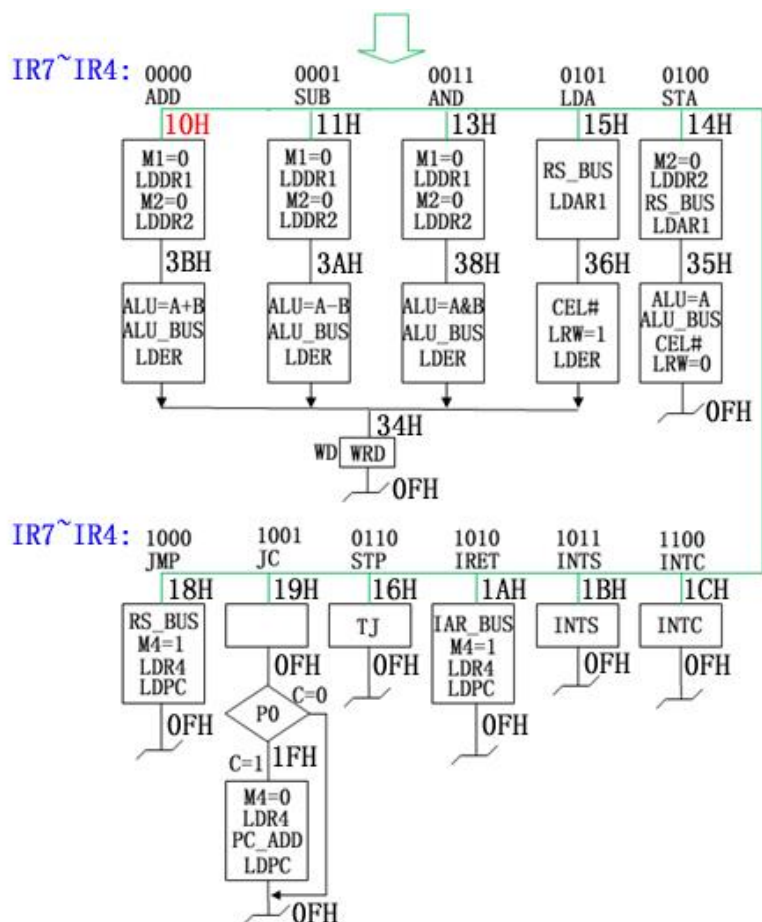
微程序流程图（一）

应当着重指出，在微指令格式的设计过程中，对数据通路所需的控制信号进行了归并和化简。细心的同学可能已经发现，微程序控制器输出的控制信号远远少于数据通路所需的控制信号。这里提供的微程序流程图，是没有经过归并和化简的。仔细研究一下微程序流程图，就会发现有些信号的出现的位置完全一样，这样的信号用其中一个信号就可以代表。请看信号 LDPC 和 LDR4，这两个信号都在微程序地址 07H、1AH、1FH、26H 出现，而在其他的微程序地址都不出现，因此这两个信号产生的逻辑条件是完全一样的。从逻辑意义上看，这

两个信号的作用是产生新的 PC，完全出现在相同的微指令中是很正常的，因此用 LDPC 完全可以代替 LDR4。还有另一些信号，例如 LDDR1 和 LDDR2，出现的位置基本相同。LDDR2 和 LDDR1 的唯一不同是在地址 14H 的微指令中，出现了 LDDR2 信号，但是没有出现 LDDR1 信号。LDDR1 和 LDDR2 是否也可以归并成一个信号呢？答案是肯定的。微程序流程图中只是指出了在微指令中必须出现的信号，并没有指出出现其他信号行不行，这就要根据具体情况具体分析。在地址 14H 的微指令中，出现 LDDR1 信号行不行呢？完全可以。在地址 14H 出现的 LDDR1 是一个**无用的**信号，同时也是一个**无害的**信号，它的出现完全没有副作用，因此 LDDR1 和 LDDR2 可以归并为一个信号 LDDR1。根据以上两条原则，我们对下列信号进行了归并和化简：

LDIR(CER)	为 1 时,允许对 IR 加载,此信号也可用于作为双端口存储器右端口选择 CER。
LDPC(LDR4)	为 1 时,允许对程序计数器 PC 加载,此信号也可用于作为 R4 的加载允许信号 LDR4。
LDAR1(LDAR2)	为 1 时,允许对地址寄存器 AR1 加载,此信号也可用于作为对地址寄存器 AR2 加载。
LDDR1(LDDR2)	为 1 时允许对操作数寄存器 DR1 加载。此信号也可用于作为对操作数寄存器 DR2 加载。
M1 (M2)	当 M1 = 1 时, 操作数寄存器 DR1 从数据总线 DBUS 接收数据；当 M1 = 0 时, 操作数寄存器 DR1 从寄存器堆 RF 接收数据。此信号也可用于作为操作数寄存器 DR2 的数据来源选择信号。

在对微指令格式进行归并和化简的过程中，我们有意保留了一些信号，没有化简，同学们可以充分发挥创造性，提出更为简单的微指令格式。



微程序流程图(二)

三、实验设备

TEC-9 计算机组成原理实验系统一台

四、实验任务及步骤

1、实验任务

用微程序控制方式执行下表的程序，并记录实验现象。

地址	指令	机器代码
00H	LDA R0, [R2]	58H
01H	LDA R1, [R3]	5DH
02H	ADD R0, R1	04H
03H	JC +5	95H
04H	AND R2, R3	3EH
05H	SUB R3, R2	1BH
06H	STA R3, [R2]	4BH
07H	STP	60H
08H	JMP [R1]	84H

2、实验步骤

(1) 按下表接线

信号 1	LDIR	LDPC	LDDR1	M1	LDAR1	IR0	IR0	IR1	IR1	IR2	IR3
信号 2	CER	LDR4	LDDR2	M2	LDAR2	RD0	WR0	RD1	WR1	RS0	RS1

信号 1 接相应的信号 2。黄色高亮信号的插孔，在微程序控制器下方；绿色高亮信号的插孔在寄存器、运算器、存储器下方。

选择开关拨至“微程序”，使实验系统处于微程序控制状态。

(2) 写寄存器，将 60H 写入 R2，将 61H 写入 R3

• DP = 0, DB = 0, DZ = 0。

• 置 SWC = 0、SWB = 1、SWA = 1，使实验系统处于写寄存器方式。按 CLR#按钮，使实验系统处于初始状态。

• 将 SW7—SW0 置为 FFH,用于写入 AR1。该存储器地址供设置通用寄存器使用（可以是任意值，但最好是不常用的一个地址，以免设置通用寄存器操作破坏重要的存储器单元的内容，本实验设置为 FFH）。

• 按一次 QD 按钮，将 0FFH 写入 AR1。

• 将 SW7—SW0 置为 02H，作为通用寄存器 R2 的寄存器号。按一次 QD 按钮，则将 02H 写入 IR，由 IR1-IR0 指定写入的寄存器号。

• 将 SW7—SW0 置为 60H，作为写入 R2 的值。按一次 QD 按钮，将 60H 写入 R2 寄存器。

• 将 SW7—SW0 置为 03H，选择通用寄存器 R3。按一次 QD 按钮，将 03H 写入 IR。

• 将 SW7—SW0 置为 61H，作为写入 R3 的值。按一次 QD 按钮，将 61H 写入 R3。

• 置 R2、R3 结束，按 CLR#按钮，使实验系统恢复到初始状态。

(3) 将程序写入存储器

从存储单元的 00H 地址开始依次存 10 条指令：58H，5DH，04H，95H，3EH，1BH，4BH，60H，84H。在 60H 单元存入 24H，在 61H 单元存入 83H。具体步骤：

• 令 DP = 0, DB = 0, DZ = 0,令 SWC = 0、SWB = 1、SWA = 0，使实验系统处于写双端口存储器工作方式。按 CLR#按钮，使实验系统处于初始状态。

• 置 SW7—SW0 为 00H，按 QD 按钮，将 00H 写入 AR1。

• 置 SW7—SW0 为 58H，按 QD 按钮，将 58H 写入存储器 00H 单元。AR1 自动加 1，

变为 01H。

- 置 SW7—SW0 为 5DH，按 QD 按钮，将 5DH 写入存储器 01H 单元。AR1 自动加 1，变为 02H。

- 重复进行下去，一直到将 84H 写入存储器 09H 单元。

- 按 CLR#按钮，使实验系统恢复到初始状态。

- 置 SW7—SW0 为 60H，按 QD 按钮，将 60H 写入 AR1。

- 置 SW7—SW0 为 24H，按 QD 按钮，将 24H 写入存储器 60H 单元。AR1 自动加 1，变为 61H。

- 置 SW7—SW0 为 83H，按 QD 按钮，将 83H 写入存储器 61H 单元。按 CLR#按钮，使实验系统恢复到初始状态。

说明：上述任务中的程序机器代码用控制台操作存入内存中，并根据程序的需要，用数码开关 SW7—SW0 设置通用寄存器 R2、R3 及内存相关单元的数据。注意：由于设置通用寄存器时会破坏内存单元的数据，因此一般应先设置寄存器的数据，再设置内存数据。也可以使用上端软件或实验台监控系统用 PS2 键盘写入内容。

(4) 用单拍 (DP) 方式执行一遍程序。观察通用寄存器堆 RF 中四个寄存器的数据，以及由 STA 指令存入 RAM 中的数据 (程序结束后从 RAM 的相应单元中读出)，与理论分析值作对比。单拍方式执行时注意观察微地址指示灯、IRBUS 指示灯、DBUS 指示灯、AR2 指示灯、AR1 指示灯和判断字段指示灯的值，以跟踪程序中取指令和执行指令的详细过程 (可观察到每一条微指令)。(在执行程序之前，确保 PC 初值为 00H，若不为 0，重新操作上述步骤，或重启后重新操作上述步骤)

- 置 SWC=0，SWB=0，SWA==0；DP=1，DZ=0，DB=0。

- 连续按动 QD，观察实验现象。

说明：在单拍执行过程中，首先要随时监测 AR2 的值和 IR 的值，以判定程序执行到何处，正在执行哪条指令。监测微地址指示灯和判断字段指示灯，对照微程序流程图，可以判断出微指令的地址和正在进行的微操作。

程序执行的结果如下：

初始状态：R0 未定，R1 未定，R2 = 60H，R3 = 61H。存储器 60H 单元的内容是 24H，61H 单元的内容是 83H。系统从存储器的 00H 单元开始执行程序。

1) 执行 LDA R0, [R2]；结果 R0 = 24H、R1=XXH、R2=60H、R3=61H

2) 执行 LDA R1, [R3]；结果 R0 = 24H、R1=83H、R2=60H、R3=61H

3) 执行 ADD R0, R1；结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H，C = 0。

4) 执行 JC +5；因为 C = 0，顺序执行。执行结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

5) 执行 AND R2, R3；结果 R0 = 0A7H、R1=83H、R2=60H、R3=61H

6) 执行 SUB R3, R2；结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H

- 7) STA R3, [R2]; 结果 R0 = 0A7H、R1=83H、R2=60H、R3=01H
存储器 60 单元的内容为 01H。
- 8) 执行 STP; 结果: 无变化
- 9) JMP [R1]; 结果: 转移到 83H。(察看 AR2 即 PC 地址)

第七节 中断原理实验

一、实验目的

- (1) 从硬件、软件结合的角度, 模拟单级中断和中断返回的过程;
- (2) 通过简单的中断系统, 掌握中断控制器、中断向量、中断屏蔽等概念;
- (3) 了解微程序控制器与中断控制器协调工作的基本原理。

二、实验电路

仍然使用前面的 CPU 组成与机器指令执行实验的电路图, 但本次实验加入中断系统。这是一个简单的中断系统模型, 只支持单级中断、单个中断请求, 有中断屏蔽功能, 旨在说明最基本的原理。

中断屏蔽控制逻辑分别集成在 2 片 GAL22V10 (TIMER1 和 TIMER2) 中。其 ABEL 语言表达式如下:

```
INTR1 := INTR;  
INTR1.CLK = CLK1;  
IE := CLR & INTS # CLR & IE & !INTC;  
IE.CLK = MF;  
INTQ = IE & INTR1;
```

其中, CLK1 是 TIMER1 产生的时钟信号, 它主要是作为 W1—W4 的时钟脉冲, 这里作为 INTR1 的时钟信号, INTE 的时钟信号是晶振产生的 MF。INTS 微指令位是 INTS 机器指令执行过程中从控制存储器读出的, INTC 微指令位是 INTC 机器指令执行过程中从控制存储器读出的。INTE 是中断允许标志, 控制台有一个指示灯 IE 显示其状态, 它为 1 时, 允许中断, 为 0 时, 禁止中断。当 INTS = 1 时, 在下一个 MF 的上升沿 IE 变 1, 当 INTC = 1 时, 在下一个 MF 的上升沿 IE 变 0。CLR 信号实际是控制台产生的复位信号 CLR#。当 CLR = 0 时, 在下一个 CLK1 的上升沿 IE 变 0。当 CLR=1 且 INTS = 0 且 INTC = 0 时, IE 保持不变。

INTR 是外部中断源, 接控制台按钮 INTR。按一次 INTR 按钮, 产生一个中断请求正脉冲 INTR。INTR1 是 INTR 经时钟 CLK1 同步后产生的, 目的是保持 INTR1 与实验台的时序信号同步。INTR 脉冲信号的上升沿代表有外部中断请求到达中断控制器。INTQ 是中断屏蔽控制逻辑传递给 CPU 的中断信号, 接到微程序控制器上。当收到 INTR 脉冲信号时, 若

中断允许位 $INTE = 0$ ，则中断被屏蔽， $INTQ$ 仍然为 0；若 $INTE = 1$ ，则 $INTQ = 1$ 。

为保存中断的断点地址，以便中断返回，设置了一个中断地址寄存器 IAR 。第二节图 4 中的 IAR (U19) 就是这个中断地址寄存器，它是一片 74HC374，有 $LDIAR$ 和 $IAR_BUS\#$ 两个信号输入端，均连接至微程序控制器。 $LDIAR$ 信号的上升沿到达时，来自程序计数器 PC 的地址会置入 IAR 中。 $IAR_BUS\#$ 为 0 时，保存在 IAR 中的断点地址会输出到数据总线 $DBUS$ 上。由于本实验系统只有一个断点寄存器而无堆栈，因此仅支持一级中断而不支持多级中断。

中断向量即中断服务程序的入口地址，在本实验仪中由 8 位数码开关 $SW7—SW0$ 提供。

三、中断的检测、执行和返回过程

微程序控制器每执行一条机器指令之后，执行下一条机器指令之前，先转到微地址 0F 处(见图 12 微程序流程图)，在条件位 $P1=1$ 时判断是否有中断请求 $INTQ$ 。如果没有 $INTQ$ ，则继续正常的机器指令执行。若检测到中断请求 $INTQ$ ，首先发出关中断信号 $INTC$ 、保存断点信号 $LDIAR$ ，并且发出停机信号 TJ ，等待手动设置中断向量。设置好 $SW7—SW0$ 后，按 QD 按钮启动，机器将中断向量读入程序计数器 PC 中，从而转到中断服务子程序去执行。

执行一条机器指令 $IRET$ ，从中断服务子程序返回时，发出 $IAR_BUS\#$ 信号，从中断地址寄存器 IAR 向数据总线 $DBUS$ 输出断点地址，再从 $DBUS$ 依次写入到 $R4$ 、 PC 中，恢复执行被中断的程序。

发生中断时，关中断由硬件负责，而中断现场(寄存器堆中的寄存器，进位标志 C)的保存和恢复由中断服务程序来处理。

四、实验设备

- (1) TEC-9 计算机组成原理实验系统一台

五、实验任务

- (1) 了解中断系统中每个信号的意义和变化条件，并将下面的主程序和中断服务程序手工汇编成十六进制机器代码，此项任务应在预习时完成。

主程序：

地址	指令	机器代码
20H	INTS	
21H	LDA R0,[R2]	
22H	ADD R0,R0	
23H	ADD R0,R0	
24H	ADD R0,R0	
25H	ADD R0,R0	
26H	ADD R0,R0	

27H	ADD R0,R0	
28H	ADD R0,R0	
29H	JMP [R1]	

中断服务程序：

地址	指令	机器代码
0A0H	AND R0,R0	
0A1H	IRET	

(2)参考 CPU 组成与机器指令执行实验,再加上中断系统,完成本次实验的线路连接。
接通电源之前应仔细检查接线,确认无误。

(3)将上述任务(1)的程序代码存入内存中,并根据需要设置通用寄存器组和内存相关单元的数据。其中,寄存器 R1 的值应置为 21H,以便程序循环执行。

(4)从地址 20H 执行程序,在程序运行中,按一次控制台的 INTR。进入中断后,用单拍(DP)方式执行,直到返回主程序为止。列表记录中断系统中有关信号的变化情况,特别要纪录好断点地址和 R0 的值。

(5)重复执行(4)两次。(一共执行 3 次)

(6)将 RAM 中 20H 单元的内容由指令 INTS 改为 INTC,重作(4),记录发生的现象。

六、实验步骤和实验结果

1、实验程序

主程序：

地址	指令	机器代码
20H	INTS	0B0H
21H	LDA R0,[R2]	58H
22H	ADD R0,R0	00H
23H	ADD R0,R0	00H
24H	ADD R0,R0	00H
25H	ADD R0,R0	00H
26H	ADD R0,R0	00H
27H	ADD R0,R0	00H
28H	ADD R0,R0	00H
29H	JMP [R1]	84H

中断服务程序：

地址	指令	机器代码
----	----	------

0A0H	AND R0,R0	30H
0A1H	IRET	0A0H

2、接线

微程序控制器与数据通路之间的线可以通过选择开关直接选择。将开关设置为“微程序”。只需连接数据通路部分的线。

信号 1	LDI R	LDP C	LDDR 1	M 1	LDAR 1	IR0	IR0	IR1	IR1	IR2	IR3	DM C
信号 2	CER	LDR 4	LDDR 2	M 2	LDAR 2	RD 0	WR 0	RD 1	WR 1	RS 0	RS 1	INT R

灰色部分连接和实验五一样，高亮部分信号在 QD 脉冲开关上方。

将选择开关拨至“微程序”。

3、存程序机器代码，设置通用寄存器 R1、R2 及内存相关单元的数据。

(1)、设置寄存器 R1、R2 的值

根据要求，设置 R1 = 21H，R2 的值由实验者自定，假定为 10H。

1) 令 DP=0，DB=0，DZ=0，使实验系统处于连续运行状态。令 SWB=1，SWA=1，使实验系统处于寄存器加载工作方式 KLD。按 CLR#按钮，使实验系统处于初始状态。

2) 在 SW7—SW0 上设置一个存储器地址，该存储器地址供设置通用寄存器使用。该存储器地址最好是不常用的一个地址，以免设置通用寄存器操作破坏重要的存储器单元的内容。例如可将该地址设置为 0FFH。按一次 QD 按钮，将 0FFH 写入 AR1 和 AR2。

3) SW7—SW0 设置为 01H，作为通用寄存器 R1 的寄存器号。按一次 QD 按钮，将 01H 写入 IR。

4) SW7—SW0 设置为 21H，按一次 QD 按钮，将 21H 写入 IR 指定的 R1 寄存器。

5) SW7—SW0 设置为 02H，作为通用寄存器 R2 的寄存器号。按一次 QD 按钮，将 02H 写入 IR。

6) SW7—SW0 设置为 10H，作为 R2 的值。按一次 QD 按钮，将 10H 写入 IR 指定的 R2 寄存器。

7) 设置 R1、R2 结束，按 CLR#按钮，使实验系统恢复到初始状态。

(2)、存程序机器代码。

本操作中，在 10H 单元存入 01H(也可以是其他值)，从 20 地址开始存 10 个机器代码：0B0H, 58H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 84H。从 0A0H 存入 2 个机器代码:30H,0A0H。

1) 令 DP=0, DB=0, DZ=0, 使实验系统处于连续运行状态。令 SWB=1, SWA=0, 使实验系统处于写双端口存储器工作方式 KWRD。按 CLR#按钮, 使实验系统处于初始状态。

2) 置 SW7—SW0 为 10H, 按 QD 按钮, 将 10H 写入 AR1。

3) 置 SW7—SW0 为 01H, 按 QD 按钮, 将 01H 写入存储器 10H 单元。

写主程序

1) 按 CLR#按钮, 使实验系统恢复初始状态。

2) 置 SW7—SW0 为 20H, 按 QD 按钮, 将 20H 写入 AR1。

3) 置 SW7—SW0 为 0B0H, 按 QD 按钮, 将 0B0H 写入存储器 20H 单元。AR1 自动加 1, 变为 21H。

4) 置 SW7—SW0 为 58H, 按 QD 按钮, 将 58H 写入存储器 21H 单元。AR1 自动加 1, 变为 22H。

5) 重复进行下去, 一直到将 84H 写入存储器 29H 单元。按 CLR#按钮, 使实验系统恢复到初始状态。

写中断程序

1) 置 SW7—SW0 为 0A0H, 按 QD 按钮, 将 0A0H 写入 AR1。

2) 置 SW7—SW0 为 30H, 按 QD 按钮, 将 30H 写入存储器 0A0H 单元。

3) 置 SW7—SW0 0A0H, 按 QD 按钮, 将 0A0H 写入存储器 0A1H 单元。

4) 按 CLR#按钮, 使实验系统恢复到初始状态。

(3) 从地址 20H 连续执行程序

从地址 20H 连续执行程序, 在程序运行中, 按一次控制台的 INTR。进入中断后, 用单拍 (DP) 方式执行, 直到返回主程序为止。列表记录中断系统中有关信号的变化情况, 特别要纪录好断点地址和 R0 的值。

1) 令 DP=0, DZ=0, DB=0, 使实验系统处于连续运行状态。

2) 置 SW7—SW0 为 20H, 作为程序的起始地址。按 QD 按钮, 启动程序从 20H 地址运行。中断允许指示灯亮。

3) 按 INTR 按钮, 发出一个 INTR 中断脉冲, 请求中断。中断后硬件自动将中断地址存入中断地址寄存器 IAR。微程序地址应为 25H。

4) 置 SW7—SW0 为 0A0H, 这是中断程序的入口地址。将 DP 由置 0 改为置 1。按一次 QD 按钮, 将 0A0H 送入程序计数器 PC。微程序地址应为 26H。

5) 按一次 QD 按钮, 进行取指微操作。微程序地址应为 05H。

6) 按一次 QD 按钮, 进行置数微操作。微程序地址应为 13H。

7) 按一次 QD 按钮, 进行 R0 & R0 操作, 这时微程序地址应为 38H。观察 DBUS 总线的值, 即为 R0 的值。

8) 按一次 QD 按钮, 进行写回微操作。微程序地址应为 34H。

9) 按一次 QD 按钮, 进行判定有无中断请求 INTQ 微操作。微程序地址应为 0FH。

- 10) 按一次 QD 按钮, 进行取指微操作。微程序地址应为 05H。
- 11) 按一次 QD 按钮, 进行从中断程序返回主程序微操作。微程序地址应为 1AH。这时观察 PC 地址, 即中断地址。
- 12) 按一次 QD 按钮, 进行判有无新的中断操作。微程序地址为 0FH。
- 13) 按一次 QD 按钮, 进行取指微操作。微程序地址为 05H。此刻, 程序已返回主程序。

由于按下 INTR 按钮的时间对主程序而言, 是随机的, 具有不确定性, 因此各次中断地址、R0 会具有不同的值。

将 RAM 中 20H 单元的内容由指令 INTS 改为 INTC, 重作 (4), 记录发生的现象。

由于 INTC 是关中断指令, 因此将不会发生中断。

扩展实验提示

实验二:

结合实验一设计实验, 写出将 01100011B 写入 R1, 将 01100011B 写入 R3, 再利用 ALU 完成 R1 加 R3 和 R1 减 R3。写出实验步骤, 并记录结果。

修改接线, 将 M1、M2 接“0”, 设置 DR1 和 DR2 的输入端是寄存器堆的 B 端和 A 端, 然后设置 LDDR1 和 LDDR2 为“1”, 按动 QD, 完成写入。先利用实验 1 的方法, 在 R1 和 R3 中写入指定数据, 然后将 R1 和 R3 的数据分别通过 B 端和 A 端写入 DR1 和 DR2, 再设置 S3-S0, M, CN 的状态, 完成运算。

实验四:

1、将存储器 11H 单元中的数写入到 R1 中。设计实验步骤, 并记录实验结果。

将 AR1 设置为 11H, 设置 LRW 和 CEL 开关, 让存储单元的数据从左端口输出到 DBUS, 然后将 DBUS 上的数据写入 R1 寄存器。

2、设计实验, 在 R0 中写入 55H, 存储单元 11H 中写入 AAH, 完成“R0 加[11H]”的计算。设计实验步骤, 并记录实验结果。(该实验已经将 S3-S0、M 和 CN 置为加的状态;)

要实现寄存器和存储单元内容的相加, 首先应将 R0 中的数据通过 A 端口或 B 端口写入 DR2 或 DR1 中(参考实验二的扩展实验), 然后将存储器 11H 单元中的数据通过 DBUS 写入另一个 DR 中(参考实验四的第一个扩展实验), 再执行加操作。