



燕山大学

计算机组成原理实验报告

Principles of Computer Organization Experiment Report

学 院：信息科学与工程学院（软件学院）

班 级：18 软件 6 班

姓 名：乔翱

学 号：201811040809

指导教师：任大伟 李季辉 陈贺敏 余扬

教 务 处

2020 年 5 月

目 录

实验 1 运算器实验	4
1.1 实验目的	4
1.2 实验要求	4
1.3 实验原理	4
1.4 实验内容与步骤	5
1.5 实验结果	7
1.6 思考与分析	7
实验 2 存储器实验	9
2.1 实验目的	9
2.2 实验要求	9
2.3 实验原理	9
2.3 实验内容与步骤	10
2.5 实验结果	12
2.6 思考与分析	13
实验 3 总线与微命令实验	14
3.1 实验目的	14
3.2 实验要求	14
3.3 实验原理	14
3.4 实验内容与步骤	16
3.5 实验结果	20
3.6 思考与分析	21
实验 4 微程序控制器实验	23
4.1 实验目的	23
4.2 实验要求	23
4.3 实验原理	23
4.4 实验内容与步骤	26
4.5 实验结果	28
4.6 思考与分析	28
实验 5 简单模型机实验	29
5.1 实验目的	30
5.2 实验要求	30
5.3 实验原理	30
5.4 实验内容与步骤	33
5.5 实验结果	36
5.6 思考与分析	37

实验 1 运算器实验

1.1 实验目的

- (1) 掌握算术逻辑运算单元的工作原理。
- (2) 熟悉简单运算器的电路组成。
- (3) 熟悉 4 位运算功能发生器 (74LS181) 的算术、逻辑运算功能。

1.2 实验要求

- (1) 做好实验预习，看懂电路图，熟悉实验中所用芯片各引脚的功能和连接方法。
- (2) 按照实验内容与步骤的要求，认真仔细地完成实验。
- (3) 写出实验报告。

1.3 实验原理

运算器实验电路如图 1.1 所示。两片 4 位的 74LS181 构成 8 位字长的 ALU。其中 74LS181(1) 做低 4 位算术逻辑运算，74LS181(2) 做高 4 位算术逻辑运算，74LS181(1) 的进位输出信号 C_{n+4} 与 74LS181(2) 的进位输入信号 C_n 相连，两片 74LS181 的控制信号 $S_3 \sim S_0$ 、 M 分别相连。74LS181(2) 的进位输出信号 C_{n+4} 可另接一个指示灯，用于显示运算器进位标志信号状态。两个 8 位的 74LS273 作为工作寄存器 DR1 和 DR2，用于暂存参与运算的操作数。参与运算的数据由数据开关通过三态门 74LS245 送入工作寄存器，ALU 的运算结果也通过三态门 74LS245 发送到数据显示灯上。参与运算的操作数由 $SW_7 \sim SW_0$ 共 8 个二进制开关来设置，当 $\overline{SW - BUS} = 0$ 时，数据通过三态门 74LS245 输出到 DR1 和 DR2。DR1 接 ALU 的 A 输入端口，DR2 接 ALU 的 B 输入端口。在 P1 的上升沿将数据打入 DR1，送至 74LS181 的 A 输入端口；在 P2 的上升沿将数据打入 DR2，送至 74LS181 的 B 输入端口。

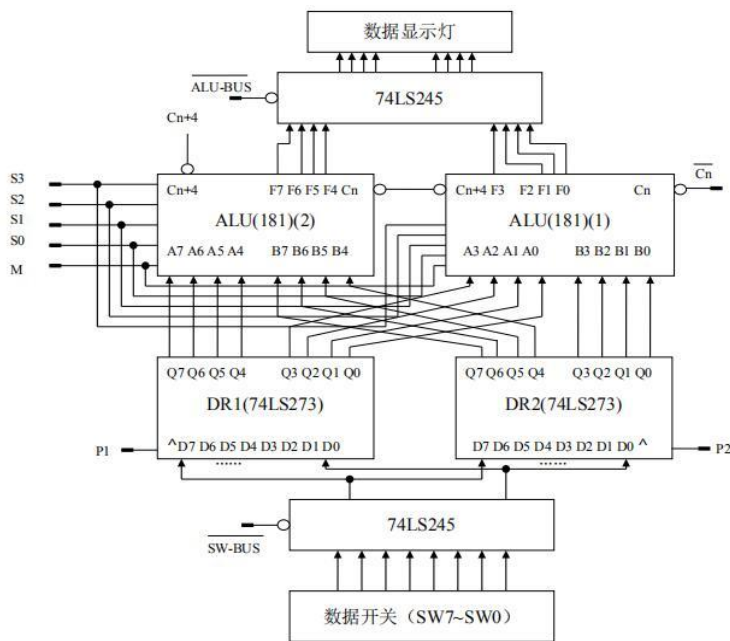


图 1.1 运算器实验电路

1.4 实验内容与步骤

1. 运行虚拟实验系统，按照图 1.1 绘制运算器实验电路，生成实验电路如图 1.2 所示：

示：

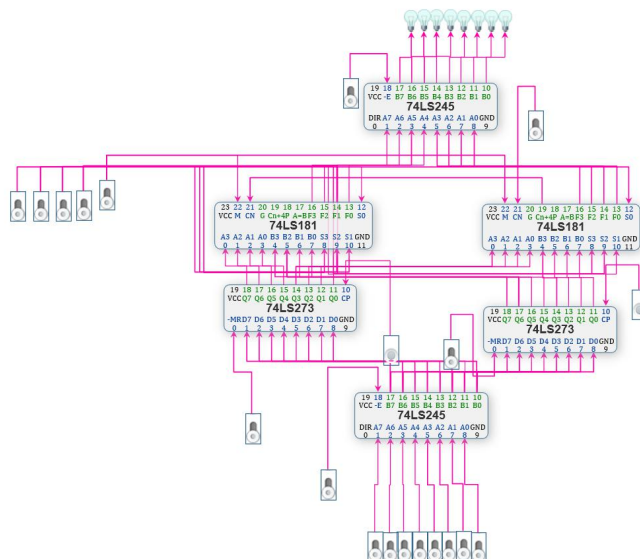


图 1.2 运算器虚拟实验电路

2. 进行电路预设置，具体步骤如下：

- (1) 将 $\overline{\text{ALU-BUS}}$ 设为高电平，关闭 ALU 输出端的三态门；
- (2) 将两片 74LS273 的 $\overline{\text{MR}}$ 都设为高电平，否则 74LS273 会一直处于清零状态。

3. 打开电源开关。

4. 设置 SW7~SW0 向 DR1 和 DR2 置数。以 DR1=65H, DR2=A7H 为例，具体步骤如下：

- (1) 将 $\overline{\text{SW-BUS}}$ 置 0，打开数据输入端的三态门；
- (2) 将数据开关的 SW7~SW0 置为 01100101；
- (3) 发出 P1 单脉冲信号，在 P1 的上升沿，数据打入寄存器 DR1；
- (4) 将数据开关的 SW7~SW0 置为 10100111；
- (5) 发出 P2 单脉冲信号，在 P2 的上升沿，数据打入寄存器 DR2。
- (6) 将 $\overline{\text{SW-BUS}}$ 置 1，关闭数据输入端的三态门；

5. 检验 DR1 和 DR2 中存的数是否正确。其具体操作如下：

- (1) $\overline{\text{ALU-BUS}}=0$ ，打开 ALU 输出端的三态门；
- (2) 设置 Cn=1，ALU 无进位输入；
- (3) 将 S3、S2、S1、S0、M 置为 00000，指示灯应显示 DR1 中数据 01100101；
- (4) 将 S3、S2、S1、S0、M 置为 10101，指示灯应显示 DR2 中数据 10100111。

6. 验证 74LS181 的算术运算和逻辑运算功能(采用正逻辑)。在给定 DR1=65H, DR2=A7H 的情况下，改变运算器的功能模式，观察运算器的输出，并填入表 1-1，并和理论值比较、验证。

1.5 实验结果

表 1-1 运算器功能验证

工作模式选择 S3 S2 S1 S0	算术运算（M=0）（Cn=1 无进位）		逻辑运算（M=1）	
	功能	输出值（16 进制）	功能	输出值（16 进制）
0000	A	65H	\overline{A}	9AH
0001	A+B	E7H	$\overline{A+B}$	18H
0010	$A+\overline{B}$	7DH	\overline{AB}	82H
0011	0 minus 1	FFH	Logical 0	0H
0100	A plus \overline{AB}	A5H	\overline{AB}	DAH
0101	(A+B) plus \overline{AB}	27H	\overline{B}	58H
0110	A minus B minus 1	BDH	$A\oplus B$	C2H
0111	\overline{AB} minus 1	BDH	\overline{AB}	40H
1000	A plus AB	8AH	$\overline{A+B}$	BFH
1001	A plus B	CH	$\overline{A\oplus B}$	3DH
1010	($A+\overline{B}$) plus AB	A2H	B	A7H
1011	AB minus 1	24H	AB	25H
1100	A plus A	CAH	Logical 1	FFH
1101	(A+B) plus A	4CH	$A+\overline{B}$	7D
1110	($A+\overline{B}$) plus A	E2H	A+B	E7H
1111	A minus 1	64H	A	65H

注意：A 和 B 分别表示参与运算的两个数，“+”表示逻辑或，“plus”表示算术求和

1.6 思考与分析

1. 运算器主要由哪些器件组成？这些器件是怎样连接的？

答：运算器由算术逻辑单元（ALU）、累加器、状态寄存器、通用寄存器组等构成，在本实验中的简化电路中两片 4 位的 74LS181 构成 8 位字长的 ALU,两个 8 位的 74LS273 作为工作寄存器 DR1 和 DR2，用于暂存参与运算的操作数。参与运算的数据由数据开关通过三态门 74LS245 送入工作寄存器，ALU 的运算结果也通过三态门 74LS245 发送到数据显示灯上。在运算器内，ALU 的输入端会和寄存器连接，接受寄存器中暂存的数据，

而 ALU 的输出端也会和寄存器连接，ALU 的计算结果可以存到寄存器内部。

2. 芯片 74LS181 没有减法：A minus B 的指令，怎样实现减法功能？

答：对于减法操作，计算机中也是通过加法来实现的，所有的减法实际上都是通过对减数补码的加法运算实现的。减去一个数等于加上这个数的补码。负数的补码等于负数的反码加 1。通过反码的方式可以将减法转换成加法运算，而对于减法的结果还需要进行补码才能得到最终的结果。

对于 74LS181 芯片只需设置 $S_3S_2S_1S_0=0110$ ，并且 M 置 0，Cn=0 就可以实现两个数相减的操作。

实验 2 存储器实验

2.1 实验目的

- (1) 掌握算术逻辑运算单元的工作原理。
- (2) 熟悉简单运算器的电路组成。
- (3) 熟悉 4 位运算功能发生器 (74LS181) 的算术、逻辑运算功能。

2.2 实验要求

- (1) 做好实验预习, 看懂电路图, 熟悉实验中所用芯片各引脚的功能和连接方法。
- (2) 按照实验内容与步骤的要求, 认真仔细地完成实验。
- (3) 写出实验报告。

2.3 实验原理

实验所用的半导体静态存储器电路如图 2.1 所示。数据开关(SW7-SW0)用于设置读写地址和欲写入存储器的数据, 经三态门 74LS245 与总线相连, 通过总线把地址发送至 AR, 或把欲写入的数据发送至存储器芯片。静态存储器由一片 6116(2K×8)构成, 但地址输入引脚 A8~A10 接地, 因此实际存储容量为 256 字节, 其余地址引脚 A0~A7 与 AR 相连, 读和写的地址均由 AR 给出。6116 的数据引脚为输入、输出双向引脚, 与总线相连, 既可从总线输入欲写的数据, 也可以通过总线输出数据到数据灯显示。共使用了两组显示灯, 一组显示从存储器读出的数据, 另一组显示存储单元的地址。

6116 有三根控制线, CE 为片选线, OE 为读线, WE 为写线, 三者的有效电平均为低电平。当片选信号有效时, OE=0 时进行读操作, WE=0 进行写操作, 本实验将 OE 接地, 在此情况下, 当 CE=0、WE=1 时进行读操作; 当 CE=0、WE=0 时进行写操作。由于 6116 的 WE 信号是由 WE 控制信号与 P1 进行与非运算得来的, 因此, WE=1 时为写操作, 其写时间与 P1 脉冲宽度一致。

读数据时, 在数据开关上设置好要读的存储单元地址, 并打开三态门 74LS245, LDAR 置 1, 发出一个 P2 脉冲, 将地址送入 6116, 设置 6116 为读操作, 即可读出数据并

在数据灯上显示。

写数据时，先在数据开关上设置好要写的存储单元地址，并打开三态门 74LS245，LDAR 置 1，发出一个 P2 脉冲，将地址送入 6116，然后在数据开关上设置好要写的数据，确保三态门打开，设置 6116 为写操作，发出一个 P1 脉冲，即可将数据写入。

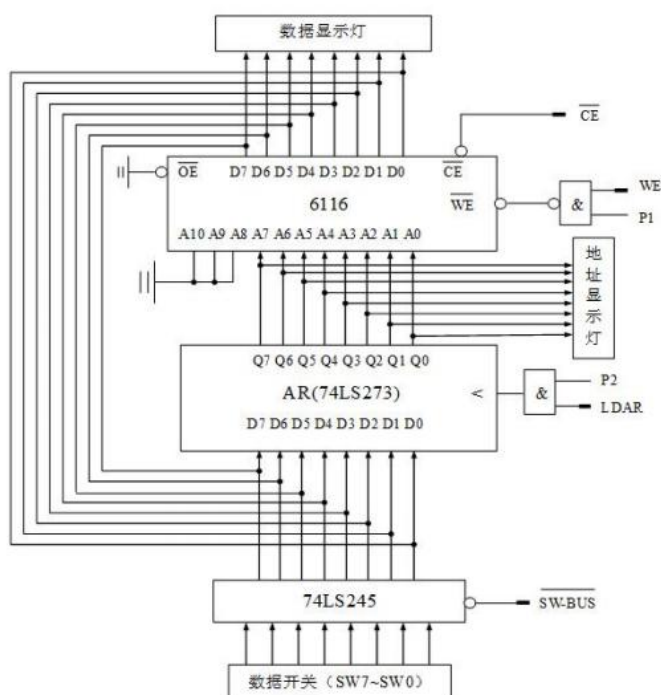


图 2.1 随机存储器实验电路

2.3 实验内容与步骤

1. 运行虚拟实验系统，从左边的实验设备列表选取所需组件拖到工作区中，按照图 2.1 所示组建实验电路，得到如图 2.2 所示的实验电路。注意：图 2.2 中没有使用总线，元器件通过两两之间连线实现彼此连接。当然，实验时也可以选用总线来连接器件。

2. 进行电路预设置，具体步骤如下：

- (1) 将 74LS273 的 MR 置 1，AR 不清零；
- (2) $\overline{CE} = 1$ ，RAM6116 未片选；
- (3) $\overline{SW} - \overline{BUS} = 1$ ，三态门关闭。

3. 打开电源开关。

4. 存储器写操作。向 01H、02H、03H、04H、05H 存储单元分别写入十六进制数据 11H、12H、13H、14H、15H，具体操作步骤如下(以向 01 号单元写入 11H 为例)：

(1) 将 SW7~SW0 置为 00000001SW - BUS =0，打开三态门，将地址送入 BUS；

(2) LDAR=1，发出 P2 单脉冲信号，在 P2 的上升沿将 BUS 上的地址存入 AR，可通过观察 AR 所连接的地址灯来查看地址，SW - BUS =1 关闭三态门；

(3) CE =0，WE=1，6116 写操作准备（注意：此时 WE =1，因而会读出此地址原有数据）；

(4) 将 SW7~SW0 置为 00010001， SW - BUS=0，打开三态门，将数据送入 BUS；

(5) 发出 P1 单脉冲信号，在 P1 的上升沿将 BUS 上的数据 00010001 写入 RAM 的 01 地址；

(6) CE =1，6116 暂停工作， SW - BUS =1 关闭三态门。

(7) 按上述步骤完成 02H、03H、04H、05H 存储单元相应数据的存储。

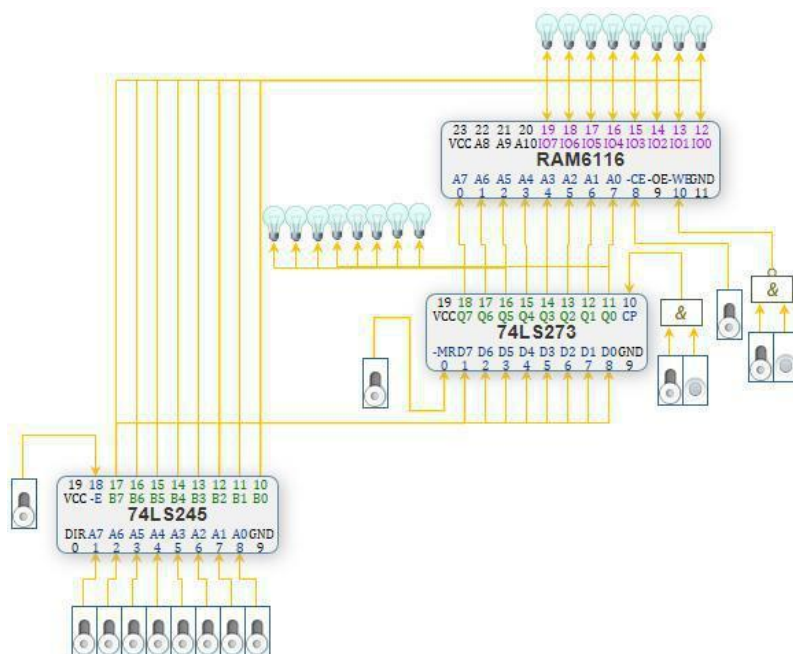


图 2.2 存储器虚拟实验电路

5. 存储器读操作。依次读出 01H、02H、03H、04H、05H 单元中的内容，观察上述单元中的内容是否与前面写入的一致。具体操作步骤如下（以从 01 号单元读出数据 11H 为例）：

- （1）将 SW7~SW0 置为 00000001，SW - BUS =0，打开三态门，将地址送入 BUS；
- （2）LDAR=1，发出 P2 单脉冲信号，在 P2 的上升沿将 BUS 上的地址存入 AR 中，可通过观察 AR 所连接的地址灯来查看地址，SW - BUS =1，关闭三态门；
- （3）CE =0，WE=0，6116 进行读操作，观察数据灯是否为先前写入的 00010001；
- （4）CE =1，6116 暂停工作。

2.5 实验结果

本实验需要记录的结果如下：

- （1）从 05 号单元读出数据 15H 时的实验电路截图：

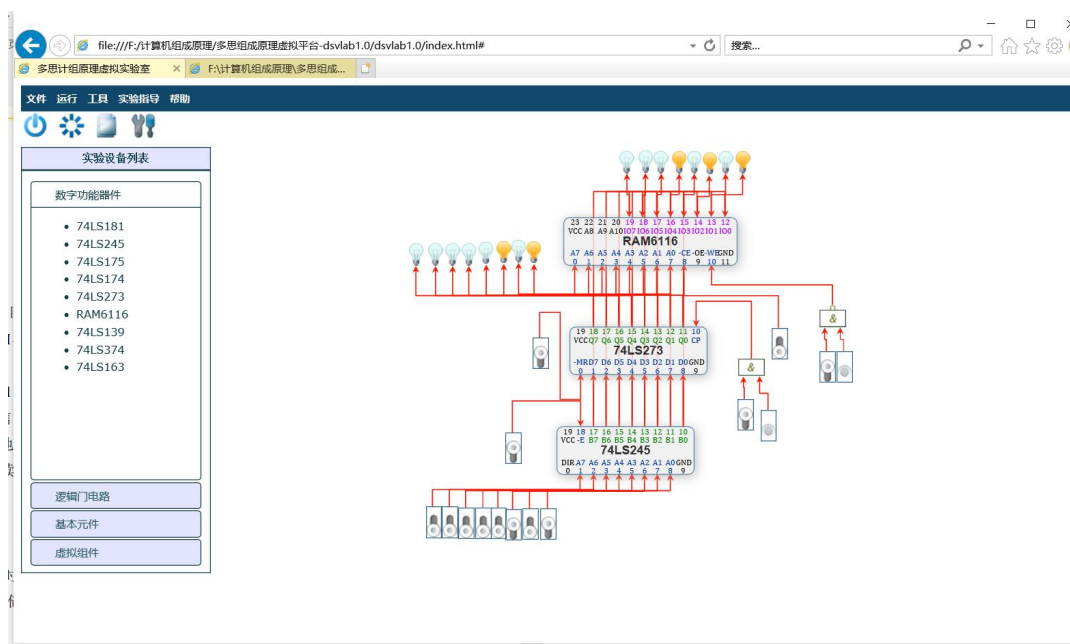


图 2.3 从 05 号单元读出数据 15H 时的实验电路截图

(2) 通过“工具”菜单中的“存储器芯片设置”查看存储器芯片中数据时的界面截图。

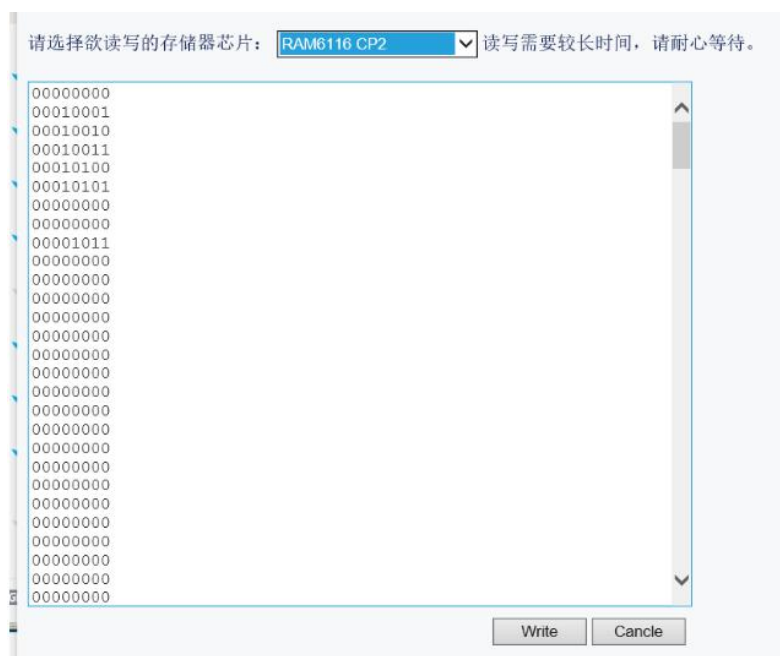


图 2.4 存储器芯片中数据截图

2.6 思考与分析

1. 静态半导体存储器与动态半导体存储器的主要区别是什么？

答：静态半导体存储器（SRAM）速度非常快，只要电源存在内容就不会自动消失。集成度相对较低，功耗也较大。一般高速缓冲存储器用它组成。

动态 RAM（DRAM）的内容在一定时间之后自动消失，因此必须周期性的在内容消失之前进行刷新。集成度高，成本较低，另外耗电也少，但它需要一个额外的刷新电路。动态半导体存储器运行速度较慢，PC 机的标准存储器都采用动态半导体存储器组成。

3. 由两片 6116（2K*8）怎样扩展成（2K*16）或（4K*8）的存储器？怎样连线？

答：两片 6116（2K*8）扩展成（2K*16）需要位扩展，也就是字长的扩展，地址线全部连接在一起，片选及控制信号全部连接在一起，片 0 对应数据线 D0-D7，片 1 对应 D8-D15。

两片 6116（2K*8）扩展成（4K*8）需要字扩展。

实验 3 总线与微命令实验

3.1 实验目的

- (1) 理解总线的概念和作用。
- (2) 连接运算器与存储器，熟悉计算机的数据通路。
- (3) 理解微命令与微操作的概念。

3.2 实验要求

- (1) 做好实验预习，在实验之前填好表 3-3，读懂实验电路图，熟悉实验元器件的功能特性和使用方法。
- (2) 按照实验内容与步骤的要求进行实验，对预习时填写好的微命令进行验证与调试，遇到问题请冷静、独立思考，认真仔细地完成实验。
- (3) 写出实验报告。

3.3 实验原理

本实验使用的主要元器件有：4 位算术逻辑运算单元 74LS181，8 位数据锁存器 74LS273，三态输出的总线收发器 74LS245，2K×8 静态随机存储器 6116，时序发生器，与非门、与门、开关、指示灯等。

时序发生器用于产生四个等间隔时序信号 T1、T2、T3 和 T4。在本虚拟实验系统中，连续发出的一轮 T1~T4 时序信号对应一个 CPU 周期。图 3.1 为时序发生器的简单电路连接图，其中，Ts 为时钟源输入信号，Stop 为停止信号，Star 为开始信号，Step 为单步运行信号。在 Step=0 时，单击 Start 连接的单脉冲按钮，时序信号 T1~T4 会周而复始地发送出去，时序发生器处于连续运行状态，若此时单击 Stop 按钮，发送完此周期时序信号后就会停机。在 Step=1 时，处于单步运行状态，即每发送完一个 CPU 周期时序信号就自动停机。本实验使用单步运行方式。

图 3.2 为本实验数据通路总框图，其中 ALU 由 2 片 74LS181 构成，DR1、DR2 和 AR 均为一片 74LS273，RAM 为一片 6116 芯片，△表示三态门 74LS245，时序发生器

为虚拟实验系统提供的虚拟组件。数据开关、数据显示灯、运算器、存储器通过总线相连。数据开关(SW7-SW0)用于设置数据或地址，数据和地址经三态门发送至总线。DR1和DR2从总线上接收数据并传送到ALU进行运算，运算结果经三态门送回至总线。地址寄存器AR从总线上获取地址并送至存储器，存储器按地址进行读写，将读出的数据发送至总线，或者从总线输入数据并写入。数据显示灯与总线相连，流经总线的所有数据和地址都将在数据灯上显示。

计算机控制器通过控制线向执行部件发出各种控制命令，这些控制命令被称为微命令，执行部件接收微命令后所进行的操作，叫作微操作。图 3.2 中的控制信号线都与控制器相连，并由控制器的相应微命令控制，例如当控制器中表示 $\overline{\text{SW-BUS}}$ 的微命令位设置为0时，低电平信号通过控制线传送到数据开关的三态门，三态门即执行“打开”微操作。

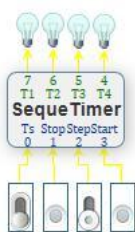


图 3.1 时序发生器

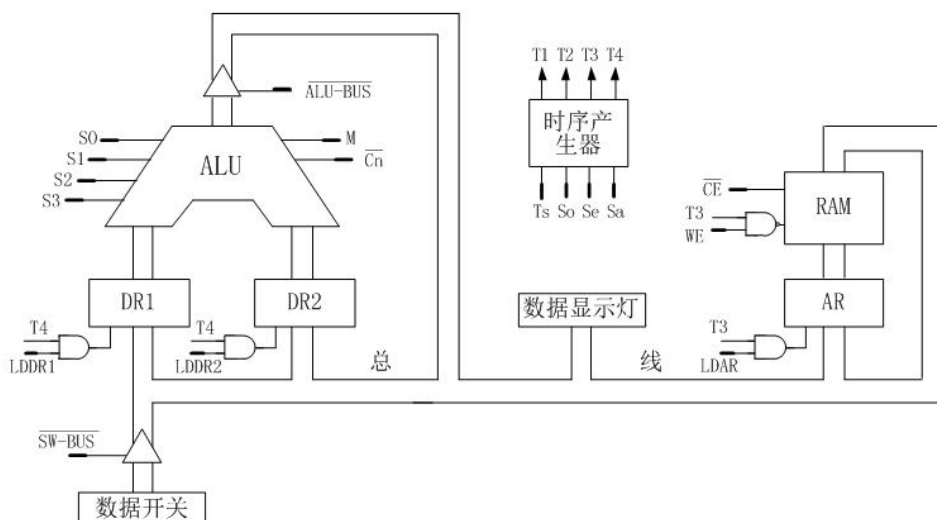


图 3.2 数据通路总框图

为方便进行实验，将图 3.2 中的所有控制信号归纳在表 3-1 中。实验的主要任务就是确定这些控制信号在每一个 CPU 周期的取值。

表 3-1 微命令集合

位	12	11	10	9	8	7	6	5	4	3	2	1	0
控 制 信 号	S 3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	LDA R	LDDR 1	LDDR 2	$\overline{\text{ALU-B}}$	$\overline{\text{SW-B}}$

可以设计不同的微命令组合，来实现不同的功能。例如，微命令组合 0000011001010 表示 DR 载入，数据开关三态门打开，存储器、DR2 和 ALU 三态门都关闭，其功能即为：将数据开关上的数据送入 DR1。注意，表 3-1 里的微命令只是实际计算机中的一部分，计算机运行所需要的微命令远不止这些。

在存储逻辑型控制器中，计算机需要用到的所有微命令组合都已预先设计好并存储在控制存储器中，由控制器根据程序自动找出需要的微命令组合，通过控制线发送给各执行部件执行，其中的每一个微命令组合又叫一条微指令。本实验用人工设置数据开关的方法来代替控制器生成微命令，完成一系列操作和任务。

3.4 实验内容与步骤

1. 运行虚拟实验系统，绘制实验电路，接好表 3-1 中列出的控制信号线，将控制信号线分别接至电路图上方的数据开关上，并仔细检查一遍，确保连接正确，连接好的电路如图 3.3 所示。

2. 进行电路预设置。将 DR1、DR2 和 AR 的 $\overline{\text{MR}}$ 置 1，时序发生器的 Step 置 1。

3. 打开电源开关。

4. 求 $A+B$ ，A 从数据开关输入，B 是存储器操作数，B 的地址也从数据开关输入，运算结果在数据显示灯上显示。具体步骤如下：

(1) 准备好要使用的微命令，如表 3-2 所示。

表 3-2 A+B 微命令

功能	微命令												
	S3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	LDAR	LDDR1	LDDR2	$\overline{\text{ALU-B}}$	$\overline{\text{SW-B}}$
数据开关→DR1	0	0	0	0	0	1	1	0	0	1	0	1	0
存储单元地址→AR	0	0	0	0	0	1	1	0	1	0	0	1	0
存储器操作数→DR2	0	0	0	0	0	1	0	0	0	0	1	1	1
DR1+DR2→BUS	1	0	0	1	0	1	1	0	0	0	0	0	1

(2) 设置控制信号：数据开关→DR1（0000011001010）；将数据开关设置为 A(00000011)；

单击时序发生器的 start 按钮，等待一个 CPU 周期后，数据开关上的值已存入 DR1。

(3) 设置控制信号：存储单元地址→AR（0000011010010）；将数据开关设为 B 的地址（00000010）；单击 start 按钮，等待一个 CPU 周期后，地址已存入 AR。

(4) 设置控制信号：存储器操作数→DR2（0000010000111）；单击 start 按钮。等待一个 CPU 周期后，B 的值已存入 DR2。

(5) 设置控制信号：DR1+DR2→BUS（1001011000001），运算结果在数据灯上显示。

5. 计算 C-D→存储单元 E，数据 C、D 和地址 E 都从数据开关输入。

(1) 设计微命令，填入表 3-3 中。

表 3-3 C-D→存储单元 E 微命令

功能	微命令												
	S3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	LDAR	LDDR1	LDDR2	$\overline{\text{ALU-B}}$	$\overline{\text{SW-B}}$
数据开关→DR1	0	0	0	0	0	1	1	0	0	1	0	1	0
数据开关→DR2	0	0	0	0	0	1	1	0	0	0	1	1	0
存储单元地址→AR	0	0	0	0	0	1	1	0	1	0	0	1	0
DR1-DR2→存储单元	0	1	1	0	0	0	0	1	0	0	0	0	1

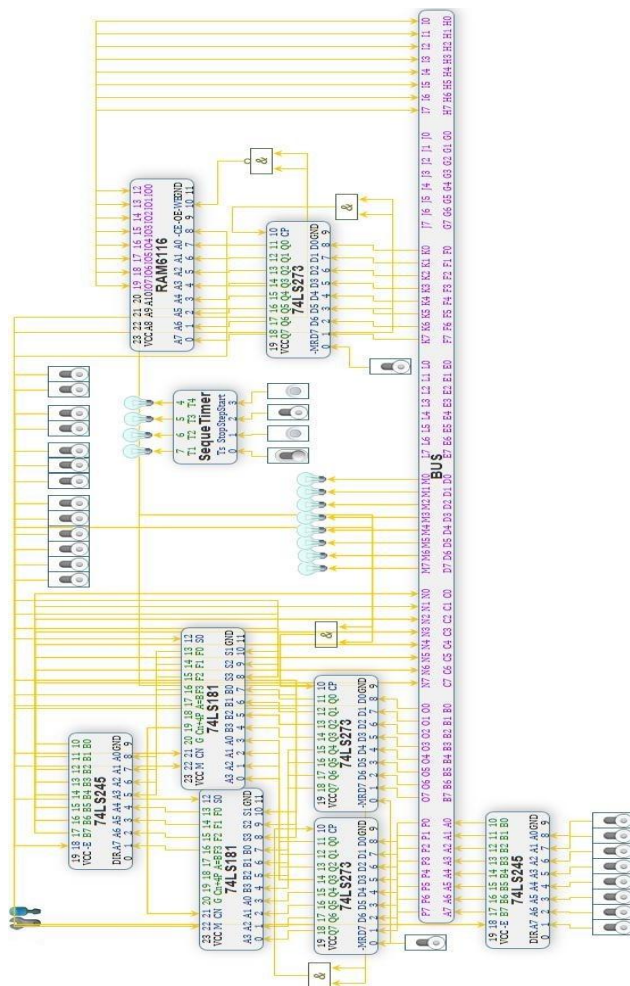


图 3.3 数据通路虚拟实验电路

(2) 设置控制信号：数据开关→DR1（0000011001010）；将数据开关设置为 C(00010111)；单击时序发生器的 start 按钮。等待一个 CPU 周期后，C 已存入 DR1。

(3) 设置控制信号：数据开关→DR2（0000011000110）；将数据开关设置为 D(00001000)；单击 start 按钮。等待一个 CPU 周期后，D 已存入 DR2。

(4) 设置控制信号：存储单元地址→AR（0000011010010）；将数据开关设置为 E(00000000)；单击 start 按钮。等待一个 CPU 周期后，地址 E 已存入 AR。

(5) 设置控制信号：DR1-DR2→存储单元（0110000100001）；单击 start 按钮。等待一个 CPU 周期后，运算结果已存入存储单元 00H。

(6) 单击菜单中的“工具/存储器芯片设置”，查看存储单元 00H 的值是否正确，

如果不正确，找到错误的原因，调试至正确为止，并将结果记录下来。

6. 计算 $F \wedge G \rightarrow$ 存储单元 H。F 和 G 都是存储器操作数，F、G 的地址以及地址 H 都从数据开关输入。

(1) 设计微命令，填入表 3-4 中。

表 3-4 $F \wedge G \rightarrow$ 存储单元 H 微命令

功能	微命令												
	S3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	LDAR	LDDR1	LDDR2	$\overline{\text{ALU-B}}$	$\overline{\text{SW-B}}$
存储单元地址 \rightarrow AR	0	0	0	0	0	1	1	0	1	0	0	1	0
存储器操作数 \rightarrow DR1	0	0	0	0	0	1	0	0	0	1	0	1	1
存储单元地址 \rightarrow AR	0	0	0	0	0	1	1	0	1	0	0	1	0
存储器操作数 \rightarrow DR2	0	0	0	0	0	1	0	0	0	0	1	1	1
存储单元地址 \rightarrow AR	0	0	0	0	0	1	1	0	1	0	0	1	0
$\text{DR1} \wedge \text{DR2} \rightarrow$ 存储单元	1	0	1	1	1	1	0	1	0	0	0	0	1

(2) 设置控制信号：存储单元地址 \rightarrow AR (0000011010010)；将数据开关设置为 F 的地址 (00000100)；单击 start 按钮。等待一个 CPU 周期后，地址已存入 AR。

(3) 设置控制信号：存储器操作数 \rightarrow DR1 (0000010001011)；单击 start 按钮。等待一个 CPU 周期后，F 已存入 DR1。

(4) 设置控制信号：存储单元地址 \rightarrow AR (000011010010)；将数据开关设置为 G 的地址 (00001000)；单击 start 按钮。等待一个 CPU 周期后，地址已存入 AR。

(5) 设置控制信号：存储器操作数 \rightarrow DR2 (0000010000111)；单击 start 按钮。等待一个 CPU 周期后，G 已存入 DR2。

(6) 设置控制信号：存储单元地址 \rightarrow AR (0000011010010)；将数据开关设置为 H (00001100)；单击 start 按钮。等待一个 CPU 周期后，地址 H 已存入 AR。

(7) 设置控制信号： $\text{DR1} \wedge \text{DR2} \rightarrow$ 存储单元 (1011110100001)；单击 start 按钮。等待一个 CPU 周期后，运算结果已存入存储单元 0CH。

单击菜单中的“工具/存储器芯片设置”，查看存储单元 0CH 的值是否正确，如果不正确，找到错误的原因，调试至正确为止，并将结果记录下来。

3.5 实验结果

本实验需要记录的实验结果如下:

1. 首先, 按要求设计微命令并填入表 3-3 中; 然后, 将对应实验步骤中的括号内容补充完整; 最后, 截图记录运算结果。

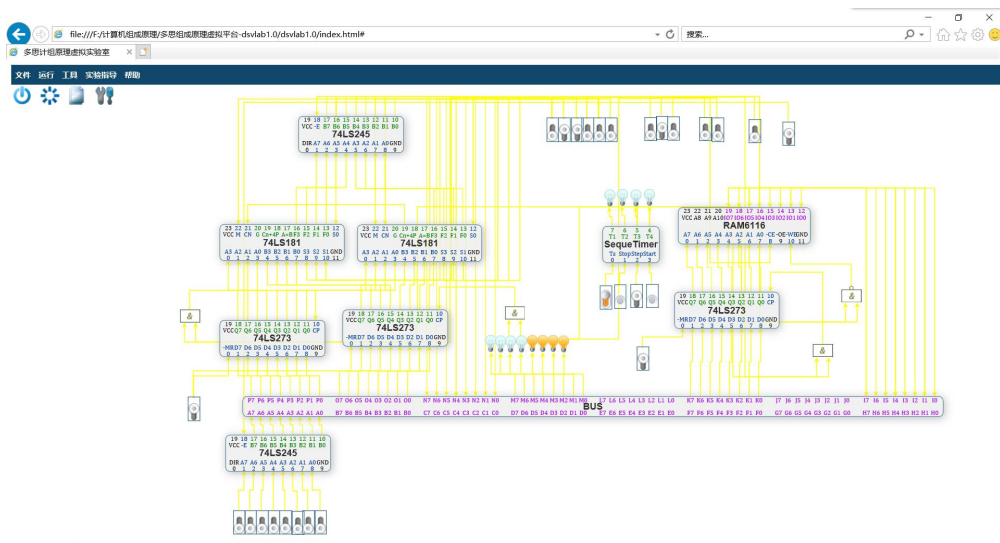


图 3-4 C-D→存储单元 E 微命令的虚拟电路运算结果图

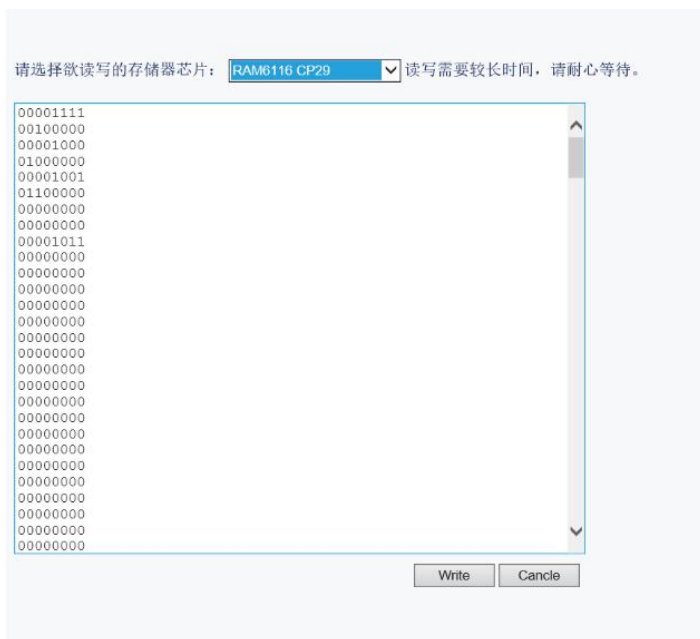


图 3-5 C-D→存储单元 E 微命令的存储芯片运算结果图

2. 首先，按要求设计微命令并填入表 3-4 中；然后，将对应实验步骤中的括号内容补充完整；最后，截图记录运算结果。

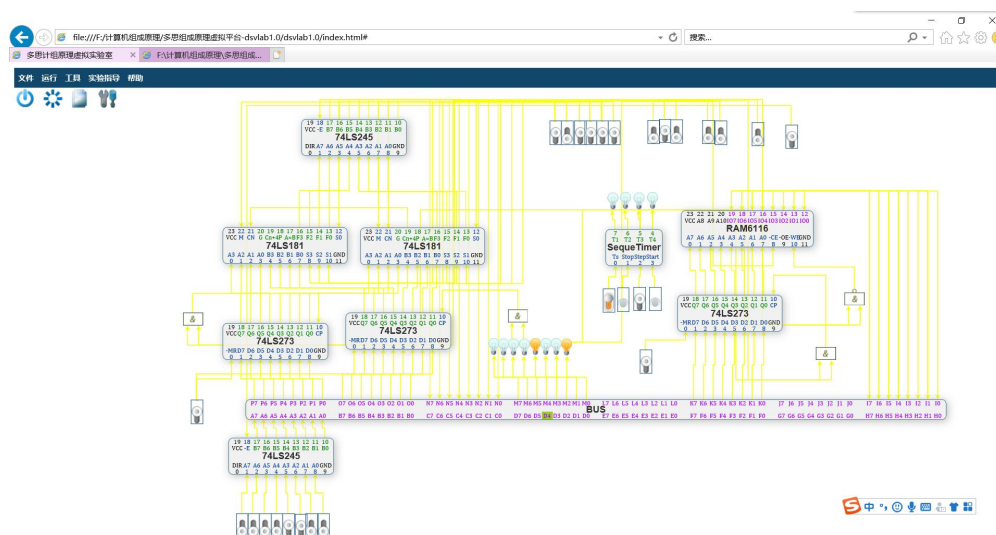


图 3-6 F/Λ→存储单元 H 微命令的虚拟电路运算结果图

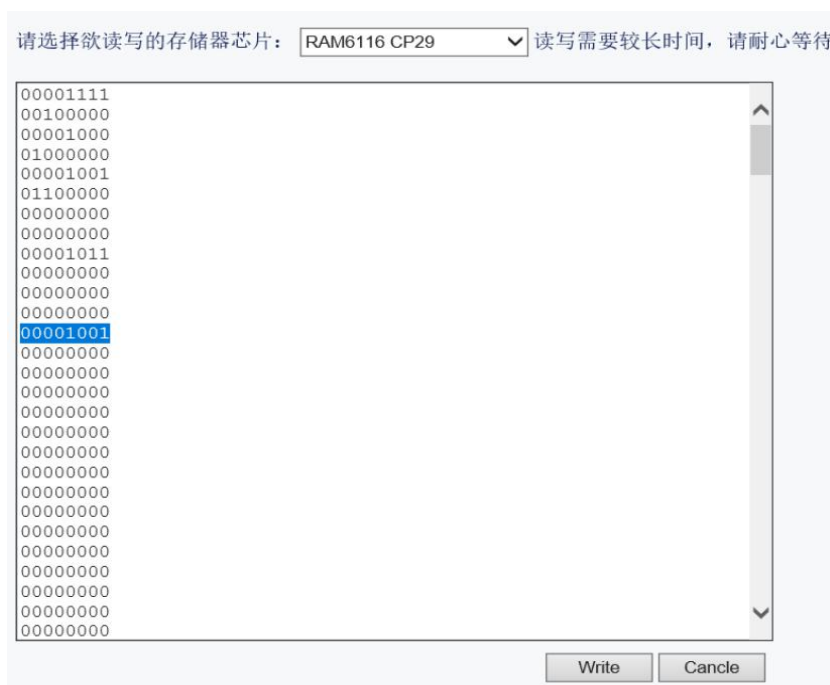


图 3-7 F/Λ→存储单元 H 微命令的存储芯片运算结果图

3.6 思考与分析

1. 总线的功能是什么？按连接部件可以分为几类？此实验中的总线属于哪一类？

总线（Bus）是计算机各种功能部件之间传送信息的公共通信干线，是 cpu、内存、输入、输出设备传递信息的公用通道，主机的各个部件通过总线相连接，外部设备通过相应的接口电路再与总线相连接。

按连接部件可以分为：数据总线、地址总线、控制总线。此实验中的总线属于数据总线

2.单总线结构有什么特点？多总线结构相对于单总线结构有什么优势？

答：在单总线结构中，CPU 与主存之间、CPU 与 I/O 设备之间、I/O 设备与主存之间、各种设备之间都通过系统总线交换信息。单总线结构的优点是控制简单方便，扩充方便。

多总线结构提高了微机系统信息传送的速率和效率，信息传送效率较高 CPU 与高速的局部存储器和局部 I/O 接口通过高传输速率的局部总线连接，速度较慢的全局存储器和全局 I/O 接口与较慢的全局总线连接，从而兼顾了高速设备和慢速设备，使它们之间不互相牵扯。

3.什么是微命令？什么是微操作？

答：微命令即控制部件通过控制线向执行部件发出各种控制命令。在微指令的控制字段中，每一位代表一个微命令。

微操作是执行部件接受了微命令后所执行的操作。

微操作是微命令的执行过程，微命令是微操作的控制信号。

实验 4 微程序控制器实验

4.1 实验目的

- (1) 掌握微程序控制器的组成原理和工作过程。
- (2) 理解微指令和微程序的概念，理解微指令与指令的区别和联系。
- (3) 掌握指令操作码与控制存储器中微程序的对应方法，熟悉根据指令操作码从控制存储器中读出微程序的过程。

4.2 实验要求

- (1) 做好实验预习，读懂实验电路图，熟悉实验元器件的功能特性和使用方法。
- (2) 按照实验内容与步骤的要求，独立思考，认真仔细地完成实验。
- (3) 写出实验报告。

4.3 实验原理

图 4.1 为实验电路图，其中 3 片 EPROM2716 构成控制存储器，1 片 74LS175 为微地址寄存器，与 74LS175 数据输入引脚相连的输入信号线及 6 个门电路构成了地址转移逻辑。注意，2716 输出信号中带后“#”的信号为低电平有效信号，不带后缀“#”的信号为高电平有效信号。为简化电路结构，本实验没有使用微命令寄存器，并且在虚拟实验系统中，将 3 片 EPROM 组合为一个虚拟 EPROM 组件。本实验使用的 EPROM 和时序发生器一样，均为虚拟实验系统提供的虚拟组件。

在存储逻辑型计算机中，一条机器指令对应了一个微程序，不同的机器指令对应了不同的微程序，执行一条指令其实就是运行其对应的一个微程序，微程序由微指令组成，是微指令的有序集合。微程序是在设计一台计算机时就预先设计好并且固化在只读存储器中的，以后每当要执行某条指令时，只需找到并运行其对应的微程序。

IR7~IR5。当判别位 P1=0 时，下一条微指的地址即为后续微地址 $\mu A3 \sim \mu A0$ ；当判别位 P1=1 时，下一条微指令的地址由指令操作码 IR7~IR5 决定，一般是将操作码进行简单变换，把变换后的值作为下一条微指令的地址，此地址就是该操作码对应的微程序的入口地址。

微地址寄存器 74LS175 为控制存储器提供微指令地址，当 $\overline{CLR}=0$ 时，微地址寄存器清零，从控制存储器 00H 地址开始执行微程序，地址转移逻辑生成下一条微指令的地址。此后，每当 T1 上升沿到来时，新的微指令地址会打入微地址寄存器，控制存储器随即输出这条微指令，地址转移逻辑继而生成下一条微指令的地址。如果时序信号连续发生，微指令也会按一定的顺序接连输出。

为了教学简单明了，本实验仅用到四条机器指令：IN(输入)、ADD(加法)、STA(存数)、JMP(无条件转移)，操作码分别为 000、001、010、011，指令格式如表 4-1 所示。

表 4-2 机器指令格式

助记符	机器码 (A 为内存地址 8bit)	长度	功能
IN	000XXXXXX	8bit	SW→R0
ADD	001XXXXXX A	16bit	R0+(A)→R0
STA	010XXXXXX A	16bit	R0→(A)
JMP	011XXXXXX A	16bit	A→PC(程序跳转到 A 地址执行)

上述四条指令的微程序流程设计如图 4.2 所示，其中一个方框就对应一条微指令，方框右上角的数字为八进制表示的微地址。一个方框也表示一个 CPU 周期，执行一条微指令需要一个 CPU 周期。四条指令对应四个微程序，每个微程序包括 N 条微指令，需要执行 N 个 CPU 周期。

图 4.2 中的每条微指令都按照表 4-1 的格式编写了二进制代码，并预存在控制存储器芯片 EPROM22716×3 中。其中部分微指令二进制代码如表 4-3 所示，注意：微地址用八进制表示。

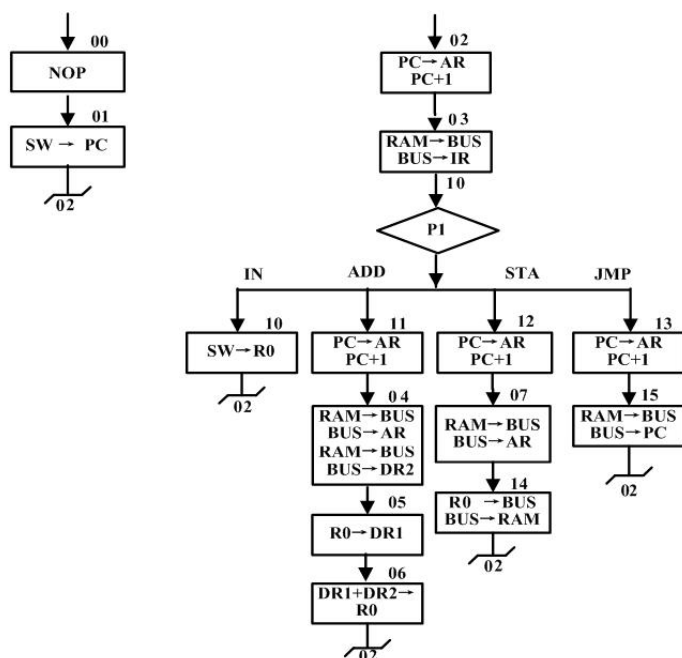


图 4.2 微程序流程图

4.4 实验内容与步骤

1. 运行虚拟实验系统，按照图 4.1 绘制实验电路，生成如图 4.3 所示电路。
2. 电路预设置：将 EPROM2716 芯片 $\overline{\text{CE}}$ 、 $\overline{\text{OE}}$ 、A4、A5 引脚置 0，微地址寄存器 74LS175 的 $\overline{\text{CLR}}$ 置 0，时序发生器的 Step 置 1。
3. 打开电源。此时由于 $\overline{\text{CLR}} = 0$ ，微地址寄存器清零，给出微程序入口地址 00H，控制存储器随之输出第 00 号微指令。
4. 将 $\overline{\text{CLR}}$ 设置为 1，否则微地址寄存器会一直处于清零状态。
5. 将 IR7~IR5 均设置为 0，思考并回答问题：若此时连续不断地发出时序信号，微程序的执行流程是怎样的？请按顺序写出前 10 条微指令的地址。

表 4-3 微程序二进制代码表

位	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
地址	S3	S2	S1	S0	M	Cn	CE	WE	LOAD	LDR0	LDDR1	LDDR2	LDIR	LDPC	LDAR	ALU-B	PC-B	SW-B	R0-B	P(1)	μ A3	μ A2	μ A1	μ A0
00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
01	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0
02	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1
03	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0
04	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1
05	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	0
06	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0
07	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0
10	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
12	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1
13	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	1	1	0	1
14	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0
15	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	1	0

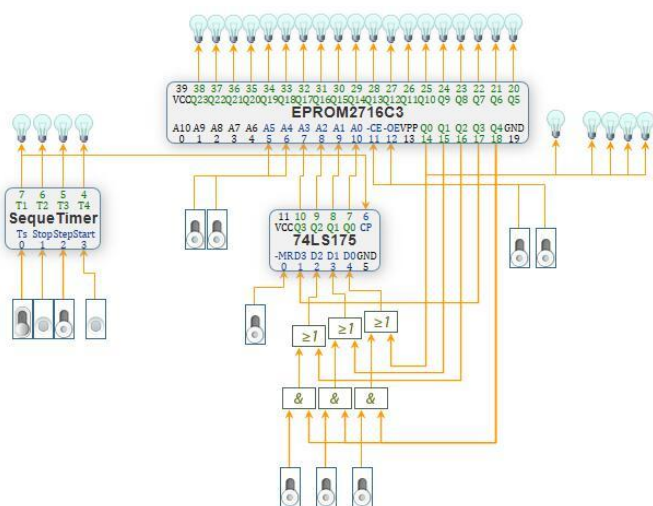


图 4.3 控制器虚拟实验电路

6. 连续单击 Start 按钮，观察微指令的输出顺序，检验控制存储器输出的微指令是
否与表 4-3 中的相符，验证上一步预测的顺序是否正确。

7. 设置 IR7~IR5 的不同组合，用单步方式分别读出 ADD、STA 和 JMP 三条指令的微程序，用后续微地址和判别指示灯跟踪微程序执行及转移情况，将表 4-3 中缺少的微程序代码补充完整。

8. 思考并回答问题：若不改变控制器实验电路，IN、ADD、STA 和 JMP 四条指令的微程序在控制存储器中的存放位置是否可以随意安排？有什么限制？为什么？

4.5 实验结果

本实验需要记录的结果是回答 4.4 节实验内容与步骤中，第 5、7、8 步提出的问题：

5. 答：将 IR7~IR5 均设置为 0，若此时连续不断地发出时序信号，微程序的执行流程是 00-01-02-03-10-02-03-10-02-03。

7. 将表 4-3 补充完整。

8. 答：若不改变控制器实验电路，IN、ADD、STA 和 JMP 四条指令的微程序在控制存储器中的存放位置不可以随意安排。微程序在控制存储器中是按照顺序存储的，本实验中采取的是逐条取指的方式来取指令，程序计数器 PC 来控制将要执行哪一条指令，给出将要执行的指令的地址。通常情况下，是 PC+1 顺序执行指令，有时也会有转移指令，使 PC 的值转移到相应的地址。所以 IN、ADD、STA 和 JMP 四条指令的微程序在控制存储器中的存放位置不可以随意安排，如果随意安排会导致程序混乱，不能正常执行。

4.6 思考与分析

1. 微程序控制器主要由哪些部件组成？各部件的功能是什么？

答：微程序控制器主要由控制存储器、微指令寄存器和地址转移逻辑三大部分组成。

控制存储器是一种只读存储器，用来存放全部指令系统的微程序。对于控制存储器来说，一般对其要求速度快，读出周期要短。

微指令寄存器是用来存放控制存储器读出的一条微指令信息。微指令寄存器保存着一条微指令的操作控制字段和判别测试字段的信息。

当微程序出现条件转移的时候，一般需要通过判别测试字段 P 和执行部件的“状态

条件”反馈信息，去修改位地址寄存器的内容，并且按照改好的内容去读下一条微指令。地址转移逻辑就承担自动完成修改微地址的任务。

2. 本实验中，地址转移逻辑电路是怎样利用判别测试字段（P 字段）实现微程序分支的？

答：当测试字段（P 字段）置 1 时，执行程序分支；P 为 0 时顺序执行下一条微指令。

3. 如果把微程序控制器看作一个黑盒子，那么它的输入信号有哪些？这些信号是哪些部件提供给它的？它的输出信号有哪些？这些信号是发送给哪些部件的？

输入信号是微程序的地址，就是指令读取并且译码后，送到微程序控制器。输入信号可以由指令寄存器 AR 等提供给它。

输出信号是一系列的控制信号，即那 23 位控制信号，可以发送给 ALU、寄存器、程序计数器等部件。

实验 5 简单模型机实验

5.1 实验目的

- (1) 通过总线将微程序控制器、运算器、存储器等联机，组成一台模型计算机。
- (2) 用微程序控制器控制模型机数据通路，运行由 4 条机器指令组成的简单程序。
- (3) 掌握微指令与机器指令的关系，建立整机概念。

5.2 实验要求

- (1) 做好实验预习，复习微指令和机器指令的概念，读懂实验电路图，熟悉实验元器件的功能特性和使用方法。
- (2) 对于实验任务中的问题，在实验前预先给出答案，以便与实验结果相比较。
- (3) 在实验过程中单步运行微程序，注意理解微程序与程序的联系和区别。
- (4) 写出实验报告。

5.3 实验原理

本实验综合了前面实验的电路，将运算器模块、存储器模块和控制器模块通过总线连接在一起，组成了一个简单的模型机，其电路如图 5.1 所示。

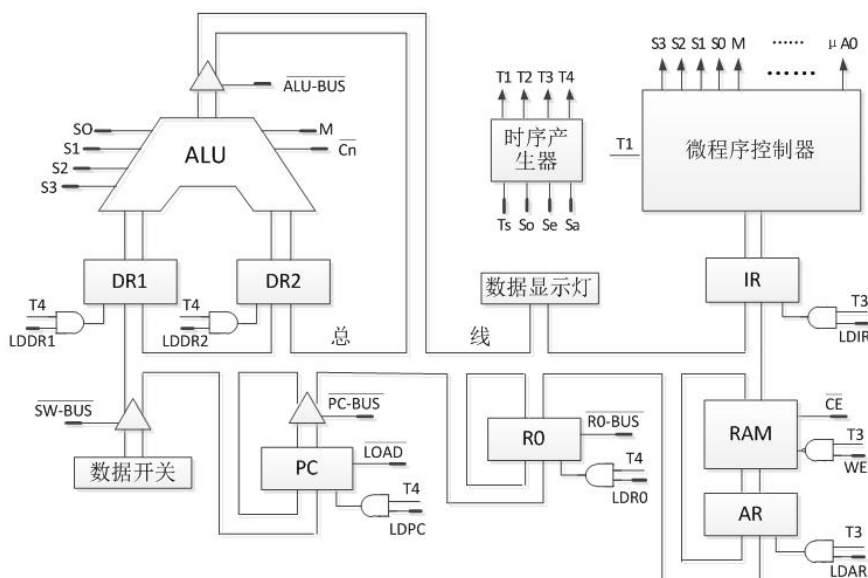


图 5.1 简单模型机总框图

在控制器实验中，实现了自动按照 AR 中的指令逐条取出对应的微指令。在本实验中，程序存储在 RAM 中，微程序存储在控制存储器中，要实现自动从 RAM 里逐条取出指令放入 IR，并按照 IR 中的指令自动从控制存储器读出相应的微程序执行。

在本实验中，逐条取指是通过程序计数器 PC 进行控制的，它用于生成下一条要执行的指令的地址。在程序开始执行前，必须将 PC 的值设置为程序的起始地址，即程序的一条指令所在的内存单元地址。在程序执行过程中，CPU 将自动修改 PC 的值，使其保持为下一条指令的地址。由于大多数指令都是顺序存储并执行的，所以修改的过程通常只是简单的对 PC 加 1。当程序转移时，转移指令实际就是将 PC 的值设置为转去的目的地址，以实现跳转。

数据开关（SW7-SW0）设置的程序起始地址经三态门发送至总线。PC 从总线上接收起始地址并设置为计数初值，PC 中的值经过三态门送至总线，PC 的值递增 1。地址寄存器 AR 从总线上获取地址并送至存储器，存储器按地址进行读操作，将读出的指令或数据发送至总线。IR 从总线上获取指令并锁存。控制存储器则按照 IR 中的指令自动读出相应的微程序执行。

本实验用到的微指令长度为 24bit，微指令格式如表 5-1 所示。

表 5-1 微指令格式

位	23	22	21	20	19	18	17	16	15	14	13	12
控制信号	S3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	$\overline{\text{LOAD}}$	LDR0	LDDR1	LDDR2

位	11	10	9	8	7	6	5	4	3	2	1	0
控制信号	LDIR	LDPC	LDAR	$\overline{\text{ALU-B}}$	$\overline{\text{PC-B}}$	$\overline{\text{SW-B}}$	$\overline{\text{R0-B}}$	P(1)	μA3	μA2	μA1	μA0

本实验使用的微程序流程如图 5.2 所示。

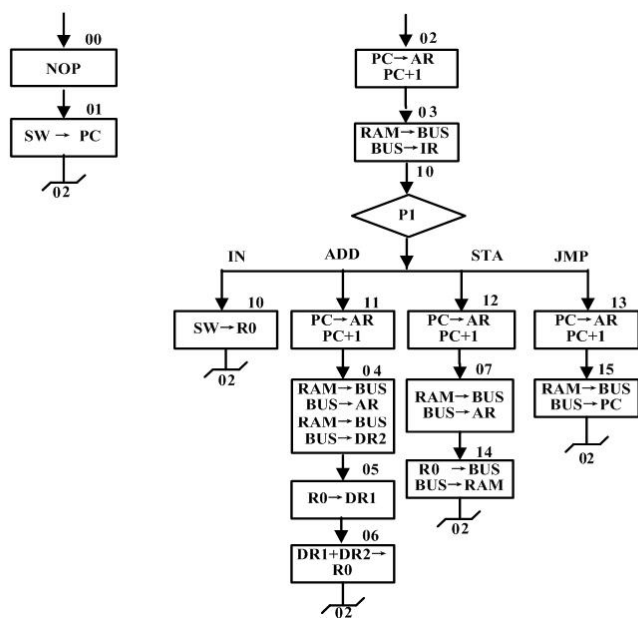


图 5.2 微程序流程图

对应的微程序代码存放在控制存储器中，如表 5-2 所示。

表 5-2 微程序二进制代码表

位	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
地址	S3	S2	S1	S0	M	Cn	CE	WE	LOAD	LDR0	LDDR1	LDDR2	LDIR	LDPC	LDAR	ALU-B	PC-B	SW-B	R0-B	P(1)	$\mu A3$	$\mu A2$	$\mu A1$	$\mu A0$
00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
01	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0
02	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1
03	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0
04	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1
05	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	0
06	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0
07	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0
10	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
12	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1
13	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	1	1	0	1
14	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0
15	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	1	0

一条指令对应一个微程序，一个微程序是多条微指令的有序集合。

模型机共包含 4 条指令，指令格式如表 5-3 所示。本实验用这 4 条指令编写了一个简单程序，并已存入 RAM。RAM 中的程序和数据如表 5-4 所示。

表 5-3 机器指令格式

助记符	机器码（A 为内存地址 8bit）	长度	功能
IN	000XXXXXX	8bit	SW→R0
ADD	001XXXXXX A	16bit	R0+(A)→R0
STA	010XXXXXX A	16bit	R0→(A)
JMP	011XXXXXX A	16bit	A→PC(程序跳转到 A 地址执行)

表 5-4 RAM 中的程序和数据

地址（八进制）	内容	含义
00	00000000	IN（开关数据自定）
01	00100000	ADD
02	00001000	10（八进制）
03	01000000	STA
04	00001001	11（八进制）
05	01100000	JMP
06	00000000	00
07		
10	00001011	
11		求和结果

5.4 实验内容与步骤

1. 运行虚拟实验系统，按照图 5.1 绘制实验电路，生成如图 5.3 所示电路。
2. 打开电源开关。
3. 进行电路预设置。将 DR1、DR2 和 AR 的 \overline{MR} 置 1，将计数器的 \overline{CR} 、ENT、ENP 置 1，时序发生器的 Step 置 1（可在开电源之前设置）。微地址寄存器 74LS175 和指令寄存器 IR 的 \overline{MR} 置 1。此时微地址寄存器和 IR 已初始化为零，模型机将从控制存储器的零地址开始运行。
4. 在数据开关（SW7~SW0）上设置好程序的起始地址（00000000）。
5. 单击 1 次时序发生器的 Start 按钮，思考并回答问题：此时执行的是微程序流程图中的第几条微指令？作用是什么？
6. 再单步执行两条微指令，思考并回答问题：这两条微指令的作用是什么？
7. 通过数据开关（SW7~SW0）设置操作数 1 的值为 00010100。
8. 单击 Start，执行微指令 SW→R0，将操作数 1 保存到累加器 R0 中。

表 5-5 总线数据表

序号	总线上数据 (二进制)	微指令编号 (八进制)	意义 (地址用二进制表示)
1	00000001	02	当前 PC 的值, 即内存地址 01
2	00000010	02	递增 1 后的 PC 值
3	00100000	03	内存地址 01 中的 ADD 指令操作码
4	00000010	11	当前 PC 的值, 即内存地址 10
5	00000011	11	递增 1 后的 PC 的值
6	00001000	04	加法一个操作数的地址
7	00001011	04	加法的一个操作数
8	00010100	05	R0 的值, 加法的另一个操作数
9	00011111	06	加法的和
10	00000011	02	当前 PC 的值, 即内存地址 11
11	00000100	02	递增 1 后 PC 值
12	01000000	03	11 地址中的值, STA 的操作码
13	00000100	12	当前 PC 的值, 即内存地址 100
14	00000101	12	递增 1 后的 PC 值
15	00001001	07	内存地址 100 中的数据, 此数据也是一个地址
16	00000000	07	内存地址 1001 中的数据
17	00011111	14	加法的和
18	00000101	02	当前 PC 的值, 即内存地址 101
19	00000110	02	递增 1 后的 PC 值
20	01100000	03	101 地址中的值, JMP 指令的操作码
21	00000110	13	当前 PC 的值, 即内存地址 110
22	00000111	13	递增 1 后的 PC 值
23	00000000	15	110 地址中的值, 跳转指令的目的地址

5.5 实验结果

本实验需要记录的结果是回答 5.4 节实验内容与步骤中，第 5、6、9、10 步提出的问题：

5. 答：单击 1 次时序发生器的 Start 按钮，此时执行的是微程序流程图中的第 01 条微指令（SW→PC），此微指令的功能是将数据开关（SW7-SW0）设置的程序起始地址即 00000000 经总线送到 PC，作为起始地址并且置为 PC 的计数初值。

6. 答：①执行 02 条微指令。此微指令的功能是将 PC 中的指令地址送到总线，程序计数器 PC 的值加一，接着总线上的指令地址送到地址寄存器 AR。

②执行 03 条微指令。此微指令的功能是从寄存器中取出操作码送到总线上，再将此操作码从总线上送到指令寄存器 IR。

9. 将表 5-5 补充完整。

10. 对每条指令的执行结果进行说明，并将使用菜单“工具/存储器芯片设置”选项，查看运算结果时的界面进行截图。

指令	结果说明
IN SW→R0	将数据开关中的 00010100B 送入 R0 寄存器
ADD R0+ (A) →R0	将 R0 寄存器中的 00010100B 送到 ALU，取出 RAM 中的 A 地址 00001011B 中的数送到 ALU，两个数相加结果送到 R0 寄存器
STA R0→ (A)	将 R0 寄存器中的值 00011111B 送到 RAM 中的 A 地址（00001001B）
JMP A→PC	跳转指令，将程序计数器 PC 的值跳转到 A 地址（00000000B）

指令 IN 运算结果截图：



指令 ADD 运算结果截图：



指令 STA 运算结果截图：



指令 JMP 运算结果截图：



5.6 思考与分析

1. 指令与微指令、程序与微程序之间有什么联系？

答：一个程序是由一条或多条指令组成的，一条指令对应一个微程序，一个微程序由一条或多条微指令构成。一条指令的功能是若干条微指令组成的序列来完成的。一条指令所完成的操作划分成若干条微指令来完成，由微指令进行解释和执行。

2. 无论是程序还是微程序都必须按一定的顺序执行其中的指令或微指令，请分别说明它们确定下一条要执行的指令或微指令的方法。

答：对于程序来说，程序计数器 PC 控制着指令的执行，通过指令是顺序存储的，所以一般 PC 递增来执行下一条指令，有时也会有跳转指令，使 PC 的值进行跳转，执行跳转后的地址中的指令。

对于微程序来说，如何确定下一条要执行的微指令就是如何确定下一条微指令的地址，通常是通过微指令顺序控制字段由设计者指定或者设计者指定的判断字段控制产生后继微指令地址。

封面设计： 贾丽

地 址： 中国河北省秦皇岛市河北大街 438 号

邮 编： 066004

电 话： 0335-8057068

传 真： 0335-8057068

网 址： <http://jwc.ysu.edu.cn>