

兰州大学信息科学与工程学院实验报告

实验成绩：_____

学生姓名：_____杨添宝_____

学 号：_____320170941671_____

年级专业：_____2017 级计算机基地班_____

指导老师：_____赵继平_____

实验课程：_____计算机组成原理实验_____

实验题目：_____模型机的总体设计_____

一、实验目的

(1) 掌握了各个单元模块的工作原理，进一步将其组成完整的系统，构造 1 台基本的模型计算机。

(2) 在本试验中，我们将规划读写内存、寄存器、数值计算等功能，并且编写相应的微程序。请具体上机调试各个模块单元以便进一步掌握整机的概念。

二、实验原理

在各个模块实验中，各模块的控制信号都是由实验者手动模拟产生的。而在真正的实验系统中，模型机的运行是在微程序的控制下，实现特定指令的功能。在本实验平台中，模型机从内存中取出、解释、执行机器指令都将由微指令和与之相配合的时序来完成，即 1 条机器指令对应 1 个微程序。

三、模型机的总体设计

1. 设计整机逻辑框图：

根据设计要求，对实验仪硬件资源进行逻辑组合，便可设计出该模型机的整机逻辑框图。如图 1 是较典型的实验计算机整机逻辑框图。

简单的模型计算机是由算术逻辑运算单元、微程序单元、堆栈寄存器单元、累加器、启停、时序单元、总线和存储器单元组成。

在模型机中，我们将要实现 RAM 的读写指令，寄存器的读写指令，跳转指令，ALU 的加、减、与、或指令。把通用寄存器作为累加器 A，进行左、右移等指令，整体构成一个单累加器多寄存器的系统。

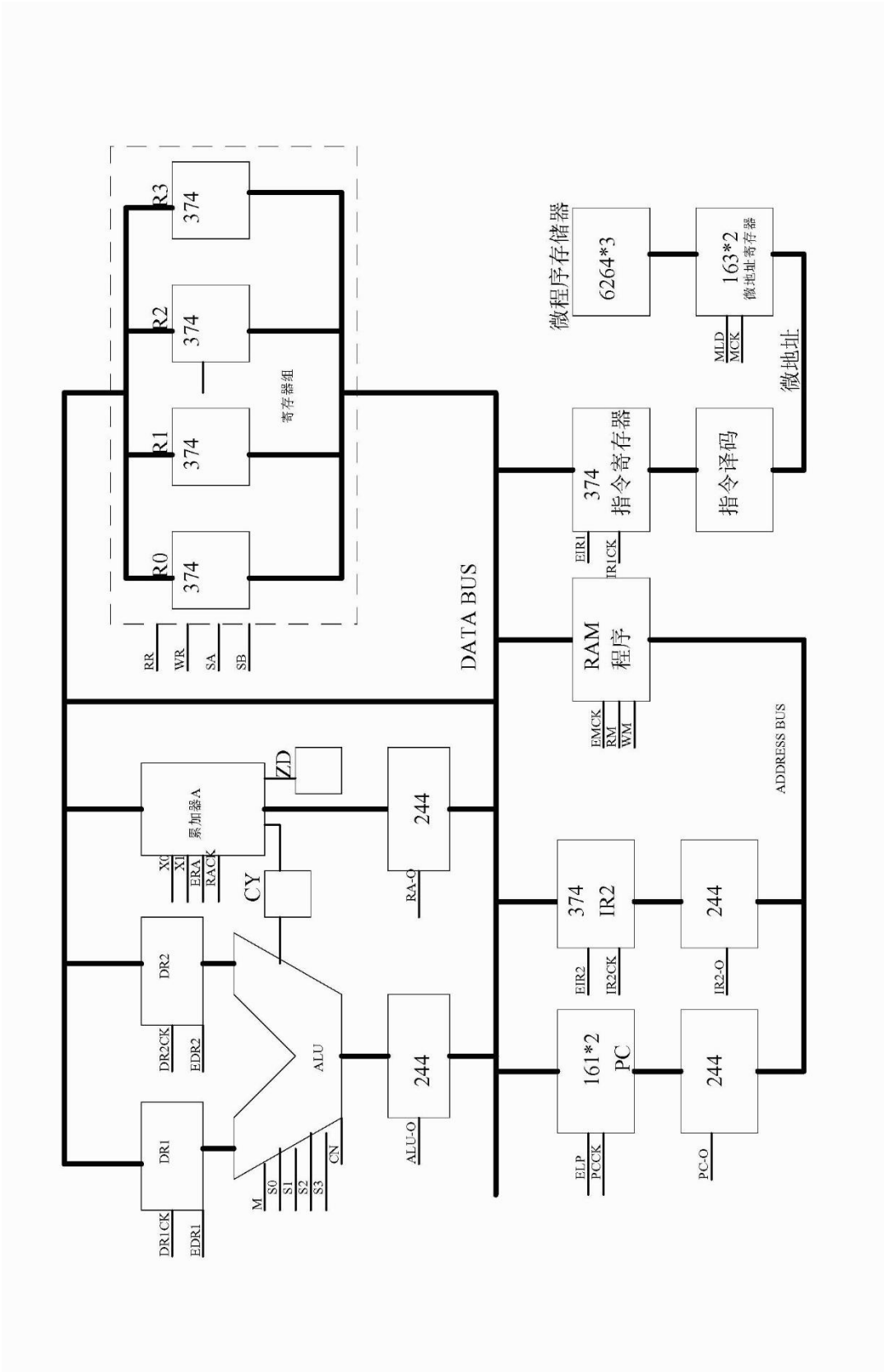


图 1

2. 设计指令系统:

本实验平台内采用的是 8 位数据总线和 8 位地址总线方式,在设计指令系统时,应考虑有哪几种类型的指令,哪几种寻址方式和编码方式。

(1) 指令类型:

①算术/逻辑运算类指令:

如: 加法、减法、取反、逻辑运算

ADD A, Ri, SUB A,Ri

②移位操作类指令:

带进位或不带进位的移位指令。

RRC A, RR A

③数据传送类指令:

CPU 内部寄存器之间数据传递

MOV A, Ri

MOV Ri, A

④程序跳转指令

跳转指令分为无条件跳转指令和有条件跳转指令。可根据寄存器内容为零来标志 (ZD)、有无进位来标志 (CY), 也可根据用户自定义标志。

JMP addr 无条件跳转

JZ addr ZD=0 时跳转

JC addr CY=0 时跳转

JN addr 自定义

⑤存储器操作类指令

存储器读/写指令。把内存某单元内容写入寄存器中或把寄存器中的内容写入存储器。如:

LDA addr (addr)->A

STA addr (A)->addr

(2) 操作数寻址方式及编码

①直接地址寻址:

如: 双字节指令

LDA addr (addr)->A

STA addr (A) ->addr



②寄存器直接寻址：

指令字节中含有寄存器选择码，决定选哪个寄存器进行操作。

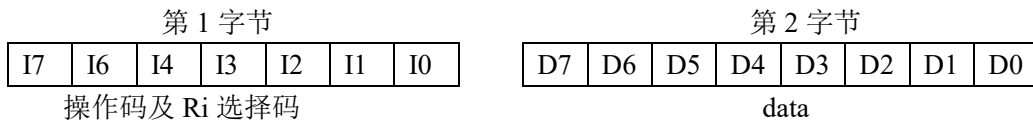
如：单字节指令

MOV A, Ri (Ri)->A



如：双字节指令

MOV Ri, #data data->Ri



③寄存器间接寻址

如：单字节指令：

MOV A,@Ri (Ri)->A



④立即数寻址

如： MOV A, # data data->A

MOV Ri, # data data->Ri

第 1 字节								第 2 字节							
I7	I6	I5	I4	I3	I2	I1	I0	D7	D6	D5	D4	D3	D2	D1	D0
操作码及 Ri 选择码								data							

四、设计微程序及其实现方法

在本实验平台的硬件设计中，采用 24 位微指令，若微指令采用全水平不编码纯控制场的格式，那么至多可有 24 个微操作控制信号，可由微代码直接实现。如果采用多组编码译码，那么 24 位微代码通过二进制译码可实现 2^n 个互斥的微操作控制信号。

由于模型机指令系统规模较小，功能也不太复杂，所以采用全水平不编码纯控制场的微指令格式。在模型机中，用指令操作码的高 4 位作为核心扩展成 8 位的微程序入口地址 MD0~MD7，这种方法称为“按操作码散转”（如下表所示）。

微程序首地址形成						
MD6	MD5	MD4	MD3	MD2	MD1	MD0
0	I7	I6	I5	I4	1	1

按操作码散转						
指令操作码						微程序首地址
MD7、MD6	I7	I6	I5	I4	MD1、MD0	MD7~MD0
0	0	0	0	0	1	003H
0	0	0	0	1	1	007H
0	0	0	1	0	1	00BH
0	0	0	1	1	1	00FH
0	0	1	0	0	1	013H
0	0	1	0	1	1	017H
0	0	1	1	0	1	01BH
0	0	1	1	1	1	01FH

0	1	0	0	0	1	023H
0	1	0	0	1	1	027H
0	1	0	1	0	1	02BH
0	1	0	1	1	1	02FH
0	1	1	0	0	1	033H
0	1	1	0	1	1	037H
0	1	1	1	0	1	03BH
0	1	1	1	1	1	03FH

每条指令由不超过 4 条的微指令组成，那么可根据下表组成每条微程序的首地址。微指令的运行顺序为下地址确定法，即采用计数增量方法，每条微指令执行过后微地址自动加 1，指向下一条微指令地址。例如：确定了一条程序的微程序入口地址为 07H，那么当执行完 07H 这条微指令后微地址加 1，指向 08H 微地址。微地址寄存器由 2 片 74LS161 组成，当模型机在停止状态下，微地址被清零。当实验平台开始运行时，微地址从 00H 开始运行。且 00H 放置一条取指指令，根据程序开始地址从内存中读出第一条指令。

00	取指微指令
01	
02	
03	减法指令微程序（1）
04	减法指令微程序（2）
05	
06	
07	MOV 指令微程序（1）
08	MOV 指令微程序（2）
09	MOV 指令微程序（3）
0A	
0B	
0C	

0D	
0E	
0D	
...	...

五、设计实验平台运行中时序安排

由于模型机已经确定了指令系统，微指令采用全水平不编码纯控制场的格式，微程序的入口地址采用操作码散转方式，微地址采用计数增量方式，所以可确定模型机中时序单元中所产生的每一拍的作用。

在本实验中为了让实验者更好地观察实验的各个中间过程中各寄存器的值，由监控单元产生一个 PLS-O 的信号来控制时序产生（如图 2）。PLS-O 信号经过时序单元的处理产生了 4 个脉冲信号。4 个脉冲信号组成一个微周期，为不同的寄存器提供工作脉冲。

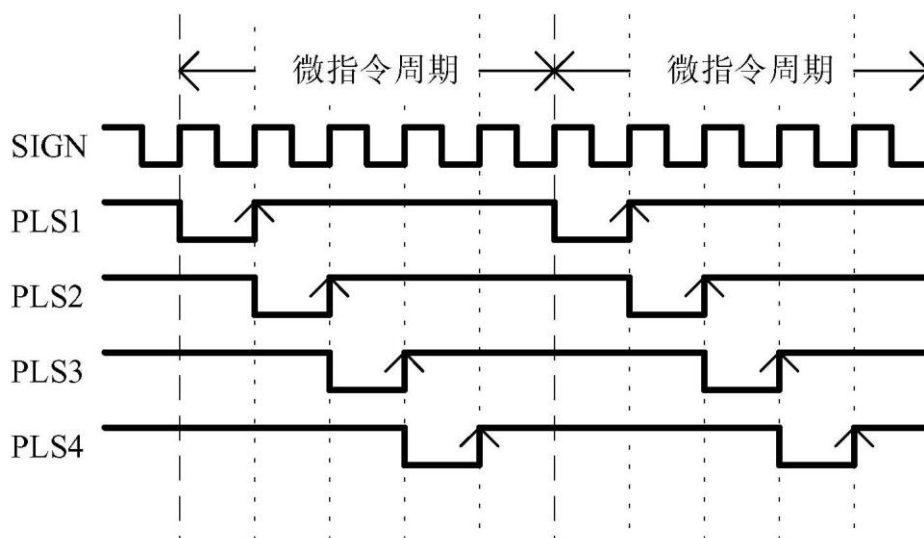


图 2

PLS1: 微地址寄存器的工作脉冲，用来设置微程序的首地址及微地址加 1。

PLS2: PC 计数器的工作脉冲，根据微指令的控制实现 PC 计数器加 1 和重置 PC 计数器（跳转指令）等功能。

PLS3: 把 24 位微指令打入 3 片微指令锁存器。

PLS4: 把当前总线上的数据打入微指令选通的寄存器中。

六、设计指令执行流程

根据模型机整机逻辑图和目前硬件条件来设计指令系统中每条指令的执行流程。在每个系统中，一条指令从内存取出到执行完毕，需要若干个机器周期，任何指令中都必须有一个机器周期作为“取指令周期”，称为公操作周期。而一条指令共需几个机器周期取决于指令在机器内实现的复杂程度。

对于微程序控制的计算机，在设计指令执行流程时，要保证每条微指令所含的微操作的必要性和合理性，还应知道总线 IAB, IDB, OAB, ODB 仅是传输信息的通路，没有寄存信息的功能，而且必须保证总线传输信息时信息的唯一性。

以下描述取指微指令执行过程：

在模型机处于停机状态时，模型机的微地址寄存器被清零，微指令锁存器输出无效（为高）。在处于停机状态时，脉冲 PLS1 对微地址寄存器

（74LS161）无效，微地址寄存器保持为零。脉冲 PLS2 对 PC 计数器无效，同时 PLS2 把 HALT=1 打入启停单元中的运行状态寄存器（74LS74）中，把模型机置为运行状态，使微程序锁存器输出有效。PLS3 把微程序存储器 00H 单元（00H 单元存放着取指微指令）中的内容打入微指令锁存器中，并且输出取指微指令。PLS4 把从程序存储器中读出的数据打入指令寄存器中。

在模型机处于运行状态时，脉冲 PLS1 将微地址寄存器（74LS161）加 1，脉冲 PLS2 将 PC 计数器加 1，PLS3 把微程序存储器中的微指令打入微指令锁存器并且输出。PLS4 把当前总线上的数据打入当前微指令所选通的寄存器。

以下举例说明指令的数据信息流：

例一：单字节指令 ADD A, Ri

该指令功能为 $(A) + (Ri) \rightarrow A$ ，需执行如下微操作：

$(PC)+1 \xrightarrow{PCLK} PC$	； PC 加 1，为取下条指令字节准备
$(A) \xrightarrow{EDR1, PLS4} DR1$	； 累加器 A 内容送 ALU，
$(Ri) \xrightarrow{EDR2, PLS4} DR2$	； 寄存器 Ri 内容送 ALU，

$(A) + (R_i) \xrightarrow{\overline{Cn}, M, S3, S2, S1, S0} IDB \xrightarrow{X0, X1, ERA} A$; 使 ALU 执行加法, 结果经 IDB 最后送入累加器 A
 $C_{n+4} \xrightarrow{ALU-O, M} CY$; 据加法结果置进位标志 CY
 $(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址
 $(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1, PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 4 个微指令周期组成:

- ①PC+1 为取下条指令字节准备、累加器 A 内容送 ALU 的 DR1 锁存器
- ②寄存器 Ri 内容送 ALU 的 DR2 锁存器
- ③ALU 将计算结果送累加器 A, 据加法结果置进位标志 CY
- ④取指微指令, 从内存读出指令送指令寄存器

例二：双字节指令 LDA addr

该指令功能为 (addr) → A, 需执行如下微操作:

$(PC) + 1 \xrightarrow{PCCK} PC$; PC 加 1, 为取本指令下一字节准备
 $(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; 根据 PC 访问内存
 $(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EDR2} IR2$; 取出本指令第二字节送 IR2
 $(PC) + 1 \xrightarrow{PCCK} PC$; PC 加 1, 为取下条指令字节准备
 $(IR2) \xrightarrow{IR2-O} IAB \xrightarrow{BUS} OAB$; IR2 形成新的地址
 $(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{X0, X1, ERA, PLS4} A$; 从内存读出操作数存入 A
 $(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址
 $(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1, PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 3 个微指令周期组成：

PC+1 从内存取得指令第 2 个字节送入 IR2。

PC+1 为取下条指令字节准备 IR2，IR2 寄存器的内容输出到地址总线上取值，将取得的值送入累加器 A

③取指微指令，从内存读出指令送指令寄存器

例三：双字节指令 JMP addr

该指令功能为 addr→PC，需执行如下微操作：

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1，为取本指令下一字节准备

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{PLS4} IR2$; 取本指令第二字节 IR2

$IR2 \xrightarrow{IR2-O} IAB \xrightarrow{ELP} PC$; 由 IR2 内容形成转移地址送 PC

$(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1, PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 2 个微指令周期组成：

PC+1 从内存取得指令第 2 个字节送入 IR2。

IR2 内容形成转移地址送 PC，从内存读出指令字节送指令寄存器

例四：单字节指令 RRC A ;该指令功能是将累加器 A 中内容带进位 CY 右环移一位，即

→CY→A7→A6→· · ·→A1→A0→，需执行如下微操作：

↑ _____ |

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1, 为取下条指令字节做准备

$(A)、CY \xrightarrow{X0,X1,ERA,PLS3} A、CY$; 控制累加器 A 带 CY 右环移一位 (移入 CY 作 A7, 移出的 A0 送 CY)

$(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 程序计数器内容作访问内存地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1、PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 2 个微指令周期组成:

①PC+1 为取下条指令字节准备、控制累加器 A 带 CY 右环移一位; (移入 CY 作 A7, 移出的 A0 送 CY)

②取指微指令, 从内存读出指令送指令寄存器

例五: 单字节指令 MOV A, @Ri

该指令功能为 $(Ri) \rightarrow A$, 需执行如下操作:

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1, 为取下条指令字节做准备

$(Ri) \xrightarrow{RR,SA,SB} IDB \xrightarrow{EIR2,PLS4} IR2$; 读出 Ri 中内容经 IDB, 存入 IR2

$(IR2) \xrightarrow{IR2-O} IAB \xrightarrow{BUS} OAB$; 由 IR2 的 8 位形成操作数地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \rightarrow A$; 从内存读出操作数送入累加器 A

$(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1、PLS4} IR1$; 从内存地址单元读出指令字节送指令寄存器 IR1。

此程序由 3 个微指令周期组成:

①PC+1 为取下条指令字节准备、送指令寄存器单元 Ri 中的内容到 IR2 寄存器

②IR2 寄存器的内容输出到地址总线上取值，将取得的值送入累加器 A

③取指微指令，从内存读出指令字节送指令寄存器

例六：双字节指令 MOV A, #data

该指令功能为 data→A，需执行如下微操作：

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1，为取本指令下一字节准备

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{X0,X1,ERA,PLS4} A$; 取出本指令第二字节
(即 data) 送 A

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1，为取下条指令字节做准备

$(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1, PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 2 个微指令周期组成：

PC+1 从内存取得指令第 2 个字节送入累加器 A。

PC+1 执行取指微指令，从内存读出指令字节送指令寄存器

例七：双字节指令 MOV Ri, #data

该指令功能为 data→Ai，需执行如下微操作：

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1，为取本指令下一字节准备

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{RR,SA,SA} Ri$; 取出本指令第二字节（即 data）送 Ri

$(PC)+1 \xrightarrow{PCCK} PC$; PC 加 1，为取下条指令字节做准备

$(PC) \xrightarrow{PC-O} IAB \xrightarrow{BUS} OAB$; PC 计数器内容作访问内存地址

$(RAM) \xrightarrow{RM} ODB \xrightarrow{BUS} IDB \xrightarrow{EIR1, PLS4} IR1$; 从内存该地址单元读出指令字节送指令寄存器 IR1。

此程序由 2 个微指令周期组成：

- ①PC+1 从内存取得指令第 2 个字节送入寄存器 Ri。
- ②PC+1 执行取指微指令，从内存读出指令字节送指令寄存器

七、实验思考

1. 在表格中空白位置填写正确答案。

指令助记符	指令编码	入口地址
MOV A, #data	5FH	17H
ADD A, R1	0DH	03H
STA addr	8FH	23H
JC addr	B7H	2FH

2. MOV A, R0 与 MOV A, @R0 的区别。

MOV A, @R0 是寄存器间接寻址，它将 R0 保存的地址中的值传送给累加器 A；MOV A, R0 是寄存器直接寻址，它将 R0 的值传送给累加器 A。

3. 结合模型机逻辑框图，试描述执行操作 ADD A, R2 地址和数据的流向。

- ①PC+1 为取下一条指令字节准备，累加器 A1 内容送入 ALU 的 DR1 锁存器中；
- ②计算器 R2 的内容送入 ALU 的 DR2 锁存器；
- ③ALU 将计算结果送到累加器 A，根据加法结果置进位标志 CY；
- ④取指微指令，从内存读出指令送到指令寄存器。